🤝 Meeting Notes

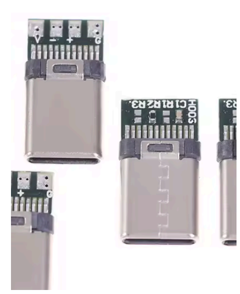
LEAVE BLANK

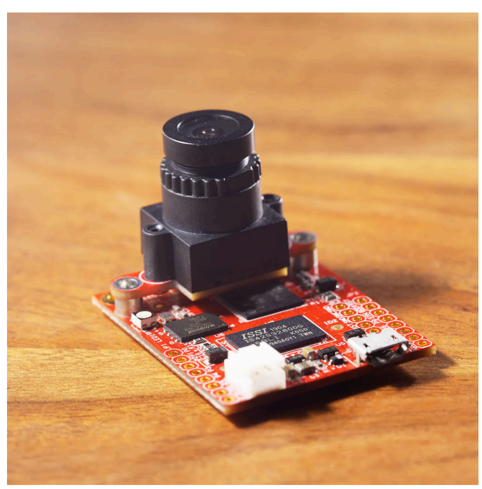
🧠 Project Ideas

Objective:
Build a phone sized eye-gaze tracker to use instead of Tobi PCEye tracker

Josephine

Idea - Adjustable eye-gaze tracker
A small compact device with mini usb camera and microcontroller that is easy to move around with and use with any phone or handheld device.

Component	Qty	Price	Image	Use
USB-C male connector component	5-10 pcs	$5+		To connect to device

Jetson Nano Camera 8MP IMX219 Binocular Camera	1	$35+		To track eye movements
3D Printed case	1	0		Designed to hold components
Microcontrollers- ESP32-S3	1	$15		To process raw data and fee to phone for control
OpenMV Cam	2	$65+ each		Alternative camera option with inbuilt Microcontroller

Additional notes:

For a **phone-based eye tracker**, you probably don't need an external MCU unless you're trying to:

1. Save phone battery, or
2. Prototype with cheap hardware before going app-only.

Most research-grade and consumer **eye gaze trackers on phones** are *just software* (camera + ML).

Beverly

My prototype idea is a compact, phone-sized eye-tracking camera that clips (e.g: with a tripod clip) under the phone and connects over USB to act as the phone's dedicated gaze sensor. Paired with third-party gaze software (examples: **Pupil Labs**' open platform for capturing gaze data, or a lightweight on-device/web solution like **WebGazer.js** or **GazeRecorder**), the module would do the heavy lifting of robust pupil/iris detection while the phone handles UI and speech output. To improve accuracy over a plain selfie camera the unit would use a small near-IR illumination ring and a camera with good frame-rate/IR sensitivity. This approach keeps the hardware minimal (a USB-C UVC camera with IR components) while giving Jessica a bottom-mounted device that's familiar to hold, reliably wired, and optimised for mobile tasks.

Hardware:

1. Small UVC USB camera module (5–12 MP, global/shutter or good low-light sensor)
    a. Example parts: OV5640/5MP USB UVC modules or small UVC machine-vision boards.
    b. **Approx cost:** $25–$100 (consumer/industrial modules vary).
2. Near-IR ring light / IR LEDs (around 850nm) + small diffuser
    a. **Approx cost:** $5–$40 (small LED boards to engineered ring lights).
3. Miniature PCB enclosure + mount/clip
    a. **Approx cost (3D-printed or off-the-shelf clip):** $5–$25
4. USB-C OTG cable or small PCB with USB-C connector
    a. For Android prototypes USB-C OTG cables allow UVC cameras to be recognized. iPhone may need MFi/adapter or dedicated app.
    b. **Approx cost:** $3–$15
5. Power management & small battery *(BONUS)*
    a. If you want the clip to be self-powered rather than drawing phone power.
    b. **Approx cost:** $8–$25

Estimated total (typical PoC): $100–$150

Software:

1. **WebGazer.js** — browser-based, webcam eye-tracking; easy to prototype UI and calibration in a web view. **Free / open**. Good for fast proof-of-concept on Android webviews.
2. **Pupil Labs (Pupil Core) + API** — open platform (hardware + open software) with a usable API and plugins for high-quality gaze data. Good if you want more accurate gaze streams and established tools. Paid hardware; software is open and extensible.
3. **Tobii Pro SDK** — free SDK for Tobii research trackers (useful if later integrating Tobii devices or comparing data). Note: PCEye (consumer Dynavox) doesn't expose the same public integration path as Tobii Pro.
4. **OpenCV / custom ML models** — for low-level pupil detection or pre/post processing if you want to build your own tracker (requires more dev time).

Notes:

- Off-the-shelf USB-C UVC webcams work readily with many Android phones (OTG), making prototyping fast. iOS is more restrictive (UVC support and app constraints), so iPhone compatibility may require a dedicated app or adapter workarounds.
- Pupil Labs (hardware + open software) is a solid option for higher-fidelity data; WebGazer.js or GazeRecorder give lighter-weight webcam approaches that run in browser/apps for faster prototyping.

Yi

I think Jo and Bev's prototype ideas are really impressive so I'd like to think from a User Journey and Experience perspective based on what I observe during our meeting.

- Problem 1: Jessica raised the accidental clicks
    - **Idea1:**
        - Replace the harsh "click" with a gradient-based dwell indicator — a circle that blooms or ripples gently when the system registers focus.
        - Include an "undo" bubble gesture that floats nearby for quick correction without losing rhythm.
        - Visually convey gaze stability through small breathing animations so she *sees* when tracking is locked.
        - Why it matters: This turns her anxiety ("I might hit something wrong") into calm confidence
    - **Idea 2:**
        - build a gaze-based pause/resume system that protects her rhythm.
        - Interaction flow:
            - Holding gaze in one corner for 2 seconds triggers "Pause tracking."
            - The UI fades gently into grayscale (calm, non-alarming).
            - A simple dwell or blink reactivates it.

            - This creates a safe zone — control over when the system listens, when it rests.

- Problem 2: Jessica mentioned how the iPhone failed her. The future gaze tracker could bridge phone and desktop seamlessly.
    - **Idea:**
        - Prototype a UI layout that adapts gaze zones automatically depending on device type — phone, tablet, or desktop.
        - Design a “transition flow” that recognizes her environment and simplifies re-calibration (e.g., connecting the eye bar instantly prompts a 3-step adaptive calibration with large, slow targets).

🚀 9/23/2025-Group Formation

Group Formation

Important Deadlines

- [x] ~~**Sep 26, 2025 : Email/Schedule with client for 1st meeting**~~
- [x] ~~**Oct 7, 2025 : Must complete 1st meeting with client**~~
- [x] ~~**When2Meet**: When to meet Jessica for the first time~~
- [x] ~~**Oct 1, 2025 : Questions for Jessica**~~

- Group members: Josephine Odusanya Beverly Yip Yi Wang (NYU)
- Before the first meeting
  - Yi Wang (NYU) Identify who on your team will be the main point of contact; send the name of the contact person to Anita/Daniel
  - Yi Wang (NYU) Have the main point of contact email the community partner- welcome and suggest meeting time (CC all group members, Anita and Daniel)
- Goals for the first meeting
  - Pick a team name - *iSee suggested*
  - Pick your internal communication plan (what platform you will use, how often you will communicate / meet)
    - **iMessage and email**
  - Pick some concrete goals for the next 1-2 weeks to get started on the project : doc
  - Focus on building relationships
    - Talk about the Community Partner's main goals for this project, or any new ideas they have
    - Make sure everyone gets to speak and introduces them-self
- Jessica
  - 24 years old; Cerebral Palsy- quadriplegic (all 4 limbs involved)
  - Uses Tobi Eye Gaze for communication
  - Student in criminal justice
  - Wants to control iphone with eye gaze

  - Jessica's story

- Jessica’s project idea
- Jessica’s team should learn to use the eyegaze.
    - Using iphone with eyegaze

Follow up feedback from Anita Sep 25th

- Send Jessica a short list of questions that your group wants to ask her. Give her a chance to prepare answers.
- She is going to use the augmentative communication device that she used in the video she made.
- When you ask questions that you haven't already given her, it will take a little while for her to respond. Just be patient waiting for her response.
- Explain the departments that your group members come from. Sometimes it's hard to differentiate between ITP and IDM so you all should work out descriptions. work for you.

📝 9/30/2025- General Class Notes

General Class Notes

Important Deadlines

Sep 30, 2025 : Establish weekly meeting times (When2Meet)

Oct 1, 2025 : Questions for Jessica

Oct 3, 2025 : Meeting with Jessica

Oct 5, 2025 : Post-meeting Discussion + Blogpost #1 Discussion + Delegate tasks in project

Oct 7, 2025 : Blogpost #1 Due

Oct 21, 2025 : Blogpost #2 Due

Milestones

1. Weekly Meetings set (subject to rescheduling) : Weekly on Thursdays, 9:30pm -10:30pm
   Upcoming meetings: Oct 5th 9pm-10pm
   Oct 9th 11pm-12pm (schedule readjustment)
2.

Weekly Deliverable:

- [x] ~~Task Delegation (Beverly Yip) here.~~
- [x] ~~Update website page with group names and project description (Josephine Odusanya)~~
  - Update blog based on first meeting with community partner
    - Update front page
    - First weekly update
- [x] ~~Take meeting notes and add to documents from the first meeting with community partner (Yi Wang (NYU)) here~~

Reminders*

Remember to add external files to DAT Drive

Update Project Details here:
https://wp.nyu.edu/ap_classes_dat_f25/wp-admin/

- ➔ **Project Information: https://wp.nyu.edu/ap_classes_dat_f25/projects/**
- ➔ **Important Pages: Project Template, Resources Template, Weekly Update Template**
- ➔ **Project websites** (2022 Example) (2023 Example) (2024 Example)

MIDTERM DELIVERABLES

- 5-7 minute presentation videos, with embedded captions (we recommend zoom) and posted to NYU Stream.
- Videos must be uploaded to team webpage before class
- Midterm Grading rubric (for faculty)
- Feedback form (for student and community partners)

FINAL DELIVERABLES

- 7-10 minute presentation videos, with embedded captions (we recommend zoom and NYU Stream) and posted to your project website.
- Final project website
- Final Grading rubric (for faculty)

Project Title Suggestions

1. iSee: Mobile Eye Gaze
2. iSee: Eye Phone Tracking (iPhone = eye phone 😅)
3. iSee: Mobile Vision Accessibility
4. iSee: Control at a Glance

For the final project, we shouldn't do a simulation- it should be usable by a person.

Reminders*
Remember to add external files to DAT Drive

🤝 10/03/2025- Meeting with Jessica

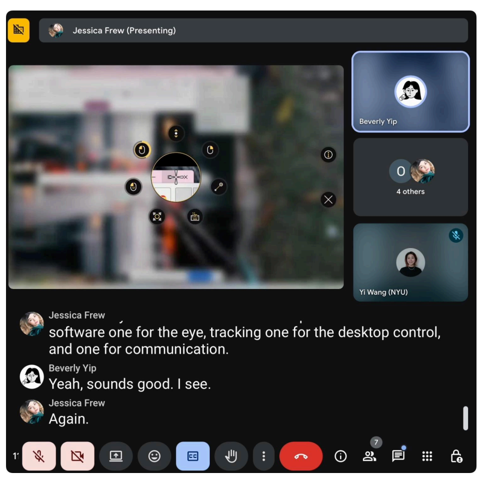
Meeting recording:
https://stream.nyu.edu/media/First+Meeting+with+Jessica.Oct+1/1_pa3inyp5

Meeting Notes - First Meeting with Jessica

Intro :
I'm 24 years old. I have cerebral palsy and communicate with a Toby eye gaze computer. I'm a model, actress, children's author and student maturing in criminal justice. I was a teacher slash counselor, student volunteer for children and adults with disabilities. In the center, I used to counsel this boy, one on one, on his behavior and train him on using the TDI series for a year and a half before I started college. In my first semester of college, I mentored a group of individuals with all types of learning.

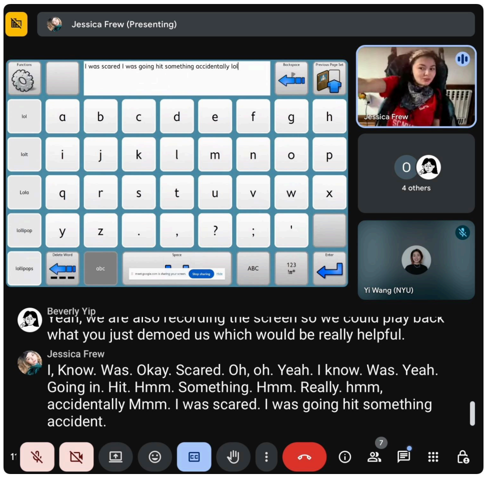
First:
Jessica shared her communication software, and showed us how she controls the desktop. There are three types of software, one for eye tracking, one for desktop control, and one for communication.

Notice – User flow
With the control software:
There are mice icon to choose the thing she want, magnifying tool icon to precise the UI on the interface,
with communication software:



When she typing one alphabet with eye track, the software will show associate words, and she will choose the right word, then the computer will play the sound to pronounce the word she choose

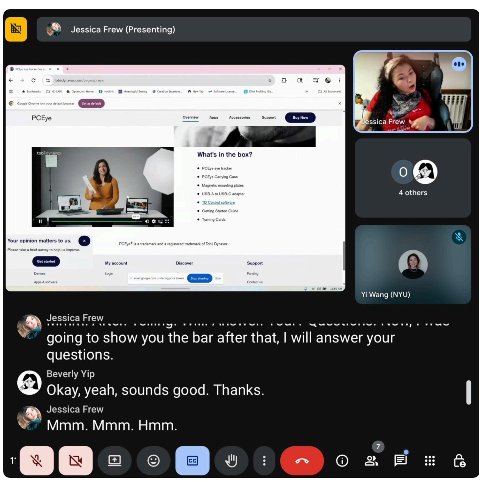
She worried she hit something accidentally, knocking loudly.



Second:
She showed us the bar of the typing software for searching on google
Website she share:
[myTobiiDynavox]

Her opinion of PCEye:
[PCEye eye tracker for accessibility - Tobii Dynavox Global]

?They lie about using it on the phone?



The middle part
That's a correct size for the phones, the size of the middle of the bar
It's need to be the size of the bottom of the phone

She is using the old phone, I can't use it so there is no point
There is no role model.
She tries to use it on her bf's iphone, it's horrible. It's all over the place.

Her ideas about the device:
It would be on the bottom connected by USB, USB would be on the top of the device.

What's been answered

1. Basic Communication / Expectations
    a. Smaller version of the PCEye (Tobii Dynavox) for her phone
2. Current understanding of tech:
    a. Jessica knows there are *three distinct software systems* — one for **eye tracking**, one for **desktop control**, and one for **communication**.
    b. e.g., magnifying tool for precise control, predictive text suggestion, etc.
3. **PCEye (Tobii Dynavox)** and **myTobiiDynavox**
    a. She's *skeptical* about Tobii's claim that it can work on phones ("They lie about using it under sunlight"), implying she's tested and found limitations firsthand.
    b. She's also thought about *hardware form factor and connection design* ("USB on top, device at bottom") / iphone eye tracking is terrible
4. Eye tracking and control:
    a. She demonstrated how she types and selects items using gaze, and noted concerns about *accidental hits* ("worried she hit something accidentally, knocking loudly").
    b. This shows she's aware of **precision issues** and **unintentional input**.
5. Head movement/distance:
    a. Not directly discussed, but her description of *eye tracking and typing workflow* suggests she's fairly stationary and uses fixed positioning when operating.
6. Connection preference:
    a. She proposed a **USB connection** (connected at the bottom, USB at the top).
    b. Suggests she finds wired more reliable than Bluetooth or wireless.
7. Daily Routines / Context:
    a. No mention of how long she uses it daily, what times she uses it, or which activities matter most.

Question Area	**Missing Info / Clarification Needed**

Expectations	What does she hope to achieve through collaboration? (awareness, prototype, usability feedback?)
Technical Development	Does she want to design and prototype this herself, or partner with engineers/designers?
Eye Movement Details	How do her eyes rest or move naturally? Are there any screen zones harder to track?
Head Range	What's her comfortable head movement range? How far is she typically positioned from the screen?
Power/Connectivity	Would she prefer a self-powered eye bar (with its own battery)? Would wireless ever be feasible?
Daily Routine	What's her typical daily use pattern — duration, comfort, main apps? When does fatigue set in?
Calibration	What happens when tracking breaks or loses calibration? How often does that occur?
Lighting Conditions	What environmental factors (lighting, distance, background) affect tracking?
Assistance	Does she need someone to help set up the system daily? Would she like more independence?
Backup Communication	What alternative communication methods does she use when the device fails?
Dream System	If she could design her ideal system — what features, look, or feel would it have beyond hardware form?

Jessca's generated meeting notes:

NYU DAT Project First Meeting

Transcript

https://otter.ai/u/g-kkV_en1E2l3kqCRxiYxQEsRpI?view=summary

In the conversation, participants discuss the use of a note-taking tool and the challenges faced by a speaker with cerebral palsy who communicates using a Tobii eye gaze computer. The speaker shares their background as a model, actress, and student in criminal justice, with experience in counseling and mentoring individuals with disabilities. They demonstrate their communication software, which includes eye tracking, desktop control, and communication components. There is a discussion about the size of a software feature, with the speaker expressing a preference for it to be smaller, particularly for phone compatibility. The conversation also touches on the recording of the meeting for future reference.

Action Items

- [ ] @Yi Wang - Provide more time for the speaker to answer the pre-listed questions.
- [ ] Record the screen during the speaker's demonstration for future reference.

Outline

Introduction and Note-Taking Setup

- Yi Wang (NYU) mentions looking for a note taker and acknowledges the auto note taker shared by Jessica.
- Speaker 1 and Unknown Speaker discuss the limitations of the auto note taker, noting it struggles with capturing who is speaking.
- Jessica joins the meeting, and Yi Wang (NYU) greets her.
- Speaker 1 informs that Josephine will join shortly.

Participant Introduction and Background

- An unknown speaker thanks everyone for their time and introduces themselves as a 24-year-old with cerebral palsy who communicates using a Tobii eye gaze computer.
- The speaker shares their background as a model, actress, children's author, and student in criminal justice, with experience as a teacher and counselor for individuals with disabilities.
- The speaker mentions their experience mentoring students with various disabilities and offers to share their screen to demonstrate their setup.

Screen Sharing and Technical Issues

- Speaker 1 and Unknown Speaker discuss the inability to share the screen on Google Meet.

- Speaker 1 provides instructions on how to share the screen, and the unknown speaker attempts to follow the steps.
- The unknown speaker successfully shares their screen and begins to demonstrate their communication software.
- Speaker 1 mentions that the screen is being recorded for future reference.

Demonstration of Communication Software

- The unknown speaker explains their communication software, which includes eye tracking, desktop control, and communication components.
- Speaker 1 and Unknown Speaker discuss the visual explanation provided and express appreciation for the detailed demonstration.
- Speaker 1 asks if the unknown speaker can answer some of the questions listed in the documentation during the meeting.
- The unknown speaker agrees to answer the questions and mentions showing a bar before answering.

Discussion on Software Features and Preferences

- Speaker 3 and Unknown Speaker discuss the middle part of the software and its size, specifically for phones.
- Speaker 1 and Unknown Speaker clarify the dimensions being referred to, with the unknown speaker indicating the need for a smaller size.
- The conversation shifts to the type of phone the unknown speaker is using, with the unknown speaker mentioning an old iPhone.
- Speaker 1 inquires about the specific model of the iPhone, and the unknown speaker confirms it is an old version due to accessibility limitations.

Josephine's generated meeting notes:
#1

Transcript

https://otter.ai/u/9mN-sWBGGcgVQx7YI10oWfGB5t0?view=summary

The meeting involved a demonstration of Josephine's communication software, which includes eye-tracking, desktop control, and communication components. Beverly Yip confirmed the recording of the screen for future reference. Josephine demonstrated the software's functionality, expressing concern about accidentally hitting something. The team discussed the size of the middle part of the device, clarifying it should be the full width for the phone or the eye-tracking portion. There was confusion about the device's compatibility with different phone models, with Speaker 4 mentioning the use of an old iPhone due to functionality issues. Speaker 1 and others shared experiences with new eye-tracking technology on iPhones, noting its poor performance.

Action Items

- [ ] Determine the appropriate size of the middle part of the software/device for use on phones.
- [ ] Investigate getting a new phone that is compatible with the communication software.

Outline

Eye Tracking and Desktop Control Demonstration

- Speaker 1 explains the use of three software: one for eye tracking, one for desktop control, and one for communication.
- Beverly Yip and Josephine Odusanya confirm that the demonstration was clear and fast-paced.
- Beverly Yip mentions that the screen is being recorded for playback, which is helpful for understanding the demo.
- Speaker 1 reassures that no accidents occurred during the demonstration.

Addressing Questions and Clarifications

- Beverly Yip asks if Speaker 1 can answer the questions listed in the documentation during the meeting.
- Speaker 1 confirms they will show the bar and then answer the questions.
- Multiple speakers express confusion and attempt to clarify the discussion, leading to a mix-up in the conversation.
- Beverly Yip and Josephine Odusanya seek clarification on the size of the middle part of the software or device.

Discussion on Device Size and Compatibility

- Beverly Yip and Josephine Odusanya discuss whether the middle part of the device should be the full width or just the eye tracking portion.
- Speaker 4 clarifies that the middle part is the correct size for phones.
- Beverly Yip asks if the middle part is the full width or only the eye tracking part.
- Josephine Odusanya confirms that the middle part is the width of the phone.

Challenges with Old Devices

- Speaker 4 mentions having a very old iPhone that is not functional.
- Speaker 6 discusses the need to send money to a cousin to get a new device.
- Speaker 7 shares a negative experience with using a new eye tracking device on their boyfriend's iPhone.
- Speaker 1 and Speaker 4 describe the difficulties encountered with the new eye tracking technology.

#2

Transcript

https://otter.ai/u/rGu1VYDgC0IqyHJTvtvVRQh6t2w?view=summary

The conversation revolves around the placement and connectivity of a smaller version of a device, which is intended to be connected to a phone at the bottom via USB. There is some confusion about the orientation of the USB port, which is clarified as being on the top of the device. Josephine Odusanya inquires about Jessica's thumb movement while using the device, which Jessica clarifies is not related to scrolling or eye gaze but a body movement. The meeting concludes with a request for Jessica to answer questions via email and for Josephine to gather options for the project's feasibility, involving Anita, their professor. Speaker 3 also mentions being busy with a $20 million movie project.

Action Items

- [ ] Answer the questions sent in the form.
- [ ] Come up with a few options for the project and send them to Jessica and Anita.
- [ ] Discuss the feasibility of the project options with Jessica and Anita.

Outline

Discussion on Device Placement and Connectivity

- Beverly Yip questions the placement and use of the smaller version of the device, expressing uncertainty about its positioning.
- Speaker 3 clarifies that the device would be placed at the bottom, connected via USB and VM.

- Multiple speakers discuss the orientation of the USB port, with Speaker 3 repeatedly stating it would be on the top of the device.
- Josephine Odusanya confirms the bottom of the phone as the intended placement, and Speaker 4 mentions charging the phone.

Clarification on USB Port and Device Placement

- Beverly Yip seeks further clarification on the device's bottom placement.
- Josephine Odusanya asks Jessica to confirm if the bottom of the phone is the intended placement.
- Speaker 4 and an unknown speaker discuss the USB port's location, confirming it would be on the bottom.
- Beverly Yip and Josephine Odusanya ensure they understand the placement and connectivity details.

Josephine's Question to Jessica

- Josephine Odusanya inquires if Jessica moves her thumb to assist with scrolling or using eye gaze.
- Jessica responds that it is not related to scrolling but rather a body movement.
- Jessica humorously mentions that it is just a body movement, causing laughter among the participants.
- Josephine Odusanya thanks Jessica for her input and mentions the need for further communication via email.

Next Steps and Meeting Conclusion

- Josephine Odusanya informs Jessica that a meeting was canceled and requests her to answer the questions sent in the form.
- Jessica agrees to answer the questions to the extent possible.
- Josephine Odusanya outlines the next steps, which include creating options for the project and discussing feasibility with Anita, their professor.
- Beverly Yip and an unknown speaker express appreciation for Jessica's time and efforts, with the meeting concluding on a positive note.

🏆 10/05/2025 - Team Meeting

Email to ask profs if they're available after class Tuesday at Washington Square just for 5 of us
Let them know what we get from Jessica and want to discuss more details
Develop some ideas for the project

Tech part

Detailed Summary

Main Themes & Discussion Points

1. **Technical Issues at Start**
   - Josephine's Wi-Fi kept cutting in and out, which caused disruptions.
   - Eventually, she switched to phone data to stabilize the call.
2. **Reviewing Jessica's Feedback**
   - Beverly had both sets of documents open and emphasized cross-checking Jessica's answers against their prepared questions.
   - Key takeaway: Jessica provided answers, but in **story form** rather than direct responses.
   - Team agreed to extract structured points from her notes and send a follow-up email only if necessary.
3. **AI Note-Taking**
   - Josephine mentioned using AI note-takers during the meeting with Jessica.

- Limitations: One tool only captured the first 30 minutes. Josephine restarted it multiple times to cover the rest.
- She promised to compile those AI-generated notes along with her manual notes.

4. **Project Direction – Eye Tracker Device**
    - Jessica expressed interest in a **small version of the eye tracker** for use with her phone.
    - Issues raised:
        - Uncertainty about her phone size.
        - Josephine suggested using an **adjustable tripod** as a flexible solution.
    - General agreement that the design should be **lightweight, adaptable, and simple**.
5. **Decision on Technical Depth**
    - Josephine asked: “How technical do we want this to be?”
    - Discussion leaned toward starting with a **basic prototype** and iterating based on Jessica’s feedback rather than overcomplicating from the start.

Next Steps Action Plan

1. Compile and Organize Meeting Notes

- **Owner:** Josephine
- **Tasks:**

- Gather AI-generated notes (multiple recordings) + manual notes.
- Clean up overlapping or incomplete sections.
- Share compiled versions with team by email/Drive.

- **Deadline:** Within 2–3 days.

---

2. Cross-Check Jessica's Feedback

- **Owner:** Beverly
- **Tasks:**
  - Compare Jessica's responses with the original list of questions.
  - Highlight which questions were answered in story form.
  - Identify unanswered/unclear items.
- **Deadline:** End of the week.

---

3. Clarify Remaining Questions

- **Owner:** Yi (with Beverly's support)
- **Tasks:**
  - Draft a short, clear follow-up email to Jessica.
  - Focus only on unanswered points (avoid repeating answered ones).

- **Deadline:** After Beverly's review of notes (next week).

---

4. Prototype Planning – Eye Tracker

- **Owner:** Whole Team
- **Tasks:**
  - Explore design options for a **small, phone-compatible eye tracker**.
  - Research adjustable tripod or universal phone mounts.
  - Note potential constraints (phone size, weight, alignment).
- **Deadline:** Prepare initial options for next team meeting.

---

5. Technical Depth Discussion

- **Owner:** Whole Team
- **Tasks:**
  - Decide scope for first prototype (minimal viable version vs. technical features).
  - Ensure focus stays on **user-centered, lightweight design**.
- **Deadline:** Next team meeting.

---

6. Documentation

- **Owner:** Beverly (with inputs from Josephine & Yi)
- **Tasks:**
  - Maintain a running document summarizing progress, design choices, and Jessica's feedback.
  - Keep the updated version accessible to all.
- **Deadline:** Ongoing, with updates after each milestone.

⚡ **Overall Priority:**

1. Finalize clean notes (Josephine).
2. Cross-check and clarify Jessica's feedback
3. Start early-stage design exploration

🏆 10/09/2025 - Team Meeting

📝 10/7/2025- General Class Notes

General Class Notes

Upcoming meetings: Oct 9th 11pm-12pm

- Midterm presentation
  iSee_Assistive Technology_Midterm Presentation
- Outline guide here: DAT-F25-MidtermGuide
- 5-7 minute presentation videos, with embedded captions (zoom and NYU Stream).
- Post link to video, transcript and slides: https://wp.nyu.edu/ap_classes_dat_f25/projects/midterm/

Update Project Details here:

https://wp.nyu.edu/ap_classes_dat_f25/wp-admin/

Important Deadlines

Oct 21, 2025 Midterm presentation due

Dec 9, 2025 Final Presentation

Milestones

3. **First blog update up along with group description**
4. **Clarified meeting direction**

Weekly Deliverable:

- [ ] Finalize project direction
- [ ] Delegate tasks
- [x] ~~Update Jessica with ideas and timeline~~
- [ ]

Additional Notes

Meeting with Anita and Daniel:

- Explore how eye gazing work on iphone

Reminders*

Remember to add external files to DAT Drive

- A figma version of what the app would like/digital version
- Evaluate her:
    - How she holds her phone
    - How she holds her phone
    - How she move around the head/eyes
- Go to Apple store
    - Ask worker in the store for teaching use eye-tracking
- It can be some problems because she using old version one
- For plug in problem, maybe because of the cable
- Research what is available:
    - Research with gaze user using iphone
    - Solutions and open source for eye gazing
    - Just go and explore as far as we can,
    - Bring our skills to the project, use the source we have in our program, department (Coding lab etc.)
    - We can end up being reminder the stakeholders such as Apple/Google/Tobi what they missed in this piece for their users
- Email to Jessica
    - What we can do based on our level and ask for her understanding
    - We hope we are in the same page in terms of expectations

Follow up:

- Doing research about the eye gazing: Apple Stores
- Thinking about how to build the it
- Doing some simple CAD prototyping design
- Combine our practices together
- Apple Accessibility Support: https://support.apple.com/en-us/111749
- Email Jessica
    - What we can do based on our level and ask for her understanding
    - We hope we are in the same page in terms of expectations
-

Reminders*

Remember to add external files to DAT Drive

Reminders*

Remember to add external files to DAT Drive

🏆 10/16/2025 - Team Meeting

Meeting summary

Quick recap

The team discussed progress updates on their CAD model and hardware components, including eye-tracking technology and potential materials from the lab. They aligned on the structure and content for their upcoming midterm presentation, which will cover hardware design, software implementation, and overall project goals. The team established a timeline for recording and submitting their presentation video, with specific roles assigned for content creation, editing, and review to meet the upcoming deadline.

Next steps

- Josephine to complete the CAD model design by tomorrow morning.
- Josephine to finalize the hardware components list and potentially purchase some items like CV cameras.
- All team members to create and update their respective sections in the Google Slides presentation by end of Saturday.
- All team members to record their individual presentation parts with captions by end of Saturday.
- Beverly to handle the video editing and compilation of all recorded parts on Sunday.
- Yi to focus on presenting the sunlight issue and small icons problem as the main pain points.
- Yi to cover the competitive landscape in her presentation section.
- Beverly and Josephine to prepare the "next steps" section for the end of the presentation.
- All team members to review the final compiled video on Monday before Tuesday's submission.
- All team members to ensure their recordings include transcripts and proper captions.

Summary

CAD Progress and Midterm Planning

Josephine provided an update on her progress with the CAD model, mentioning she would complete it that night and share it the following morning. She also discussed potential hardware components, including CV cameras, and noted that her lab might have spare materials. Beverly and Yi agreed to focus on aligning for the midterm presentation, which Josephine planned to cover hardware design and future plans. Beverly confirmed the structure for the midterm presentation, which would include hardware, software implementation, and overall design, addressing the problems Jessica had mentioned.

Eye Tracking Web App Demo

Beverly demonstrated a web app she developed using the WebGazer API for eye tracking, which includes a calibration process using a 5x5 grid and a red dot to indicate eye position. She explained that while the current version is not fully stable, it shows promise for future development. Beverly also showed a third view where she used her eyes to dwell and gaze without clicking, though the video quality made it difficult to see the full demonstration.

Eye-Tracking API Proof of Concept

Beverly presented her progress on a proof of concept that integrates eye-tracking technology with an API, aiming to test feasibility and viability for a final project. She explained the technical details and next steps, emphasizing the connection to Josephine's hardware work. Yi shared observations from a meeting recording with Jessica, highlighting challenges in using the app, such as the difficulty of targeting specific points due to the small gap between the keyboard and screen. Beverly and Yi discussed the need to align their efforts for the project's success.

Presentation Structure and Focus Areas

The team discussed the flow and structure of their upcoming presentation, with Beverly proposing a sequence that starts with Yi's introduction of user problems and research, followed by Joe's design presentation, and concludes with both software and hardware next steps. They agreed to prioritize the sunlight issue and small icons as main focus areas, with 2-3 additional "nice-to-have" items to be determined. The team also confirmed they need to record presentation videos with captions, which Beverly will coordinate to be done in parts and compiled together.

Google Slides Presentation Planning

The team discussed using Google Slides for their project presentation, with Yi sharing a link to an existing presentation. Josephine agreed to request edit access for the team, which Beverly and Yi also did. The group planned to start adding content to the presentation once access was granted.

Presentation Video Recording Plan

The team discussed the timeline and process for recording and submitting a presentation video. They agreed to record individual segments on Saturday, with each person speaking for 2 minutes per section, and then compile the footage on Sunday to create a seamless final video. Beverly will handle the editing, while Josephine and Yi will focus on recording their parts. They decided to use Zoom to record with captions, which will be edited and reviewed before submission. The team aims to complete all tasks by the end of Saturday, with a final review on Monday and submission on Tuesday.

📝 10/21/2025- General Class Notes

General Class Notes

- Today we just had live feedback on the midterm presentations.
- Feedback from class:
  - Use a dongle that has two ports so that you can charge at the same time, since Jessica won't bring the laptop with her outside
  - Magnetic charger
  - Make a stand to hold the phone (there are portable chargers that exist that allow phones to stand when plugged in below)
- Breaking news: Jessica Frew (iSEE) is now Miss Wheelchair NJ!!!
-

Important Deadlines

- [ ] **Tuesday, December 9th (Final Project Presentation)**

Milestones

5. Completed midterm presentation
6. Updated Blog post #2
7.

Weekly Deliverable (Due 10/28/25):

- [ ] Give feedback to your classmates on midterm projects
- [ ] Complete midterm reflection
- [x] ~~Send Embed Code to Daniel~~
- [ ] Keep your team website up to date!
- [ ] AT Presentation signup
- [ ] Submit Makerspace reflection form (if not done)
- [ ] Reading: Design for Empowerment
- [ ] Questions for Guest Lecture (Sami) (Due by 1pm 11/4)

Additional Notes

- [ ] Sent email to Jessica about updating Yi Wang (NYU)

Reminders*

Remember to add external files to DAT Drive

🏆 10/23/2025 - Team Meeting

Tasks Delegation (Due Oct 30, Thurs)

Jo

Bev

Yi

- Send follow-up email to Jessica? Faculty sending it?

All

- Continue researching/ scoping out what we did for midterms
- Report back on a possible timeline of your own individual research for the next 7 weeks
- We'll see how to combine everyone's works into 7 overall weeks

Meeting summary

Quick recap

The team discussed workload management and upcoming travel plans, including calendar passcode setup and recording requirements for classes. They reviewed feedback from their midterm presentation and discussed project timelines, with approximately 7 weeks remaining until the final presentation. The team agreed on a new meeting structure with weekly updates on Tuesdays and more detailed discussions on Thursdays, while also planning to spend the next week researching individual project areas.

Next steps

- Beverly: Add passcode to the calendars
- Josephine: Purchase components by October 25th
- Josephine: Start putting web gazer.js or similar software on the microcontroller and gather components for wiring
- Josephine: Work on the casing
- Yi: Send a follow-up email to Jessica
- All team members: Continue researching and scope out what can be done for the project
- All team members: Report back on Thursday with possible timeline of individual research over the next 7 weeks
- All team members: Complete mid-term reflection by next week

- All team members: Keep team website up to date
- All team members: Complete AT presentation sign up
- All team members: Submit makerspace Reflection Form
- All team members: Do reading on Design for Empowerment

Summary

Workload and Travel Planning

Beverly and Josephine discussed their workload and upcoming travel plans. Beverly mentioned she had to finish a lot of work before her flight the next morning. They briefly talked about adding a passcode to calendars and Josephine's experience fixing recording issues at Fenway. Beverly expressed frustration about having to record demos for half of her classes, which added to her workload, especially for group projects.

Microcontroller Project Planning Discussion

Josephine discussed her plan to start working on a project involving microcontrollers and web gazer.js software. She mentioned having two unused microcontrollers at home and offered to purchase additional components by October 25th. Josephine also mentioned working on the casing for the project simultaneously.

Midterm Presentation Feedback Review

The team discussed feedback received for their midterm presentation, which primarily focused on hardware-related areas. They debated whether to send a follow-up email to inform their supervisor about the presentation, with Yi suggesting it might be redundant since their supervisor had already been informed. The team also considered revising their food bag design based on the feedback, though Beverly noted it was mostly hardware-focused and didn't indicate any major issues with their overall direction.

Semester Assignments and Planning

The team discussed upcoming assignments, including a self-evaluation due next week and a feedback form that will be shared the following week. Josephine reviewed the assignment page, which included tasks like providing feedback, completing a mid-term reflection, and submitting a makerspace Reflection Form. Beverly noted that after the faculty discussion, they agreed on the scope of work for the semester and suggested using the current session to plan and scope out their projects.

Project Timeline and Meeting Structure

The team discussed their project timeline, with approximately 7 weeks remaining until the final presentation. They agreed to spend the next week exploring and researching their individual areas, similar to what they did in their first meeting. They decided to meet on Tuesdays for 30 minutes to discuss updates, with Thursday meetings reserved for more detailed discussions. Beverly will organize their project delegation file to reflect this new structure. Yi mentioned he would send a follow-up email soon.

📝 10/28/2025- General Class Notes

General Class Notes

- ○ Clearly send the questions with yes and no
- ○ Send the choice we made with clearly reason

Important Deadlines

- [ ] **Tuesday, December 9th (Final Project Presentation)**

Milestones

8.
9.

Weekly Deliverable (Due 10/28/25):

- [ ]
- [ ] Reading: Design for Empowerment
- [ ] Questions for Guest Lecture (Sami) (Due by 1pm 11/4)

Additional Notes

- [ ] Sent email to Jessica about updating Yi Wang (NYU)

Reminders*
Remember to add external files to DAT Drive

🏆 10/30/2025 - Team Meeting

Timeline
Oct 30, 2025 (today)
Nov 4, 2025
- Jo puts together hardware this week
    - Will determine scope of project
- Bev wait for Jo's feedback + update timeline to match Yi's (Figma designs)

Nov 11, 2025
Nov 18, 2025
Nov 25, 2025
Dec 2, 2025
- Have an idea of what we will be presenting for final presentation
- Task delegation for slides (recording/ transcripts/ video compilation etc)

Dec 9, 2025 : Final presentation

Yi's Timeline: Team Notes

Midterm Feedback: Team Notes
- Clarify software component
    - App?? Phone permissions??
- Backup: work on hardware design and software prototype design (no actual software)

TODO:
1. Talk to faculty during class on Tuesday about Midterm Feedback !!

Meeting summary

Quick recap

The team discussed their project timeline and confirmed key deadlines, with Beverly and Josephine agreeing to continue working on their respective hardware and software components. They reviewed progress on various aspects including hardware parts delivery, design work, and prototype development, while acknowledging the need to create a final presentation video. The team also addressed challenges with implementing eye-tracking software and discussed feedback from their professor regarding the project's focus on

iPhone usage and eye gaze technology, deciding to seek further clarification during the next class session.

Next steps

- **Josephine: Assemble hardware parts this weekend once camera arrives on Saturday**
- **Josephine: Update team via DM/text when hardware assembly is complete**
- **Josephine: Research phone app permissions regarding whether apps can maintain settings/control when closed**
- **Beverly: Update timeline document based on Yi's timeline and design collaboration needs**
- **Beverly: Work closely with Yi on lo-fi prototype and design during November period**
- **Beverly and Yi: Collaborate on Figma prototype for app design**
- **Beverly, Yi, and Josephine: Schedule office hours discussion with professor during next Tuesday's online class to clarify project direction and software requirements**
- **Beverly and Yi: Update Josephine after meeting with professor about feedback and direction**

Summary

Project Timeline and Roles Review

Beverly and Josephine discussed the timeline for their project, noting that they have approximately five to six weeks left until the final presentation on December 9th. They reviewed the syllabus and confirmed that there are no additional presentations required before the final one. Beverly mentioned that she would be working on the hardware side of the project, while Josephine focused on the software aspect. They agreed to continue working on their respective parts and to review a timeline document later.

Hardware Project Progress Update

Josephine discussed her progress on a hardware project, explaining that she is waiting for parts to arrive, including a camera, by the end of the weekend. She mentioned that she had to change the camera model due to its insufficient quality for a minimal viable product. Josephine plans to start assembling the parts once she has them and will update Beverly on her progress, either before or after Thursday's meeting. Beverly agreed to follow up with Josephine via text if there are any updates before the next meeting.

Design Progress and Timeline Alignment

Yi will work on resolving small icon issues for iPhone and conduct research on Pinpong, while also developing a low-fidelity concept for scars in Figma and creating prototypes based on hardware designs. Beverly will assist with design feedback and potentially adjust her timeline to align with Yi's progress, particularly focusing on balancing design quality with proof-of-concept goals. The team acknowledged the need to create a final presentation video and agreed to reassess their timeline to accommodate this task.

Midterm Feedback Review Confusion

The team discussed midterm feedback, which was unclear and confusing. The professor wanted them to focus on using the iPhone specifically and to acknowledge eye gaze technology, though Josephine was unsure why this was a problem since they had shown images of smaller devices. Beverly and Josephine reviewed the comments, which included feedback about including information on how other people access their phones and using a specific keyboard for communication. Yi suggested they may have misunderstood the professor's focus, as they had emphasized eye gaze technology throughout their presentation.

Eye-Tracking Software Integration Challenges

The team discussed challenges with implementing eye-tracking software for a phone-based project, with Beverly expressing confusion about how to integrate the software with existing phone systems. Josephine suggested two possible methods: plugging in a device or using Bluetooth, both requiring some form of app connection. The team agreed to seek clarification from their instructor during the next class session, potentially using the scheduled office hours for this purpose. As a backup plan, Beverly proposed focusing on the hardware design and user experience, while Josephine committed to researching further and understanding the instructor's feedback to avoid future confusion.

📝 11/4/2025- General Class Notes

General Class Notes

- Look into patents/ iPhone/ Apple physical & software restrictions - how far does Apple let you manipulate device settings?
- https://apps.apple.com/us/app/eye-gaze-communication-board/id6446154016
    - Designed for the iPad
    - Faster than typing
    - Communication Cards
- Jessica's feedback: "*The only thing is the USB connector. It should be on* <u>*top of the device*</u> *instead of on the side of it. So you can use it vertically.*"
    - 

    - There exists portable chargers that you plug UNDER the phone and you can still have it stand upright.
- It is <u>okay</u> to look into feasibility and limitations of software and include that in final packet/ presentation ("this is what we are going to do next")
    - a **Figma prototype and documentation packet**—covering hardware, interaction flow, and identified challenges—would be sufficient.framing the project as a **prototype packet** to hand off for future development:
        - Hardware process & considerations (note that reaching out to Tobi/ Apple was a deadend)
        - UX & interaction logic
        - Technical or legal feasibility issues (e.g., Apple's hardware restrictions
- Concerns were raised about app behavior (e.g., calibration or control

Reminders*
Remember to add external files to DAT Drive

stopping when the app is closed).
    - How do non-verbal users use the iPhone?
    - **Confirmed:** Bev researches high-level overview on software restrictions with iPhone + starts working with Yi on Figma Prototype

---

Important Deadlines

- [ ] **Dec 9, 2025 - Final Presentation**

Milestones

10. Nov 6, 2025 - Get physical prototype assembly feedback from Jo
11. Nov 13, 2025 - Software (and/ or hardware) Restrictions with iPhone from Bev
12. Nov 13, 2025 - Follow up with Jessica regarding her experience using the iPhone's built-in Eye Tracking feature (Bev)
13. Nov 13, 2025 - Low Fidelity Wireframe/ design ideas from Yi

---

Weekly Deliverable (Due 11/11/25):

- [ ] Screen Reader Tutorial
    - [ ] Windows: NVDA OR JAWS
    - [ ] Mac OS (access through OS Accessibility settings)
- [ ]

Additional Notes

- [ ]

Reminders*

Remember to add external files to DAT Drive

🏆 11/13/2025 - Team Meeting

Timeline

Nov 4, 2025

- Jo puts together hardware this week
  - Will determine scope of project
- Bev wait for Jo's feedback + update timeline to match Yi's (Figma designs)

Nov 13, 2025 (today)

Nov 18, 2025

Nov 25, 2025

Dec 2, 2025

- Have an idea of what we will be presenting for final presentation
- Task delegation for slides (recording/ transcripts/ video compilation etc)

Dec 9, 2025 : Final presentation

Designs:
https://www.figma.com/design/djyWW2wGiASZ8g0luf5ggn/Jessica-Tobii-User-Journey-Map?node-id=0-1&t=mcC2AFJjPt3dU0ah-1

TODO:

Yi:

- [ ] Send email out to Jessica (draft below) - cc Anita + Daniel
  - [ ] Include annotated frames (image)

Bev:

- [ ] Work with Yi on redesigning

Jo:

- [ ] A hardware to web app type poc for final presentation

Updates

Beverly's Research on iOS Feasibility: Team Notes

Yi's Research:

- Small icon research: Team Notes
- Designing Interface: Team Notes
- Lo-fi Ideas: Team Notes

Jo's Research:

- Instruction Manual: Team Notes

What To Present

Software: poc / the scope + research with ios

Hardware: research / the build / test it locally / present the goods and the bads

Designs:

- One thing we could think about is redesigning existing tools she already uses with Tobi Eye Gaze
- Typing tool (keypad) how a iPhone keyboard can be accessible to eye gazing user

To Jessica:

- We still need to understand what her exact issues were with the iPhone eye tracker

---

Email to Jessica:
Hi Jessica,
I hope you're doing well! We're currently at the stage of redesigning our iOS prototype (**a low fidelity design is attached in this email**), and we want to ensure that what we create aligns closely with the tools you regularly use. To help us move forward, we wanted to follow up with a few specific questions:

1. Tools You Use With Tobii EyeGaze
Could you share which tools or apps you commonly use with your Tobii setup? For example:

- What keyboard or communication apps do you typically use?
- Are there any specific features within those apps that are especially helpful or important to you?

2. Your Experience With iPhone Eye Tracking
You mentioned previously that the iPhone's eye gaze feature was "very bad." We'd love to understand this better. Could you elaborate on:

- Whether the problem was with the calibration process?
- Or if the issue was with using eye gaze itself — and if so, what exactly wasn't working well?

Any details you're able to share will be incredibly helpful as we work on redesigning similar tools for iOS.

If it's easier to discuss this verbally, we're more than happy to set up a quick call anytime that works for you.

Thank you again for all your insight — it's been truly invaluable to our project!

🏆 11/19/2025 - Team Meeting

Timeline

Nov 19, 2025 (today)

- Jo: demo hardware
- Yi/ Bev: figure out designs
- All: waiting on Jess's response

Nov 25, 2025

Dec 2, 2025

- Have an idea of what we will be presenting for final presentation
- Task delegation for slides (recording/ transcripts/ video compilation etc)

Dec 9, 2025 : Final presentation

Designs:
https://www.figma.com/design/djyWW2wGiASZ8g0luf5ggn/Jessica-Tobii-User-Journey-Map?node-id=0-1&t=mcC2AFJjPt3dU0ah-1

TODO (due next meeting):

Yi:

- [ ] Research accessible apps that have non-traditional sized keyboards and enlarged icons and put them in the figma file

Bev:

- [ ] Research accessible apps that have non-traditional sized keyboards and enlarged icons and put them in the figma file

Jo:

- [ ] Get camera working

Updates

What To Present (from last week)

Software: poc / the scope + research with ios

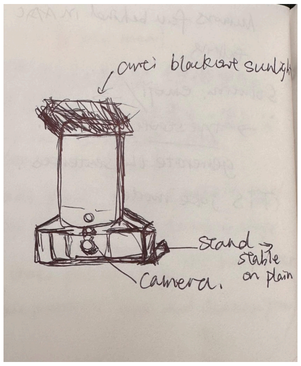
Hardware: research / the build / test it locally / present the goods and the bads

Designs:

- One thing we could think about is redesigning existing tools she already uses with Tobi Eye Gaze
- Typing tool (keypad) how a iPhone keyboard can be accessible to eye gazing user

To Jessica:

- We still need to understand what her exact issues were with the iPhone eye tracker


blackout sunlight
stand →
stable
on plain
camera.

📝 11/20/2025 - Jessica Feedback

Jessica Reply

Tools You Use With Tobii EyeGaze

Could you share which tools or apps you commonly use with your Tobii setup?
I use Tobii eye tracking settings, TD control, and Communicator 5

What keyboard or communication apps do you typically use?
I use Communicator 5 and the TD control keyword when I'm using the desktop.

Are there any specific features within those apps that are especially helpful or important to you?
Well, all apps are required, but in Communicator 5, you have many different communication boards to choose from, which is very helpful. For example, there is a symbol communication board, a keyboard communication board, and a dedicated area for saving and storing words, sentences, and paragraphs.

2. Your Experience With iPhone Eye Tracking
You mentioned previously that the iPhone's eye gaze feature was "very bad." We'd love to understand this better. Could you elaborate on:
Whether the problem was with the calibration process? Or if the issue was with using eye gaze itself, and if so, what exactly wasn't working well?
I can calibrate, but it still isn't accurate; for example, if I look at something on the phone screen, it goes to something else, not even close, or doesn't work at all.

Any details you're able to share will be incredibly helpful as we work on redesigning similar tools for iOS.
To be honest, all I care about is the accuracy. I can use any apps or tools as it is accurate. However, I have a preference for a communication app that features a keyboard communication board and a dedicated area for saving and storing words, sentences, and paragraphs. Also, it can be used when I'm recording videos, calling, FaceTiming, zooming, etc.

Reminders*
Remember to add external files to DAT Drive

📝 11/25/2025- General Class Notes

General Class Notes

- Jo: start with the CAD and putting the hardware components together
  - Solution for sunlight issue: flower for camera lenses? Get a mini to put on current camera
- Yi: complete wireframes to include enlarged keyboard designs
- Bev: start creating presentation slides outline and template
  - Include software research + report
  - Include Yi's research and findings and lead into presenting wireframe designs

---

Important Deadlines

- [ ] **Dec 2, 2025**
  - [ ] Have an idea of what we will be presenting for final presentation
  - [ ] Task delegation for slides (recording/ transcripts/ video compilation etc)
- [ ] **Dec 4, 2025**
  - [ ] Any clean-ups, wrap ups, preps for presentation (recordings)
- [ ] **Dec 9, 2025 - Final Presentation**
  - [ ] **DAT-F25-FinalGuide**
  - [ ] 7-10 minute presentation videos, with embedded captions (we recommend zoom and NYU Stream) and posted to your project website.
  - [ ] Videos must be uploaded to class webpage before 1pm, Dec. 9th
  - [ ] Following digital accessibility best practices, must include link to slides and video transcript

---

Milestones

14.

Weekly Class Deliverables (Due 12/2/25):

- [ ]
- [ ]

Additional Notes

- [ ]

Reminders*

Remember to add external files to DAT Drive

📝 12/2/2025- General Class Notes

General Class Notes/ Tasks

- Slides format for accessibility
  - Fonts: Montserrat
  - All normal texts in 12pt Normal;
  - Heading in 14pt Bold;
  - Sub Title in 36pt Black
  - Title in 68p Black
- Transcript Prepare
  - BEGIN WITH “Thanks x…” etc.
  - Include audio description of image in the slide in the transcript
  - END WITH “I’m going to pass it on to x” etc.
- Jo:
  - Fill in content for slides on Hardware and Next Steps sections (3-4min) **Due Dec 6, 2025**
  - Make sure our website is updated (weekly updates, front page, resource page). We will be graded on “Clarity and Accessibility of content posted to website” **Due Dec 8, 2025**
- Yi:
  - Fill in content for slides on User Research and iOS Designs sections (3-4min) **Due Dec 6, 2025**
  - Go over our final slides and make sure it satisfies ALL accessibility requirements **Due Dec 8, 2025**
- Bev:
  - Make our project page *better looking*? *(optional)*
  - Record sections (3-4min) **Due Dec 6, 2025**
  - Responsible for putting video recording and transcript together **Due Dec 8, 2025**

Designs:
https://www.figma.com/design/djyWW2wGiASZ8g0luf5ggn/Jessica-Tobii-User-Journey-Map?node-id=0-1&t=mcC2AFJjPt3dU0ah-1

Important links

➔ **Final Presentation Slides:** iSee - Final Presentation Slides
  - [x] ~~Does Central Question slide 6 answer Project Goals requirement?~~

➔ *Final Presentation Outline (Scroll Down)*

➔ Final Presentation Grading Rubric: DAT-F24-Final-Project-Rubric

➔ Final Transcript: Team Notes

Reminders*

Remember to add external files to DAT Drive

[DAT-F24-Final-Project-Rubric]

Important Deadlines

- [ ] **Dec 2, 2025**
  - [ ] Have an idea of what we will be presenting for final presentation
  - [ ] Task delegation for slides (recording/ transcripts/ video compilation etc)
- [ ] **Dec 5, 2025 night**
  - [ ] **ALL RECORDINGS, TRANSCRIPTS SHOULD BE IN THE DRIVE + SLIDES DONE**
  - [ ] **To-do (over the weekend):** Any clean-ups, wrap ups, preps for presentation
    - [x] ~~Yi: Alt text/ captions and visual aids/ Accessibility related touch-ups to slides~~
    - [x] ~~Bev: Compile recordings and transcripts for submission~~
    - [ ] **Due Dec 8, 2025**
- [ ] **Dec 9, 2025 - Final Presentation**

Milestones

15.

Weekly Class Deliverables (Due 12/16/25):

- [ ] DAT, F25 Self Assessment #2
- [ ] Final updates to website (front page, weekly updates, resources)
- [ ] Final version of video

Final Presentation Requirements (tldr)

- **DAT-F25-FinalGuide**
- 7-10 minute presentation videos, with embedded captions (we recommend zoom and NYU Stream) and posted to your project website.
- Videos must be uploaded to class webpage **before 1pm, Dec. 9th**
- Following digital accessibility best practices, must include link to slides and video transcript

Final Presentation Outline

1. Title Page / Introduction

Reminders*

Remember to add external files to DAT Drive

2. Overview of project (Understanding the Problem)
3. Project Goals
4. Competitive Analysis and Research
5. Design Process:
    a. Ideation and Sketching
    b. Prototyping
    c. User Testing and Iterations
6. Outcomes and Documentation
7. Recommendations
8. Next Steps
9. Literature Review / Research Sources (Links)

Reminders*

Remember to add external files to DAT Drive

📝 FINAL PRESENTATION FEEDBACK

Final Presentation Feedback

1.

Important Tasks

- [ ]

General Class Notes

16.

Additional Notes

- [ ]

Reminders*

Remember to add external files to DAT Drive

✅ Project Delegation + Timeline

Task Delegation

Divide the project into segments that each person handles to be stored here ?
Create a timeline to establish when certain tasks are done here or wherever convenient.

Thurs, Oct 30

DUE:

1. Continue researching/ scoping out what we did for midterms
2. Yi: Email Jessica about midterm updates (cc faculty and team!)

TODO:

1. All: Report back on a possible timeline of your own individual research for the next 7 weeks
2. Consolidate everyone's research and findings into a seven-week project timeline for the whole group.

Thurs, Oct 9

DUE:

3. Updates on research + communication with anyone (hardware / software / design spaces) = any leads??
4. Yi: Email Jessica + market research on existing similar prototypes / products

TODO:

3. Jo: some sort of 3D model done/ scope out things that need to be printed
4. Bev: a pretty good idea on what software works with what hardware + the technical feasibility of implementing with the timeline we have
5. Yi: User Journey + Competitive Analysis (market research)

Thurs, Oct 16

DUE:

1. Report on your TODOs from last week
2. Jo: coming up with model + drawing
3. Bev: look more in depth into software (green)
4. Yi: user journey + competitive analysis (market research)

TODO:

1. Edit Slides Here
2. Write script (2 mins per person) for your part

3. Record slides + Add your own captions + A transcript of your part (**DUE SATURDAY**)
    a. Midterm Instructions Slides **- CHECK THIS BEFORE RECORDING**
    b. END WITH "I'm going to pass it on to x" etc.
    c. Record through zoom with captions (QC IT FIRST)
4. Edit Recordings + Final Edits (**DUE SUNDAY**)
    a. Bev

Tues, Oct 21 ➡️ Midterm ‼️

Presentation Flow:

Yi - Interaction Design (2 mins)

- User Flow/ Pain points/ User Journey
- Competitive Analysis

Jo - Hardware (2 mins)

- Hardware concepts / cheapest options / CAD design of a prototype
- Timeline/ next steps for final prototype (hardware side)
- Speak about people you've reached out to + most recent updates

Bev - Software (2 mins)

- Software POC demo + next steps/ timeline + how software connects to hardware

Yi (2 mins)

- Finalize/ prioritize the user pain points that we want to focus on
    - Sunlight (1), Small icons (2) - high priority
    - 2-3 nice to haves

Bev then Jo (1 min each)

- Explain next steps on what our proof of concepts will resolve

---

Jo - Hardware

1. Hardware concepts / cheapest options / CAD design of a prototype
2. Reach out to relevant companies
    a. TobiEyeGaze
    b. Meta (Matthew - Accessibility Lead)
        i. Let's schedule a call all together after the Oct 9's meeting.
    c. Apple

Bev - Software

1. Software concepts + algo / existing options / scope of technical feasibility
2. Reach out to relevant professionals
    a. TobiEyeGaze

b. Apple Accessibility (Accessibility Support: https://support.apple.com/en-us/111749)

Yi - Interaction Design

1. Email Jessica and keep her up to date with our talk with faculty Oct 7, 2025
    a. Expectations on end deliverables
    b. Our midterm presentation date and what we plan on accomplishing from today till our midterm date
    c. Is she okay with that?
2. Reach out to the experts working on vision and user interface (Jessica cannot use devices outside in the sunlight: high contrast etc.)
3. Research/Reach out to the users who use eye gazing

Midterm Deliverables/ Approx Content We Will Cover

1. Who we've reached out to + who did not respond
2. What we learned from people (user, designers, stakeholders) we spoke to
3. What we learned from our research - hardware / software / interaction design

🎉 Final

LEAVE BLANK

Full Transcript

Full Transcript

Bev: Hi everyone, we're iSee. Our project focuses on creating accessible gaze-based technology for iPhone users with limited mobility. I'm Beverly and I lead software research and development. Yi focuses on user research and interaction design, and Jo on hardware.

Bev: Let me start by introducing our client.

Bev: Meet Jessica Frew - a model, actress, and disability advocate born with cerebral palsy that limits her upper-body mobility. Since age 9, she's used eye gaze technology to communicate and control devices. Jessica leads an incredibly active life, but struggles to find assistive technology that balances portability with precision.

Bev: Here's the problem: Jessica needs mobile eye gaze with desktop-level precision. The iPhone's built-in eye gaze has terrible tracking accuracy - making it essentially useless. Desktop solutions like Tobii work great, but they're bulky and impractical for mobile use. No existing solution bridges this gap.

Bev: We've identified two technical barriers. First, icons are too small - densely packed mobile screens make accurate selection impossible. Second, lighting challenges - current systems fail in sunlight, dim lighting, or changing conditions, losing calibration mid-use.

Bev: Our project aims to answer: Can we bridge the gap between mobile portability and professional-grade eye tracking by addressing both hardware limitations and interface design? Our approach consists of two parts - a physical hardware prototype to address lighting challenges, and interactive Figma wireframes with enlarged icons and optimized interfaces, plus a web prototype to validate UX concepts. To ground our design in real user needs, we began our comprehensive research. Yi will now walk us through our research findings.

Yi: Thanks Bev. I'll begin with our research.

Yi: Our first phase focused on user-centered insights, and you can see two images from our session with Jessica. One shows her eye-tracking control menu, a circular magnifier with tiny icons that appear when she focuses her gaze. The other shows her using a communication keyboard during a video call, typing entirely with her eyes. Meeting her grounded our project in reality. We saw challenges that rarely show up in documentation: calibration slipping under bright light, the struggle of selecting small icons, and the eye fatigue that builds with prolonged typing. She even shared her own idea for an iPhone setup connected through the bottom port.

Yi: We mapped a simple user flow showed on this slide to visualize how eye-only navigation moves through selecting, scrolling, and confirming.

Yi: In phase two, we conducted a competitive analysis of systems like Tobii and PCEye. They're reliable, but still face issues with outdoor lighting, calibration precision, and long-term comfort.

Yi: Phase three was a deeper dive into Jessica's specific concerns: larger icons, clearer feedback before selection, and reduced effort across screens. At the top of this slide is a WCAG contrast chart, paired with green check and red X icons that show how color and shape improve clarity.

Yi: We also tested high versus low contrast with the table shown on the slide. High contrast around seven-to-one significantly improved accuracy, while low contrast around three-to-one, especially in bright environments led to more mis-selections.Together, these phases created a foundation shaped not only by research but by lived experience, revealing what users truly need to feel confident and comfortable with eye-tracking.I'll now hand it to Jo for our prototype design.

Jo: Hi everyone, my process followed three main phases - ideation and sketching, prototyping the hardware and how to conduct user testing with iterative improvements.

Jo: Phase 1 - Ideation and sketching

Jo: A few key decisions guided the technical direction: I selected a global-shutter camera to avoid motion blur, which is essential for pupil detection. The Pi Zero 2W offered the best balance of size and computing power. After encountering OS issues with CSI cameras, I switched to a USB camera for reliability. Power stability was addressed by using a compact 5-volt power bank.

Jo: Phase 2 - Hardware Assembly

Jo: I assembled the Pi Zero and USB global-shutter camera, tested angles with a temporary rig, and confirmed a power bank works for early use.

Jo: The prototype image on this slide is a hardware mock-up showing how the camera mounts above the device, allowing early testing of angle, field of view, and user comfort. After trying multiple enclosure ideas, I chose an adjustable car-phone holder for stable, flexible positioning.

Jo: Phase 3 - User Testing and Ideation

Jo: Because the project involves collecting eye images, which qualify as identifiable biometric data, user testing must be reviewed by the (IRB).

Jo: Post IRB: To evaluate performance, I would conduct informal testing with peers. I'd measure image clarity, glare resistance, and pupil-tracking stability indoors and outdoors. Note that Jessica highlighted that outdoor lighting was the biggest issue with the Tobii Eye-gaze tracker– I believe the sunlight overwhelmed the sensor and reduced contrast.

Jo: A major achievement of this project was developing detailed CAD models that guided the hardware design. I modeled the full eye-tracking assembly—camera, Pi Zero housing, and the mounting system—and created a compact clip-on phone mount that positions the camera at an optimal angle.

Jo: Future CAD iterations will integrate optical elements like a flower-style lens hood to support outdoor performance.

Jo: Beverly will now go over the software research.

Bev: Thanks Jo for walking us through the hardware prototype component. Now I'll cover our software exploration, which happened in three phases.

Bev: Phase one - we built a browser-based prototype using WebGazer.js to simulate gaze interaction. This includes calibration, dwell selection, feedback. This validated our core UX patterns and proved conceptual feasibility.

Bev: Phase two - we researched building a custom iPhone app with system-wide control and improved calibration. But we hit major roadblocks - iOS sandboxing blocks third-party system control, and TrueDepth camera access is heavily restricted. Apple's built-in Eye Tracking seems to be our only option, but Jessica confirmed that it doesn't meet her needs when we reached out to her again.

Bev: This brings us to Phase three, our conclusion - third-party iOS control is technically impossible. But this research was valuable. With it, we documented viable paths for Jessica and validated interaction patterns. The key takeaway from this is that software alone can't solve her precision needs, which reinforces our hardware prototype approach and directs our focus toward designing a better interface for Jessica. Our web prototype from Phase 1 still remains useful for demonstrating interaction logic. Yi will now walk you through our iOS app designs and how we optimized them for eye gaze interaction.

Yi: Thanks Bev. We began shaping our iOS app.

Yi: In ideation, we built a design-principles table grounded in Jessica's concerns and WCAG guidelines. That led us to a few key choices: larger target sizes, high-contrast visuals around seven to one with bright icons on a dark background, and a clear two-step gaze interaction, where users first highlight and then confirm to avoid accidental selections.

Yi: In visual design, we chose a black background with white text to reduce eye fatigue, especially in bright environments. The blue halo became our gaze indicator because it's high-contrast, accessible for users with color-vision differences, and already familiar in iOS as a focused state.

Yi: On this slide, you see how feedback works. A soft blue halo appears when the user begins to gaze, and it gradually shrinks over about a second to confirm the selection.

Yi: Our lo-fi prototype walks through that flow shown on the slide: gazing at the Message button, entering the inbox, and using the magnifier to zoom into a conversation with enlarged text for easier reading.

Yi: When we shared this with Jessica, she emphasized how much she depends on symbol boards, keyboard boards, and saved phrases.

Yi: In our final prototype, you can see that reflected in the typing experience, you'll see in the video.

Yi: Hello everyone, this is the iSEE iOS App Design Demo. Our focus is making eye-gaze interaction easy, accurate, and comfortable for users. In this demo, you'll see our simplified home screen with three to five main features, and we'll walk through the messaging flow with the communication keyboard.

When the user opens the iPhone and gazes at the Message icon, a soft blue halo appears. Holding the gaze for about 600 to 1000 milliseconds confirms the selection, and the user enters the Messages app. From there, they can gaze at the Magnifier button to zoom in and read message details more clearly.

Inside a conversation, gazing at the text input box triggers the communication keyboard. Users can type using enlarged, easy-to-hit keys and then select from suggested phrases that appear above the keyboard. They can also choose from categorized phrase boards, allowing quick access to customized and frequently used expressions.

Once a phrase is selected, it fills into the text box, and the user returns to the dialog view to send or continue the conversation.

Yi: Now, I'm moving to Jo for the next steps.

Jo: To finish off, I will be going over suggested next steps. These include: calibration + gaze mapping, IR illumination, enclosure refinement, algorithm improvements, user interaction testing, and open-source preparations.

Jo: The team and I would love to thank you for listening to our presentation and we hope to see future developments.

Yi Final Transcript

Yi Final Transcript

User Research

Yi: Thanks Bev. I'll begin with our research.

Phase 1: User centered research

Yi: Our first phase focused on user-centered insights, and you can see two images from our session with Jessica. One shows her eye-tracking control menu, a circular magnifier with tiny icons that appear when she focuses her gaze. The other shows her using a communication keyboard during a video call, typing entirely with her eyes. Meeting her grounded our project in reality. We saw challenges that rarely show up in documentation: calibration slipping under bright light, the struggle of selecting small icons, and the eye fatigue that builds with prolonged typing. She even shared her own idea for an iPhone setup connected through the bottom port.

User Flow

Yi: We mapped a simple user flow showed on this slide to visualize how eye-only navigation moves through selecting, scrolling, and confirming.

Phase 2: Competitive Analysis

Yi: In phase two, we conducted a competitive analysis of systems like Tobii and PCEye. They're reliable, but still face issues with outdoor lighting, calibration precision, and long-term comfort.

Phase 3: Deep-Dive Research

Yi: Phase three was a deeper dive into Jessica's specific concerns: larger icons, clearer feedback before selection, and reduced effort across screens. At the top of this slide is a WCAG contrast chart, paired with green check and red X icons that show how color and shape improve clarity.

Testing High vs Low contrast

Yi: We also tested high versus low contrast with the table shown on the slide. High contrast around seven-to-one significantly improved accuracy, while low contrast around three-to-one, especially in bright environments led to more mis-selections.Together, these phases created a foundation shaped not only by research but by lived experience, revealing what users truly need to feel confident and comfortable with eye-tracking.I'll now hand it to Jo for our prototype design.

iOS App Design

Yi: Thanks Bev. We began shaping our iOS app.

Ideation

Yi: In ideation, we built a design-principles table grounded in Jessica's concerns and WCAG guidelines. That led us to a few key choices: larger target sizes, high-contrast visuals around seven to one with bright icons on a dark background, and a clear two-step gaze interaction, where users first highlight and then confirm to avoid accidental selections.

Visual Design

Yi: In visual design, we chose a black background with white text to reduce eye fatigue, especially in bright environments. The blue halo became our gaze indicator because it's high-contrast, accessible for users with color-vision differences, and already familiar in iOS as a focused state.

Visual feedback example

Yi: On this slide, you see how feedback works. A soft blue halo appears when the user begins to gaze, and it gradually shrinks over about a second to confirm the selection.

Lo-fi Prototype

Yi: Our lo-fi prototype walks through that flow shown on the slide: gazing at the Message button, entering the inbox, and using the magnifier to zoom into a conversation with enlarged text for easier reading.

User Feedback

Yi: When we shared this with Jessica, she emphasized how much she depends on symbol boards, keyboard boards, and saved phrases.

Final Prototype

Yi: In our final prototype, you can see that reflected in the typing experience, you'll see in the video.

Video Transcript:

Hello everyone, this is the iSEE iOS App Design Demo. Our focus is making eye-gaze interaction easy, accurate, and comfortable for users. In this demo, you'll see our simplified home screen with three to five main features, and we'll walk through the messaging flow with the communication keyboard.

When the user opens the iPhone and gazes at the Message icon, a soft blue halo appears. Holding the gaze for about 600 to 1000 milliseconds confirms the selection, and the user enters the Messages app. From there, they can gaze at the Magnifier button to zoom in and read message details more clearly.

Inside a conversation, gazing at the text input box triggers the communication keyboard. Users can type using enlarged, easy-to-hit keys and then select from suggested phrases that appear above the keyboard. They can also choose from categorized phrase boards, allowing quick access to customized and frequently used expressions.

Once a phrase is selected, it fills into the text box, and the user returns to the dialog view to send or continue the conversation.

Yi: Now, I’m moving to Jo for the next steps.

Jo's Transcript

Hardware Prototypes

Intro
Jo: Hi everyone, my process followed three main phases: ideation and sketching, prototyping the hardware and how to conduct user testing with iterative improvements.

***Jo** : Phase 1- Ideation and sketching*

Jo: A few key decisions guided the technical direction:
I selected a global-shutter camera to avoid motion blur, which is essential for pupil detection. The Pi Zero 2W offered the best balance of size and computing power. After encountering OS issues with CSI cameras, I switched to a USB camera for reliability. Power stability was addressed by using a compact 5-volt power bank.

Jo:*Phase 2- Hardware Assembly*

Jo: I assembled the Pi Zero and USB global-shutter camera, tested angles with a temporary rig, and confirmed a power bank works for early use.

***Jo** : Prototype*

Jo : The prototype image on this slide is a hardware mock-up showing how the camera mounts above the device, allowing early testing of angle, field of view, and user comfort. After trying multiple enclosure ideas, I chose an adjustable car-phone holder for stable, flexible positioning.

Jo: *Phase 3 - User Testing and Ideation*

Jo:Because the project involves collecting **eye images**, which qualify as identifiable biometric data, user testing must be reviewed by the **Institutional Review Board (IRB)**.

Jo: Post IRB: To evaluate performance, I would conduct informal testing with peers. I'd measure image clarity, glare resistance, and pupil-tracking stability indoors and outdoors. Note that Jessica highlighted that outdoor lighting was the biggest issue with the Tobii Eye-gaze tracker– I believe the sunlight overwhelmed the sensor and reduced contrast.

Achievements slide

Jo : A major achievement of this project was developing detailed CAD models that guided the hardware design. I modeled the full eye-tracking assembly—camera, Pi Zero housing, and the mounting system—and created a compact clip-on phone mount that positions the camera at an optimal angle.

Jo: Future CAD iterations will integrate optical elements like a flower-style lens hood to support outdoor performance.

Jo: Beverly will now go over the software research.

Next Steps

Jo: Our suggested next steps including :

- Calibration + gaze mapping
- IR illumination
- Enclosure refinement
- Algorithm improvements
- User interaction testing
  And Open-source prep

Thank you

Jo :The team and I would like to thank you for listening to our presentation and we hope to see future developments.

💻 Blog Posts

LEAVE BLANK

#2

Weekly Update 2 Yi Wang (NYU)

What is commercially available to solve the problem?

Describe at least 3 potential approaches (either commercially available products or DIY tutorials) that could be used by your partner.
Include pictures and URLs for the products you found. For each, describe what differentiates your solution.
1) Tobii Dynavox PCEye (Windows add-on eye tracker). Lets users control a Windows device with their eyes; marketed as usable "indoors and outdoors," ships with TD Control for Windows navigation.
URL:https://www.tobiidynavox.com/products/pceye?variant=40160250429632

2) Tobii Dynavox TD I-Series (all-in-one AAC with outdoor eye tracking). Integrated Windows tablet + tracker; specifically advertises bright-sunlight tracking; includes TD Control/TD Browse/TD Phone.
URL: https://www.tobiidynavox.com/products/td-i-series

Differences:
Sunlight kit (hardware + software): lightweight **clip-on sun hood** for the screen (avoids IR-blocking films) + **auto high-contrast mode** and **thicker focus outlines** when ambient light spikes. (Commercial devices claim outdoor support; we tailor it to Jessica's setup with minimal hardware and no device swap.)

What research has been done in similar domains?

Briefly describe at least 3 related research papers from the academic field most relevant to your project (for example, Occupational Therapy, Rehab Science, Human-Computer Interaction, Engineering, …)
Include citations and link to resources page
Reflect on the impact of this solution

Human-Computer Interaction:

1. **Bubble Cursor (CHI 2005).** Dynamically enlarges the effective activation area to the nearest target, significantly improving selection speed/accuracy—directly informs our "Gaze Bubble." https://www.dgp.toronto.edu/~ravin/papers/chi2005_bubblecursor.pdf
2. **MAGIC Pointing (CHI 1999).** Combines gaze to "warp" the cursor near the intended target and hand/switch for fine control; inspires our optional "confirm-to-act" and reduced dwell time on tiny icons. https://research.ibm.com/publications/manual-and-gaze-input-cascaded-magic-pointing
3. **Evaluation of Eye Gaze Interaction (CHI 2000).** Early controlled studies showing gaze interaction can be fast but needs strategies to avoid unintended activation—supports our pause/confirm patterns. https://www.cs.tufts.edu/~jacob/papers/chi00.sibert.pdf

   These findings suggest target-expansion + cascaded confirmation can offset eye-tracker error, especially under glare or fatigue, while keeping interactions quick. "Gaze Bubble + confirm" directly operationalizes this.

Engineering?

Josephine Odusanya Beverly Yip

1. **WebGazer.js**
2. **GazeRecorder**: Proprietary but allows local experiments; captures eye-movement data for webpage analytics.

3. **EyeJS / GazePointer (community forks):** WebGazer-like projects using TensorFlow.js; more modern architectures but less documented.
4.

What is the relationship between your proposed solution and the current technologies your partner uses?

Stays on Jessica's existing Tobii-on-Windows workflow. Develop her user experience with Tobii phone use.

How will your group ensure that your solution won't interfere with your partner's current daily activities?

Our solution builds directly on Jessica's existing Tobii Eye Tracker setup and Windows-based control environment, ensuring compatibility and minimal disruption.

Software: Beverly Yip
We are testing eye-tracking performance under harsh sunlight to address the calibration and visibility issues Jessica experiences outdoors. A gaze-interaction simulation tool allows us to prototype adjustable target sizes and observe how different visual contrasts affect usability.

Hardware: Josephine Odusanya
Through integration and calibration tests, we will fine-tune how the overlay and hardware assist work together.

User Testing & Validation:
Jessica will participate in user testing sessions to validate comfort, calibration accuracy, and task efficiency. Her feedback will directly inform design refinements, ensuring that our system complements her daily workflow rather than replacing it.

How will your group design a solution so it doesn't interfere with other devices in your partner's environment?

🧑‍💻 Software Notes

Last Updated: 10/7/2025

🧠 SOFTWARE OPTIONS — Filtered for the Jo's Hardware

1️⃣ OpenMV (MicroPython-based)

For: OpenMV Cam H7 / H7 Plus / H7 R2 (or the listed $65+ OpenMV Cams)

Why relevant:
OpenMV runs **MicroPython** and supports **real-time computer vision** on-board (no external phone CPU needed). Perfect for fast prototyping.

Capabilities:

- Built-in functions for **frame differencing, blob detection, circle detection, eye tracking prototypes**.
- Can detect **dark pupil / light glint** using IR illumination.
- You can send detected coordinates over **UART / I²C / USB / Wi-Fi (ESP32 bridge)** to a phone app.
- Has an **IDE with live video view + script upload**, great for iteration.

You can then stream `(x, y)` gaze data via USB serial to your phone or ESP32.

Feasibility:
✅ Great for proof-of-concept
⚠️ Limited by onboard processing (QVGA max FPS ≈ 25–60)
⚠️ Accuracy modest, no built-in calibration mapping — you'll need to code calibration on the phone side.

2️⃣ ESP32-S3 (with OpenCV or TensorFlow Lite Micro)

For: Microcontroller-centric pipeline with USB camera

Why relevant:
ESP32-S3 can process **small ML models** or **edge computer vision tasks**, especially with IR-filtered grayscale frames.
It can handle simple **pupil detection, thresholding, or blob finding** onboard and pass reduced data to the phone.

Possible stack:

- Use **ESP-IDF** or **Arduino** environment.
- Capture frames via **USB/UVC camera (if supported)** or attach camera via **I2C / SPI (OV2640)**.
- Perform thresholding / contouring to find pupil center.
- Use **TFLite Micro** model (tiny CNN) for pupil localization.
- Send gaze vector data to phone via USB serial or Bluetooth.

Why it matters:
This keeps heavy image processing off the phone, while the phone runs calibration + UI logic.

Feasibility:
✅ Viable for early prototype
⚠️ Compute limited — can't run full gaze mapping or deep ML
✅ Excellent for "bridge" mode (capture + basic preprocessing)

3️⃣ Jetson Nano / IMX219 Binocular Camera

For: Highest fidelity gaze tracking pipeline (Linux environment)

Why relevant:
The Jetson Nano can run **Pupil Labs software** or **custom OpenCV/TensorFlow gaze models**.
Since it's Linux-based, it's compatible with almost every open eye-tracking library.

Open-source software compatible:

Software	How to Use It	Why It Fits Jetson
Pupil Core (Pupil Labs)	Install on Jetson Nano (Ubuntu ARM) and connect IMX219 camera. Use pupil detection plugin for IR video.	Works out of the box; highly accurate; easy to modify.
OpenIris	Can be compiled under Linux/ARM. Use one camera stream.	Modular pipeline; allows writing custom plugin to output gaze over serial/WebSocket.

OpenCV + custom scripts	Write your own cv2.HoughCircles() or threshold/ellipse fit for pupil.	Lightweight, full control over code, can be ported later to MCU or phone.
DeepGaze (Python)	Lightweight CNN-based gaze estimator (torch).	May run slowly but gives insight into ML-based gaze models.

Typical Jetson pipeline:

```
Camera → OpenCV (pupil detect) → calibration mapping (Python)
    → WebSocket / USB stream → Android app
```

Feasibility:
✅ Highest accuracy
✅ Runs any open-source package you choose
⚠️ Power-hungry, larger footprint, less portable than MCU setup
✅ Ideal "reference platform" before miniaturizing

4️⃣ WebGazer.js (Browser Prototype)

Even if you later offload to Jetson/ESP32, **WebGazer.js** is still valuable for:

- Quick **UI / calibration testing**
- Browser-based data visualization
- Early-stage UX validation before hardware is ready

You can later feed it real gaze data via WebSocket for simulated sessions.

🧩 Recommended Software / Hardware Pairings

Hardware Option	Recommended Software Stack	Role / Benefit
OpenMV Cam	OpenMV IDE (MicroPython) + UART to phone	Easiest on-board detection prototype

ESP32-S3	ESP-IDF + OpenCV-lite or TensorFlow Lite Micro	Efficient bridge; preprocessing before phone
Jetson Nano + IMX219	Pupil Core / OpenCV Python / OpenIris	Full-featured development + data accuracy baseline
Phone-only (fallback)	WebGazer.js / Android app + OpenCV	Low hardware dependence; fastest iteration

---

⚙️ Integration Plan (Step-by-Step)

1. **Phase 1: Jetson Prototype (Reference System)**
   - Connect IMX219 binocular camera to Jetson Nano.
   - Run **Pupil Core** or **OpenCV-based pupil detection**.
   - Log gaze vectors and calibration mapping accuracy.
   - Use results as baseline for smaller MCU tests.
2. **Phase 2: OpenMV / ESP32-S3 Feasibility**
   - Port a simple pupil detection script.
   - Stream (x, y) gaze center over UART to a phone mock app.
   - Measure latency and stability.
3. **Phase 3: Android Integration**
   - Use **Kotlin app** or **WebView** to display gaze overlay.
   - Integrate serial input from ESP32 or Jetson.
   - Test calibration, dwell selection, and speech output.
4. **Phase 4: Optimization + Enclosure**
   - Once software works, finalize IR ring brightness, clip alignment, and camera FOV.
   - Refine gaze smoothing and drift correction.

Tasks to figure out

1. **Evaluate Pupil Core on Jetson**

   → Confirm IR camera works and track gaze precision.

2. **Prototype Pupil Detection with OpenCV (Python)**

   → Test different thresholding / circle detection on sample IR frames.

3. **Test OpenMV "blob detection" script**

   → Validate if the small MCU can reliably output pupil coordinates.

4. **ESP32-S3 Micro test**

   → Check if TensorFlow Lite Micro can run a lightweight CNN for pupil detection.

5. **Android WebGazer.js prototype**

   → Simulate calibration and UI logic using synthetic data.

🔍 Conclusion

Goal	Recommended Path
Fastest working prototype	OpenMV + IR ring + phone serial viewer
Most accurate development baseline	Jetson Nano + Pupil Core
Most scalable long-term	ESP32-S3 + optimized OpenCV-lite / TF-Lite Micro
Simplest UX mock	WebGazer.js in phone webview

Midterm Prep Breakdown

Last Updated: 10/14/2025

TIMELINE

Phase Focus (Midterm) → Simplest UX Mock

- Purpose: Demonstrate **conceptual and interaction feasibility**.
- Deliverable: A browser or Android-based prototype simulating gaze input + calibration flow.
- This includes:
    - Existing software concepts + algorithms behind gaze tracking.
    - What's achievable with open-source tools (WebGazer.js, GazeRecorder, etc.).
    - Limitations and where hardware integration would eventually fit in.

Phase Focus (End of Semester) → Fastest Working Prototype

- Purpose: Turn the UX mock's logic into a minimal working device with **OpenMV Cam + phone serial viewer**.
- Deliverable: Proof-of-concept that detects gaze direction and sends coordinates to phone.

Simplest UX Mock: Software Concepts + Feasibility

1️⃣Software Foundation: Web-Based Gaze Tracking

WebGazer.js (main candidate for mockup phase)

- **Core concept:** Uses the laptop/phone's front camera and maps eye movements to screen coordinates using regression models trained in-browser.
- **Algorithm pipeline:**
    - Face detection (via facial landmark model, e.g., *clmtrackr*).
    - Eye region extraction.
    - Pupil center estimation (based on intensity + shape).
    - Screen calibration (maps eye image → screen coordinate via linear regression).
- **Easeability:**
    - Works entirely in the browser (no installation, no backend).
    - You can simulate "gaze pointer" interactions on a webpage (e.g., hover → highlight, dwell → click).
    - Built-in calibration flow using click points.

- **Feasibility:** Works best under good lighting and fixed head position.
- **Scope of customization:**
    - You can modify calibration mapping logic, dwell time, or gaze smoothing.
    - You can replace camera feed later with incoming (x, y) from external device via WebSocket.

🔗 Example:

```
WebGazer.setRegression('ridge')
  .setGazeListener((data, elapsedTime) => {
    if (data) {
      moveCursor(data.x, data.y);
    }
  })
  .begin();
```

2️⃣ Optional Comparison: GazeRecorder / EyeJS

- **GazeRecorder**: Proprietary but allows local experiments; captures eye-movement data for webpage analytics.
- **EyeJS / GazePointer (community forks):** WebGazer-like projects using TensorFlow.js; more modern architectures but less documented.

3️⃣ UX Simulation Plan

Create a **browser prototype** mimicking your client's daily use case:

- **UI elements:** Home screen → basic communication board / dwell-select buttons.
- **Simulated input:** Cursor moves where eyes "look" (via WebGazer).
- **Feedback:** Visual highlight + optional speech output using `speechSynthesis` API.
- **Calibration flow:** Tap 5x5 grid of 25 dots to map gaze model.

4️⃣ Technical Feasibility: What's Real vs Simulated

Feature	Achievable (WebGazer)	Needs Real Hardware
Gaze pointer (cursor)	✅	
Dwell-based clicking	✅	
Calibration mapping	✅	
IR illumination tracking	❌	✅
Pupil glint-based precision	❌	✅
Low-light performance	⚠️	✅ (IR-based)

Linking to the Final (Fastest Working Prototype)

Transition Plan:

Midterm (Web Mock)	Final (Hardware Prototype)
Gaze coordinates simulated from webcam	Real coordinates from OpenMV pupil detection
Web UI shows gaze-controlled actions	Phone serial viewer or Android app
Web calibration + dwell logic tested	Calibration + smoothing ported to app
No hardware dependency	Uses IR ring + camera clip

Next Steps (After Midterm)

1️⃣ Validate UX Flow

- Test how intuitive dwell selection and calibration are with users (use mock).
- Refine timing, feedback, or visual cues before hardware integration.

2️⃣ **Prototype OpenMV Script**

- Test basic `blob_detection` to locate pupil center in IR-lit frames.
- Stream coordinates via UART to your laptop/phone mock for testing.

3️⃣ **Bridge Software**

- Replace WebGazer input with serial gaze data from OpenMV.
- Maintain same UI → ensures continuity from simulation to physical build.

4️⃣ **Integrate Hardware**

- Combine IR LED ring, OpenMV Cam, and phone interface into a compact clip.
- Run same gaze interaction logic with live data.

Midterm Presentation Outline (Software Section)

1. Problem Context *(Link to Jessica's hardware + research challenge)*

- **Challenge goal**: Recreate *core gaze-interaction behaviors* (tracking, calibration, selection) in a lightweight browser-based prototype to support hardware design.
- **Software role**: Act as a low-cost sandbox for experimenting with gaze-based user experience before hardware integration.

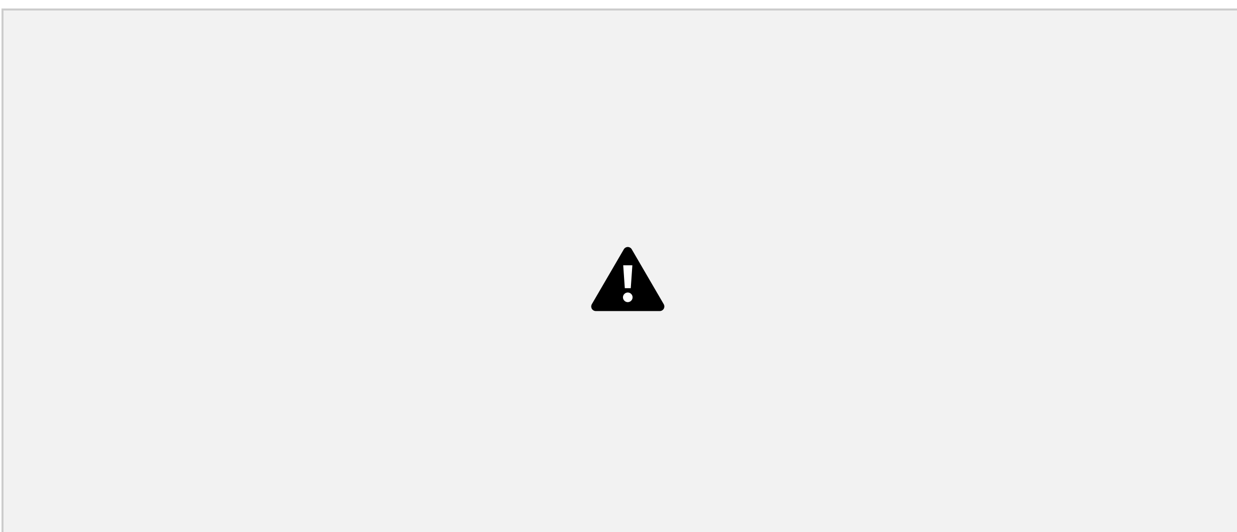
Elaboration:
→ Mention software simulates the user experience that hardware would enable, giving your team rapid iteration without needing the full device.
→ Show a <u>short clip or live demo</u> of prototype at this point to contextualize.

---

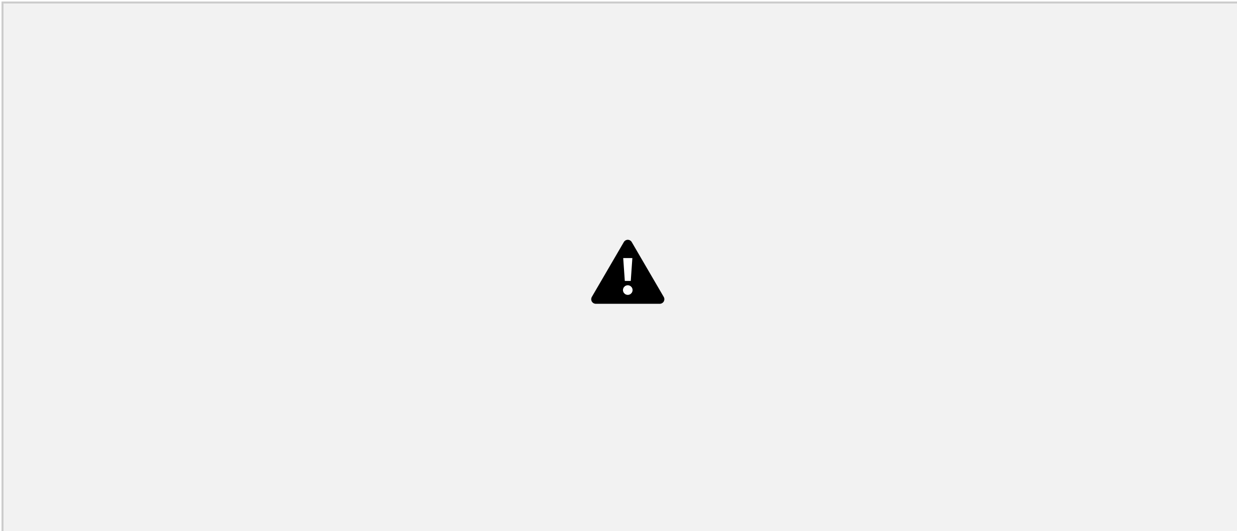
2. Simplest UX Mock → Software Overview

- **Prototype stages:**
    1. **View 1:** Webcam & gaze tracking setup.

2. **View 2:** Calibration through 5x5 red-dot following.



3. **View 3:** Interactive gaze-based reading (blue circles → dwell → voice feedback).



- **Goal:** Model the “eyes as a controller” interaction loop: *camera → gaze → action → audio feedback*.

Elaboration:
→ Emphasize how the experience scaffolds the user — from onboarding to calibration to action — mimicking what a real gaze interface would feel like.
→ Highlight your design decisions (e.g., dwell time threshold, audio and visual feedback, confetti for completion).

---

3. WebGazer.js Concept + Algorithm Visualization

- **How it works:**
  - WebGazer.js uses facial landmark detection + ridge regression to predict gaze coordinates.
  - Continuous training improves accuracy over time as the user interacts.

- **Visual aid:**
  - Show diagram or animation of: *webcam → eye landmarks → regression mapping → screen coordinates*.

- **Calibration as a training loop:**
  - Each red dot followed helps train the model to match real eye positions to screen space.

Elaboration:
→ Mention how this model learns per-user differences (like head tilt, lighting, etc.).
→ Connect to your teammate's research on human factors — accuracy, fatigue, and accessibility concerns.

---

4. Proposed Mockup Flow (Calibration → Dwell Select → Output)

- **Flow overview:**
  - **Calibration:** User tracks moving red dot.
  - **Dwell Selection:** User gazes at blue word circles for 1.5s to "click."
  - **Output:** Visual + audio response (word turns green + spoken aloud).
  - **Completion feedback:** Confetti animation on task completion.

- **User feedback loops:**
  - Visual feedback = stability + confidence.
  - Auditory feedback = accessibility reinforcement.

Elaboration:
→ Explain that this flow parallels real gaze UI patterns in assistive devices (like Tobii Dynavox).
→ Mention how you intentionally structured timing and colors for clarity and accessibility.

---

5. Feasibility Table (What's Simulated vs. Real)

Feature	Current Implementation	Real-World Equivalent	Notes
Eye Tracking	WebGazer.js (browser-based regression)	IR-based pupil tracking (Tobii)	Simulates user behavior for testing UI logic
Calibration	Visual dot-following	Infrared gaze calibration	Approximates learning process

Dwell Detection	Timed fixation (2s)	Continuous gaze stability detection	Tunable dwell threshold
Speech Feedback	Web Speech API (TTS)	Synthesized or recorded voice	Adds accessibility + UX reinforcement
Confetti Animation	JS simulation	Haptic / LED feedback	Represents multimodal reward signal

Elaboration:
→ Emphasize what's *conceptually validated* vs. what still depends on physical hardware (sensor fidelity, latency, etc.).

---

6. Technical Scope → What Code/Modules to Test

- **Modules implemented:**
  - WebGazer initialization & calibration
  - Gaze smoothing algorithm (moving average)
  - Dwell detection logic
  - Speech feedback loop
  - Dynamic DOM updates for word color change + confetti trigger
- **Test metrics:**
  - Gaze stability (smoothness)
  - Dwell accuracy (false positive rate)
  - Speech feedback timing
- **Integration potential:**
  - Replace WebGazer.js with hardware gaze data (e.g., Tobii SDK or custom IR camera).

Elaboration:
→ Note how modular structure (HTML/CSS/JS separation) will make future integration with Jessica's device data straightforward.

7. Next Steps Toward Working Prototype

- **Short-term:**
  - Refine smoothing filters to reduce jitter.
  - Add calibration quality indicator.
  - Optimize layout for phone/ mobile screen sizes.

- **Medium-term:**
  - Replace simulated WebGazer input with hardware gaze data stream.
  - Sync speech + motion outputs with real device feedback.

- **Long-term vision:**
  - Create a full gaze-controlled reading or selection interface integrated with physical hardware (Jessica’s module).

Elaboration:
→ The bridge between mock software simulation and the future (final) integrated prototype.

Midterm Presentation Transcript

Beverly: Thanks Jo for the insight on the hardware components. Building on her work on hardware fabrication, my software prototype explores how users interact with the system — from calibration to gaze-based reading and feedback.

Beverly: My goal was to create a lightweight, browser-based eye-tracking solution for mobile, letting us test gaze interaction before hardware integration.

Beverly: The prototype has three screens:

First, users set up their webcam to lock onto & track their facial features.

Second, they calibrate by following a 5x5 grid of red dots, holding each gaze for 3 seconds, ensuring accuracy and stability.

Third, users gaze at blue word circles. Holding their gaze 1–2 seconds simulates using a keyboard with eye-gaze technology: word turns green, dot disappears, the word you successfully selected gets read aloud — giving users audio and visual feedback.

Beverly: The prototype uses WebGazer.js: video captured → eye landmarks detected → coordinates smoothed → dwell detection triggers feedback. The calibration section trains the model, simulating real gaze-controlled interfaces. Smoothing is also added to reduce jitter for accurate tracking.

Beverly: Currently, this simulates gaze in the browser. Our next steps include: connecting to hardware like ESP32, OpenMV, or Jetson, optimize calibration, and to test this on phones.

Next, I'll pass it to Yi, who will discuss prioritizing user pain points.

Ending Part

Beverly: Building on the pain points we identified, our software and hardware prototypes tackle these challenges by simulating eye-tracking on a phone.

For harsh sunlight, the software tests gaze accuracy under different lighting. For small mobile icons, it simulates adjustable target sizes. This lets us refine circle size, spacing, and dwell duration, ensuring users can reliably select items with their eyes in a controlled, low-risk environment before integrating hardware.

Report: iOS Feasibility

Task: A high-level research report on the software and hardware restrictions of the iPhone/iOS/Apple ecosystem

Research Report: Feasibility of Third-Party System-Wide Eye-Gaze Control Application on iOS

Key Findings: iOS and Third-Party App Restrictions

The development of a third-party application with system-wide control capabilities is fundamentally constrained by Apple's core security and privacy architecture.

1. iOS Security Model: Sandboxing and System Control

The most significant barrier is the iOS Sandboxing model.

- **Principle of Isolation:** Every third-party application on iOS operates within a secure, isolated environment (a "sandbox"). This prevents an app from accessing the data, memory, or files of other applications or the operating system itself.
- **System-Wide Control Prohibition:** A direct consequence of sandboxing is the prohibition of system-wide control. A third-party app cannot inject input, simulate taps, or select elements outside of its own application window. This means an app cannot be developed to "select elements in her iPhone to navigate it" across the entire OS (e.g., opening Settings, interacting with a third-party messaging app, or navigating the Home Screen).
- **Accessibility API Limitations:** Although Apple offers strong Accessibility APIs (like UIAccessibility) that let apps share their interface elements with built-in tools such as VoiceOver or Switch Control, these APIs only allow apps to *receive* input from those tools — not to *control* the system itself. In addition, third-party apps aren't allowed to create new accessibility features that work across the entire system the way Apple's own tools do.

2. Hardware Access: TrueDepth Camera and Eye-Tracking Data

Accurate eye-gaze technology requires high-fidelity data, often sourced from the TrueDepth camera system (which includes an infrared projector and sensor).

- **Limited TrueDepth Access:** Third-party developers are granted limited access to the TrueDepth camera via the ARKit framework. This access is sufficient for augmented reality features (like face tracking and gross head/eye position) but is not designed for the high-precision, low-latency eye-gaze tracking required for reliable system navigation.

- **Privacy and Security:** Apple strictly controls access to the raw, high-resolution depth and infrared data necessary for medical-grade eye-tracking. This data is highly sensitive and is primarily reserved for system functions like Face ID and Apple's own built-in accessibility features. This restriction is a deliberate privacy measure, making it impossible for a third-party app to achieve the necessary accuracy and reliability.
- **Alternative (Front-Facing Camera)**: While the standard front-facing camera can be used for eye-tracking, it lacks the infrared and depth data of the TrueDepth system, leading to the "really bad" performance your client has likely experienced with existing third-party solutions.

3. Apple's Built-in Solution: Eye Tracking (iOS 18+)

Apple has recognized the need for this technology and has developed its own solution, which highlights the difficulty for third parties.

- **System-Level Integration:** Apple's new Eye Tracking feature (introduced in iOS 18/iPadOS 18) uses the front-facing camera and on-device machine learning to provide system-wide navigation and control.
- **Implication for Third Parties:** The existence of this feature confirms that the required level of system integration and hardware access is only possible for Apple itself, as it can bypass the sandboxing and API restrictions imposed on third-party developers. Any third-party attempt to replicate this functionality would be immediately rejected by the App Store review process for violating security and privacy guidelines.

Conclusion and Timeframe Justification

Based on the technical and policy restrictions of the iOS platform, the goal of developing a feasible working app that provides specialized control to use eye gaze and select elements in Jessica's iPhone to navigate it is not achievable by a third-party developer.

The core limitation is the **iOS Sandboxing model**, which prevents any application from controlling the operating system outside of its own window. Furthermore, the necessary high-fidelity hardware access (TrueDepth camera data) is restricted for privacy and security reasons.

Possible Alternatives To Consider:

- **Replicating System Control:** To achieve the desired system-wide control, a developer would need to find an undocumented exploit or a private API, which is a violation of App Store guidelines and would result in immediate rejection. Even if a proof-of-concept were developed, it would be *unstable* and *non-distributable*.
- **Developing a Custom Solution:** Developing a custom eye-tracking solution that is limited to a single app (which is the only thing a third-party app can do) would still

require significant time for machine learning model training, calibration, and UI/UX design, making a one-month delivery of a high-quality product *highly unlikely*.

Recommendation:
The most viable and stable solution is to **utilize Apple's built-in Eye Tracking feature** (available on devices running iOS 18 or later). This feature provides the system-wide control and integration that a third-party app is technically forbidden from achieving.

But Jessica has already specified how *bad* the feature is for her use case. Therefore, while Apple's built-in Eye Tracking feature is the only technically feasible option for achieving system-wide control, its current implementation still *falls short* of meeting Jessica's specific accessibility needs.

🪛 Hardware Notes

Challenges & constraints for a phone-based eye tracker

Here are things to keep in mind when trying to use a small camera with a phone (Android or iOS) for gaze tracking:

Constraint	Implication / what to watch out for
Interface / connectivity	Phones usually interface with cameras via built-in modules (on the mainboard). To use an *external* camera, you often need **USB OTG / USB host** support, or use something like a *USB camera module* that is UVC compliant.
Power & voltage	The external camera must be powered from phone's USB OTG port (5 V, limited current). So the camera module must be low-power.
Driver / UVC / compatibility	The camera module should ideally be UVC (USB Video Class) so that the phone's OS can talk to it without custom drivers. Many USB camera boards advertise "works on Android" or "UVC" features. For iOS it's much harder to use external cameras, unless using specific external hardware + SDK.
Latency & throughput	The USB link and the phone's processing must keep up with the frame rate. High fps and low latency matter.
Mounting & optics	The module must be mountable (on glasses, phone case, etc.), and lens distance, FOV, focus must suit eye distance.
Lighting / IR illumination	Same as before — you'll likely want IR LEDs and IR filters etc.

Given those, the strategy is to pick a small USB camera module that is UVC and low-power, then integrate it via OTG to the phone.

🔍 Candidate camera modules (small USB / UVC) you can try

Here are some modules you can experiment with. None are ideal out-of-the-box for eye-tracking, but some are good starting points:

Adafruit Ultra Tiny USB Webcam Camera
$6.95
•
Adafruit Industries + others

USB OV7675 Camera Module
$4.87
•
AliExpress

OTG External Camera 1080p Module
$29.51
•
Walmart - Personal Care Specific

HD USB Camera Module (OV9726)
$9.18
•
Walmart - Seller

USB Camera Module 720p
$14.87
•
Walmart - Seller

GC2093 2MP USB Camera
$23.99
•
Waveshare + others

OV9726 USB Camera Module
$11.40
•
Walmart - Seller

OV2659 HD Camera Module
$7.35
•
Walmart - Home Co., Ltd.

Here are some of the picks worth highlighting:

- **Adafruit Ultra Tiny USB Webcam Camera** — extremely small, low power. Good starting point to test external USB camera on a phone.
- **USB OV7675 Camera Module** — low resolution, but simplicity and low latency could help in prototyping.
- **OTG External Camera 1080p Module** — designed for OTG / phone usage, which is promising for compatibility.

- **HD USB Camera Module (OV9726)** — somewhat higher quality with OV9726 sensor; likely better for eye detail.
- **USB Camera Module 720p** — a balanced mid option.
- **GC2093 2MP USB Camera** — higher MP, more data to handle.
- **OV9726 USB Camera Module** — alternative OV9726 version.
- **OV2659 HD Camera Module** — HD mid-tier option.

🛠️ Better modules / global shutter options (higher end)

If you can stretch budget or work from a flexible budget, these are more capable modules:

- Arducam sells a **1MP global shutter USB camera board, 100 fps, UVC, with a low-distortion lens** — it mentions compatibility with "Android device" support. Arducam
- ELP has a **global shutter USB camera module (AR0234 sensor, 1080p up to 90 fps, supports Android / UVC)**. Amazon+2ELP CCTV+2
- There is a **See3CAM_24CUG** model: a Full HD global shutter USB camera with AR0234 sensor, compatible with Android via UVC. e-con Systems

These are better suited for eye-tracking (less motion distortion), though more expensive.

🎯 What to try first (a minimal path)

1. **Get a small UVC USB camera module** like the Adafruit Ultra Tiny or the OTG External Camera listed above.
2. **Use an OTG (USB host) adapter cable** to connect it to your Android phone (if your phone supports OTG mode).
3. **Use a UVC camera app or library** — there are Android apps/libraries that can display and access external USB cameras (e.g. *USB Camera – Connect EasyCap or UVC* apps, or using `android.hardware.usb` + camera2 / custom code).

4. **Run pupil detection / gaze software** on the phone: e.g. using OpenCV (Android port) to detect the pupil / glints in the external camera frames, map to gaze.

If that works, you can iterate: add IR LEDs, better optics, calibration, etc.

Midterm Presentation Notes

Excellent — adding the **electrical design** section will give your midterm presentation a more complete, systems-level view of the hardware.

Here's how we can **extend your existing script and slide set** to include the electronics — without making it too long for a 5–7 min talk.

⚡ Updated Midterm Presentation Script — Hardware + Electrical Design

(Eye-Gaze Tracker Pen Project)

Slide 1 — Introduction & Motivation

> "This project aims to miniaturize the Tobii PC Eye Tracker into a **portable, phone-based gaze-tracking pen**.
> The hardware combines **mechanical design**, **embedded electronics**, and **camera integration** to enable low-cost accessibility for blind or low-vision users.
> I'll walk through the **mechanical CAD**, **electrical design**, **materials**, and **next steps**."

(Visual: Tobii vs. your concept render.)

Slide 2 — CAD & Parametric Design

> "The 3D model was built in Fusion 360 using **parametric design principles**.
> Key dimensions—camera mount diameter, screw hole spacing, and wall thickness—are all parameter-driven, allowing easy iteration for different camera modules or manufacturing tolerances."

(Visual: Fusion 360 model + parameter table.)

Slide 3 — Electrical Architecture

> "The electronic subsystem consists of three main parts:

1. A **UVC USB micro-camera** for real-time eye imaging.

2. Two **IR LEDs** positioned symmetrically around the lens for pupil illumination.

3. A **USB-C OTG connection** to the phone, which supplies 5 V power and streams video frames for processing in an Android or iOS app.

A small **boost-regulated LED driver circuit** controls current to the IR LEDs, preventing overheating and ensuring uniform brightness.
Future versions could integrate an **IMU** for motion compensation or a **microcontroller** for preprocessing."

(Visual: simple schematic diagram showing camera, LEDs, resistors, USB-C.)

Slide 4 — PCB & Wiring Design

"A compact PCB or flex-cable routes power and data neatly within the housing.
We used **KiCad** to design a single-layer layout with SMD resistors and a JST connector for modular wiring.
Wire management channels are included in the CAD to minimize strain and interference near the optics.
The design follows **low-EMI routing** and **ground-plane continuity** to maintain camera signal integrity."

(Visual: 2D PCB layout or wiring sketch over the 3D model.)

Slide 5 — Materials, Manufacturing & Tolerancing

"The enclosure will be 3D-printed in **PLA or PETG** for prototyping and possibly **resin-printed** for finer tolerances.
Critical dimensions, like the camera press-fit seat and LED holes, are assigned ±0.1 mm tolerances.
After printing, each unit will undergo **fit-check and alignment** tests to ensure the optical axis and LED beams converge correctly."

(Visual: render + exploded view.)

Slide 6 — Assembly & Animation

> “An assembly animation can help visualize how the components fit together.
> In Fusion 360’s *Animation* workspace, I can use **Move Components** to show the order of insertion for the camera, LEDs, PCB, and shell halves.
> This not only demonstrates manufacturability but also serves as a **virtual quality-check** for clearances before printing.”

(Visual: short animation clip or exploded sequence screenshots.)

Slide 7 — Next Steps

> “Immediate next steps include:

1. **Printing and assembling** the first prototype.
2. **Testing the IR LED illumination** and verifying eye visibility under low light.
3. Measuring **camera alignment and focus distance** for optimal pupil detection.
4. Evaluating **power draw and thermal performance** on phone OTG power.
5. Integrating the **OpenCV-based gaze-tracking app** and calibrating mapping accuracy.

Future work will focus on miniaturizing the PCB, refining ergonomics, and exploring **injection-molded housing** for mass production.”

(Visual: checklist graphic or prototype roadmap.)

💡 Optional: Add to Quality Control Section

> “Electrical QC will include **current draw measurements**, **IR intensity uniformity tests**, and **signal integrity checks** on the USB line using an oscilloscope.”

⚙️ Quick Note — How to Visualize the Circuit

- Use **Fritzing** or **KiCad’s 3D viewer** to export a board render.

- Overlay the wiring paths on your CAD model using Fusion 360's *Canvas Insert* feature for a hybrid mech–elec visualization.
- If you want to simulate power flow, **Tinkercad Circuits** works well for quick IR LED + resistor prototypes.

Perfect — let's break down **exactly how to design your schematic** for the **eye-gaze tracker pen's electrical system**, step by step.
This will give you a clear picture of both the **circuit design** and **layout planning** before moving to PCB or wiring integration.

⚡ 1. Core System Overview

You're building a **low-power USB device** powered by your phone via **USB-C OTG**, consisting of:

- A **UVC camera module** (like Adafruit or OV5640 USB camera)
- **Two IR LEDs** for eye illumination
- **Current-limiting resistors or LED driver**
- Optional: a **switch** or **microcontroller** (for power control or future upgrades)

Think of it as a simple *"hub" circuit* that manages power distribution and illumination, with the camera's data line going directly to the phone.

🧭 2. High-Level Circuit Flow

```
[Phone USB-C OTG Port]
      |
   +5V / GND
      |
  -------------------
  |              |
[Camera Module]   [LED Branch]
```

```
        |
    +---> [Resistor] ---> [IR LED 1]
    +---> [Resistor] ---> [IR LED 2]
```

- **Camera module** receives power and transmits data directly via USB (no MCU required for first prototype).
- **LED branch** takes 5V and runs two IR LEDs (typically 850 nm or 940 nm), each with its own resistor (to ensure even brightness).
- You can add a **switch or transistor** to toggle LEDs if needed.

🧩 3. Detailed Component Breakdown

Component	Function	Suggested Part	Notes
UVC USB Micro Camera	Eye imaging	ELP 5MP USB Camera or Adafruit 1937	Must support UVC for Android/iOS
IR LEDs (x2)	Eye illumination	SFH 4650 / Vishay TSAL6400	850–940 nm range
Resistors (x2)	Current limit	100–150 Ω (¼ W)	Value depends on LED forward voltage
Switch (optional)	LED ON/OFF	SPST tactile switch	Useful for debugging
USB-C Connector	Power/data	USB-C breakout board	Connect VBUS and GND to both LED and camera
PCB (optional)	Integration	Custom via KiCad	Combine LEDs, resistors, and connector neatly

---

🔌 4. Electrical Design in Fritzing or KiCad

Option A — *Fritzing* (simpler, for presentations)

- Place a **USB-C connector** (VBUS → +5V, GND → ground).

- Add the **camera module** as a peripheral (or label it as "UVC Module").
- Connect **+5V** → **resistor** → **IR LED** → **ground** for both LEDs.
- Group both LED branches under a single "LED Power" label.
- Use clean labeling, color-coded wires (red = 5V, black = GND, blue = data if needed).

💡 *Tip:* Fritzing lets you visually show wires and connectors — perfect for a midterm schematic slide.

Option B — *KiCad* (professional workflow)

1. Create a **new schematic** with `power +5V` and `GND` nets.
2. Add components:
   - `USB-C_Connector` (from library)
   - `LED_IR` (2x)
   - `Resistor` (2x)
   - Optionally, a `Switch_SPST` or `NPN transistor` for LED control
3. Label nets (e.g., "LED_PWR", "VBUS", "GND").
4. Run **Electrical Rules Check (ERC)**.
5. Export the schematic to PDF or use **KiCad 3D viewer** for your slide.

⚙️ 5. Example Design Parameters

For a standard 850 nm IR LED:

- Forward voltage ≈ **1.2 V**

- Desired current ≈ **20 mA**

Resistor value:

R = (5 V - 1.2 V) / 0.02 A ≈ 190 Ω

→ Use **180 Ω or 200 Ω** (standard E12 values)

📏 6. Layout Considerations

If you make a PCB:

- Keep the **camera's signal lines** (D+/D–) short and twisted (if routed).
- Separate **LED power** from data lines to avoid noise.
- Mount LEDs at equal angles on either side of the camera lens (≈20–25° outward) for even corneal reflection.
- Use **vias** and **ground plane** under LEDs for thermal relief.

🧪 7. Testing & Validation

Before connecting to the phone:

1. Test LED brightness using a multimeter (check current).
2. Confirm the phone recognizes the USB camera.
3. If LEDs flicker when camera is active → use a **decoupling capacitor** (10 μF near the connector).
4. Record sample eye footage and verify pupil visibility in IR.

🔮 8. Future Electrical Upgrades

Later iterations can include:

- **Tiny MCU (e.g., ATtiny or ESP32)** for LED modulation or blink sync.
- **Ambient IR sensor** for adaptive LED brightness.
- **Battery power** for standalone operation.

Final Hardware Transcript

Hardware Breakdown (video1)

Start

Thank you Yi for that wonderful overview on user research. In the next 4 slides, I will focus on the hardware side, turning those insights into our compact phone base eye tracking pen.

Our community partner, Jessica, has been using the Tobii PCEye tracker for years depicted in the top right hand corner. In order to meet her wants, we are exploring a smarter- sleeker design that resembles the size of a pen depicted in the bottom right hand corner.

The CAD model demonstrates how easy it'll be to modify the dimensions, such as the camera diameter and the wall thickness without redrawing the entire thing. The image shows a simple base geometry that's going to support the small camera module, microcontroller and IR LEDs.

We want to accomplish three things out of this physical prototype: a smaller design to reduce bulk, multi-device compatibility with the addition of the USB Type C cable, and most importantly - something that has an anti-reflective coating to reduce glare and improve pupil detection outdoors.

In this cost breakdown table we have limited the design to five components, all under $9 each. The provided enclosure and microcontroller will be absolutely free.

Now over to Beverly for the software.

End

Hardware Next Steps and End(video2)

Start

We decided to build this dedicated design instead of relying only on iPhone software, just because it gives us more control over accuracy and usability. The built in phone cameras aren't really optimized for eye-gaze tracking, especially in brighter variable lighting. But with our hardware, we can add infrared illumination, better optics and design for multiple devices, not just an iPhone. Now this is what has influenced the next steps in terms of the hardware. We're going to combine all those components mentioned earlier into a compact unit. We're gonna calibrate the gaze tracking for accuracy under varied lighting. Then we're just gonna test the performance in sunlight with a simulated gaze interaction task. And it's with this test that we will refine our software mapping and hardware alignment based on results. Well, I would like to thank you for listening to our presentation and coming along with us on this journey. We look forward to presenting the final prototype. Thank you.

End

Hardware development

NOVEMBER
2/11/25t : Acquired new camera module, link.
Linked camera module to Pico 2w Zero
Realised issue with memory card for environment set up and module testing.
3/11/25: Confirmed issue with memory card. Repurchased new SD.
4/11/25 to 7/11/12 (Out of town)
Thursday Evening to-do list created:

- [x] ~~Written update for team~~
- [ ] Final Schematic design
- [ ] Provide environment details and programming language needed
- [ ] Code flow (maybe)

8//11/25 (Not available)
9/11/25 : resume prototyping and design

Redirection: Test features with Android

1) Build prototype
2) Build software app
3) Design app

Thoughts: not possible to do all 3. Decide what to focus on. Jessica had issues with the phone and feedback from the midterm showed a lack of focus on it- how can we make up for that?

⚙️ Hardware layout (minimal viable build)

Components

- **Pi Zero 2 W** – runs camera driver, image capture & OpenCV
- **Global-shutter camera** – CSI ribbon cable to Pi Zero's camera connector
- **2 × IR LEDs (850 nm)**
- **1 × logic-level N-MOSFET** (AO3400 / IRLZ44N / SI2302)
- **2 × resistors for LEDs** (150–180 Ω @ 5 V)
- **1 × 100 Ω gate resistor** + **10 kΩ pulldown**

Environment: Raspberry Pi Pico 2W OS

Language: Python

🧠 Why We Chose Hardware over iPhone Eye Tracking

1️⃣ Precision & Lighting Control

- iPhone uses a single RGB or TrueDepth camera → struggles in sunlight and variable lighting.
- Our module adds **IR illumination + global shutter**, giving stable pupil tracking anywhere.

2️⃣ Open Data & Modularity

- Apple's system is closed — no access to raw gaze or pupil data.
- Our hardware provides full image access → ideal for **research, calibration, and prototyping**.

3️⃣ Customizability & Integration

- iPhone eye tracking is limited to on-screen cursor control.
- Ours integrates with **other devices** (Android, microcontrollers, or haptic tools).

4️⃣ User-Centered Flexibility

- Apple requires face-level positioning.
- Our device can be **mounted freely** — tabletop, glasses, or pen — adapting to user posture.

5️⃣ Accessibility Beyond Vision

- iPhone still assumes visual UI interaction.
- Our design supports **audio or tactile feedback**, making it inclusive for **blind and low-vision users**.

Potential redirect- based on progress made on Sunday.

📱 Concept: Anti-Glare Eye-Tracking Case

🔹 1. Purpose

- Shade the front camera from sunlight → improve exposure consistency.
- Reduce reflections on both the **camera lens** and **user's eyes**.
- Potentially add a **filter window** (IR-pass or ND filter) for controlled imaging.

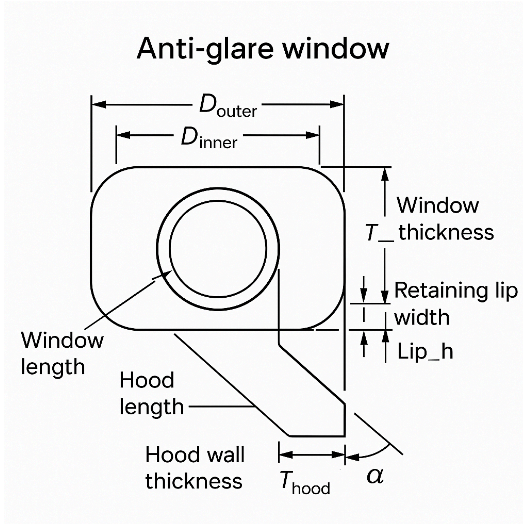
Essentially, it becomes a **mini optical lab around the selfie camera**.

🧩 Design Components

Element	Function	Material / Notes
Camera hood / shroud	Blocks side sunlight	3D-print or thermoform black ABS/TPU; matte interior
Anti-glare / matte window	Diffuses incoming light	Matte acrylic, or AR-coated polycarbonate
Optional IR-pass film	Blocks visible light (for custom camera apps)	850 nm IR-pass filter sheet
Mount ring / bezel	Snaps around front camera area	Toleranced to camera cutout on case
Case body	Protects phone & integrates mount	Off-the-shelf case with added printed insert
Vent slots	Prevent condensation in sunlight	Small holes on side of shroud

---

🔧 Prototyping Method (1–2 weeks total)



1. **Pick a base case** – soft TPU case for your iPhone model.
2. **Model an attachment ring** in Fusion 360 or SolidWorks that fits around the selfie camera region.
3. **Add a short hood (5–10 mm deep)** angled slightly forward/downward.
4. **Line the inside** with black velvet paper or matte paint to absorb reflections.
5. **Add a replaceable window** (matte acrylic or IR-pass plastic) with friction fit or small screws.
6. **Print it in black PETG or TPU** – light, heat-resistant, flexible.
7. Optionally **add a clip-on variant** for quick testing before integrating into the case.

☀️ Expected Gains

Issue	Improvement from Case
Sun glare into lens	70–90 % reduction with matte hood
Face brightness inconsistency	More stable exposure, fewer blown highlights

Pupil visibility	Clearer contours for both iOS and custom tracking apps
Reflection off glasses	Reduced due to shading geometry

🔬 Add-on Enhancements

- **Neutral Density (ND) filter film**: lowers brightness uniformly, keeps contrast high.
- **Removable hood**: test different depths (short vs long) to balance aesthetics and function.
- **Magnetic mount**: swap between anti-glare and IR-pass versions quickly.

🧠 Presentation Angle

If you include this in your project:

> “To address outdoor reliability, we’re prototyping an anti-glare iPhone case with an integrated optical shroud and filter window. It reduces sunlight interference by controlling the light reaching the front camera, effectively extending the usability of mobile eye-tracking into outdoor and high-glare environments.”

Best path (my recommendation)

Go with “Pi-centric MVP.”
Use the **Raspberry Pi Zero 2 W + your global-shutter camera** for capture and stream it to your **Android** over Wi-Fi. Drive your **IR LEDs** from the Pi through a small MOSFET. This gives you a compelling, reliable demo with the parts you already own—no new camera or firmware rabbit holes.

Why this path

- Uses the **camera you have** (CSI + Pi OS driver).

- **Robust outdoors** once you add IR + the small anti-glare hood/filter.
- Simple, impressive demo: Android shows the live feed; you toggle IR and the pupil “pops.”

Two good alternatives (only if your goals shift)

B. MCU-only build (no Pi): swap to a **Himax HM01B0/HM0360** camera that an RP2040/ESP32 can read directly.

- Pros: smallest hardware, lowest power.
- Cons: requires buying a different camera; lower res/fps.

C. Add iPhone/Android control later: keep the Pi camera pipeline, add an **ESP32-S3 or nRF52840** emitting **BLE HID** so gaze events become taps/scrolls in iOS Switch Control.

- Pros: real phone control.
- Cons: extra coding; not needed for midterm.

What to do next (12 hours total over 2 weeks)

Week 1 (~6 hrs): Wire + Stream + IR proof

1. **Wire IR LEDs via MOSFET** (AO3400/SI2302), each LED with **150–180 Ω** to 5 V; MOSFET gate ← **GPIO18** through **100 Ω**, gate pulldown **10 kΩ** to GND.
2. **Enable camera** on the Pi Zero 2 W; verify `libcamera-hello` works.

3. **Stream video** with `mjpg-streamer` (or `libcamera-vid --stdout | ffmpeg`) and view on Android at `http://<PiIP>:8080`.

4. **Toggle IR** brightness in Python (`gpiozero.PWMLED`) and film **IR off vs on**.
 Deliverable: 20–30 s clip proving IR makes pupils/glints clear.

Week 2 (~6 hrs): Simple processing + slides

1. Install **OpenCV**, show a live **thresholded pupil mask** (OTSU or adaptive threshold).

2. Capture a short side-by-side (raw vs mask).

3. Add your **anti-glare hood/filter** piece to reduce sunlight washout; snap a photo of the setup.

4. Build **3 slides**: Hardware diagram → IR effect → Next steps (gaze mapping + BLE HID).

Minimal BOM (if you're missing anything)

- N-MOSFET: **AO3400** (or SI2302/IRLZ44N)
- Resistors: **2× 150–180 Ω** (LEDs), **1× 100 Ω** (gate), **1× 10 kΩ** (pulldown)
- IR LEDs: **850 nm**, 20–30 mA each
- 5 V supply (power bank) & jumper wires
- (Optional) **Matte/IR-pass window** & **3D-printed hood** for the camera

---

Success criteria (so you know you're done)

- Android shows **stable 30–60 fps** stream from the Pi.
- With IR **off** → **on**, your **pupil mask** becomes clean and persistent.
- Outdoor/bright test improves with the **hood/filter**.

11/13/2025 Meeting

-Set up gaze-tracking environment
-generate instruction list
-talk about dynamics and next steps in terms of project
-reiterate simple tracking is easy to set up but in light of the above not too confident about direction

Instructions in the subtab

Environment Setup Instruction Manual

📘 Raspberry Pi Zero 2 W — Eye Gaze Tracker Setup Manual (Instruction Guide)

Version 1.0 – Hardware Assisted IR Eye Tracking Prototype

---

1. Overview

This manual documents the steps used to set up and access the Raspberry Pi Zero 2 W ("robojojo") for the eye-gaze tracking prototype.
You will learn how to:

✔ Connect to the Pi over SSH
✔ Verify network connectivity
✔ Install system updates
✔ Enable the camera
✔ Prepare the Pi for running the eye-tracking software

This guide reflects the actual commands and troubleshooting done so far.

---

2. Preparing for SSH Access

2.1 Ensure Raspberry Pi is on Wi-Fi

The Pi Zero 2 W must be connected to the same Wi-Fi network as your computer or mobile hotspot.

---

3. Checking Network Connectivity

You can test whether the Pi is discoverable using `.local` hostname resolution.

3.1 Ping the Hostname

Open **Command Prompt** on Windows:

```
ping robojojo.local
```

If successful, you will see replies like:

```
Reply from 2600:1017:b4ad:1fab:99:a615:ea1b:463a: time=27ms
```

If unsuccessful:

- Device may be using **IPv6 only**, or
- Hostname resolution may be failing.

3.2 Force IPv4 Ping

This is the most reliable method:

```
ping -4 robojojo.local
```

Output example (correct):

```
Pinging robojojo.local [172.20.10.13] with 32 bytes of data:
Reply from 172.20.10.13: bytes=32 time=10ms TTL=64
```

Now you know the **Pi's IPv4 address** (in this example: `172.20.10.13`).

---

4. SSH Into the Raspberry Pi

4.1 Use SSH with the Correct Username

You initially tried:

```
ssh jojo@robojojo.local
```

But authentication failed.
You then used the correct username **robojojo**:

```
ssh robojojo@robojojo.local
```

If prompted:

```
The authenticity of host can't be established...
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Type:

```
yes
```

Then enter your password.

Successful login looks like:

```
Linux robojojo 6.12.47+rpt-rpi-v8 #1 SMP PREEMPT ...
```

---

5. Update System Packages

Once logged in:

```
sudo apt update && sudo apt upgrade -y
```

This downloads and installs system updates.

Your session showed:

- 132 packages upgrading
- 6 new packages installing
- Raspberry Pi OS moving from “testing” to “stable” suite

---

6. Enable Camera Interface

To use the global-shutter camera, run:

```
sudo raspi-config
```

Then navigate:

Interface Options → Camera → Enable

Reboot after enabling:

```
sudo reboot
```

---

7. Install Required Software for Eye Tracking

After reboot, SSH back in and install dependencies:

```
sudo apt install python3-opencv python3-picamera2 python3-gpiozero -y
```

Optional (for streaming to Android):

```
sudo apt install mjpg-streamer -y
```

---

8. Testing Camera Operation

Try:

```
libcamera-hello
```

If successful, the camera is working.

---

9. Starting Eye-Tracking Pipeline (via SSH)

In the final setup, you will start your eye tracker with one command after SSH:

```
python3 eye_tracker_demo.py
```

This script:

- Activates IR LEDs (via GPIO PWM)
- Opens camera feed
- Displays live frames + pupil mask
- Streams video if mjpg-streamer is enabled

You can also run the streaming server:

```
mjpg_streamer -i "input_libcamera.so --width 640 --height 480 --fps 60" \
              -o "output_http.so -p 8080"
```

Then visit on Android:

http://<PI_IP>:8080

---

10. Troubleshooting Log (Based on Your Session)

Issue	Explanation	Fix
“Ping request could not find host robojojo.local”	Windows not resolving MDNS	Use `ping -4` or direct IP

SSH “Permission denied”	Wrong username	Use `ssh robojojo@robojojo.local`
`sudo raspi-config` not found on Windows	You attempted to run it on PC, not Pi	SSH into Pi first

---

11. Recommended Final Setup Workflow

1. Connect phone hotspot → Pi joins it

On laptop: `ping -4 robojojo.local`
`ssh robojojo@robojojo.local`

2. Enable IR LEDs and camera: `python3 eye_tracker_demo.py`
3. Optional: open stream on Android

End session: `exit`

---

12. Appendix

To change the Raspberry Pi user password:

```
passwd
```

To check the Pi camera:

```
vcgencmd get_camera
```

Expected output:

```
supported=1 detected=1
```

👁️ UX/UI Notes

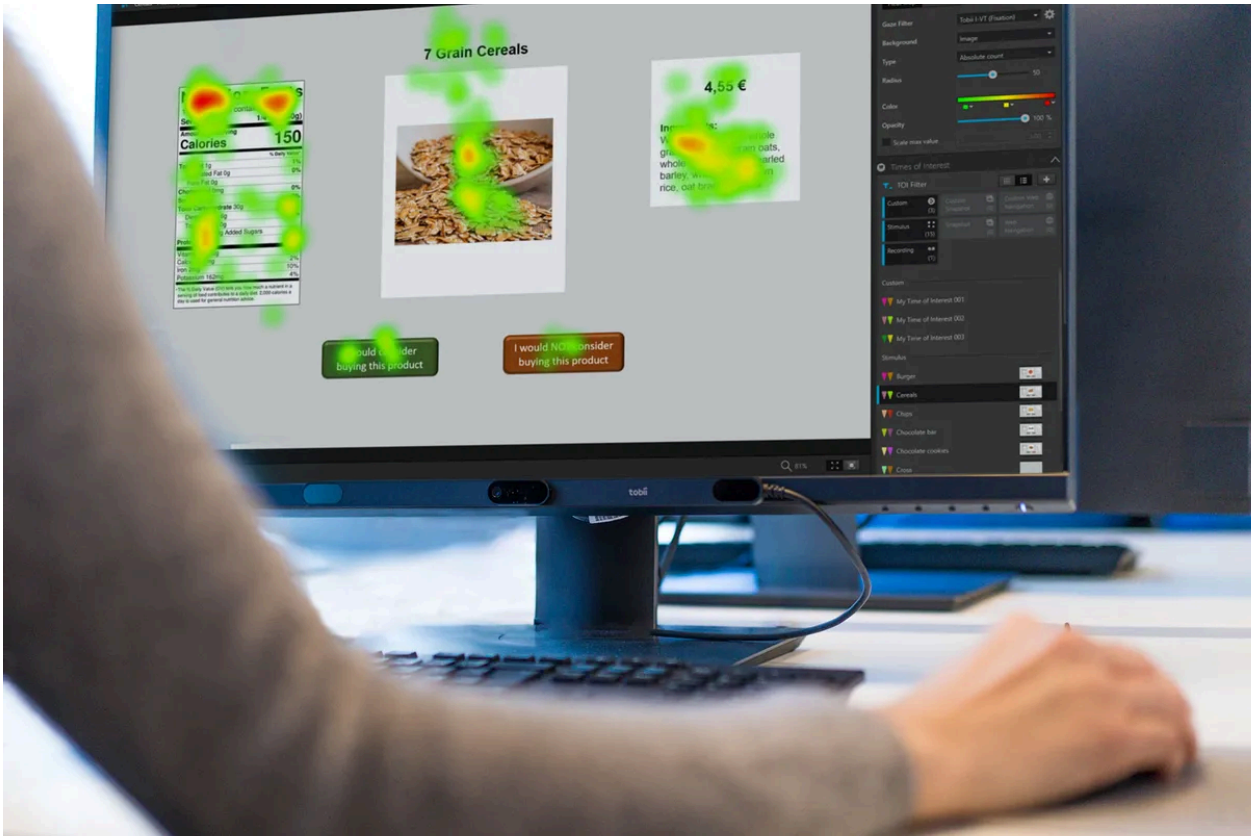
Yi To Do
Oct 16th
competitive analysis:
What the getting products and targets in similar users
What the current

Oct 9th
Get the short answer about sunlight: it's because of the tech part, not the software or interface

Or maybe try to search on Web Accessibility to check if any principle would be relevant.
WCAG 2 Overview | Web Accessibility Initiative (WAI) | W3C

Eye Tracking

Eye tracking technology records the movement and fixation points of the eyes, providing data on where individuals look, how long they look, and the sequence of their gaze. Real eye tracking involves the use of hardware such as wearable eye trackers or screen-based eye trackers equipped with cameras that track eye movements in real-time. This method captures natural shopping behavior in realistic shopping environments. Predictive eye tracking, by contrast, relies on algorithms trained on visual data to estimate attention distribution across a design. These tools, like Tobii Pro Lab, attempt to replicate human visual attention using heatmaps and gaze plots without real-time human input. While cheaper and quicker, predictive tools sacrifice depth and accuracy for convenience.



Eye Tracking Market Summary

The global eye tracking market size was estimated at USD 638.8 million in 2021 and is projected to reach USD 8.06 billion by 2030, growing at a CAGR of 33.4% from 2022 to 2030. The increasing use of smart sensors for process control and decision making, and the rising demand for contactless biometrics, coupled with the growing adoption of Artificial Intelligence (AI), Virtual Reality (VR), and Augmented Reality (AR) in consumer electronics & other commercial application areas, are expected to drive the market growth significantly.

Key Market Trends & Insights

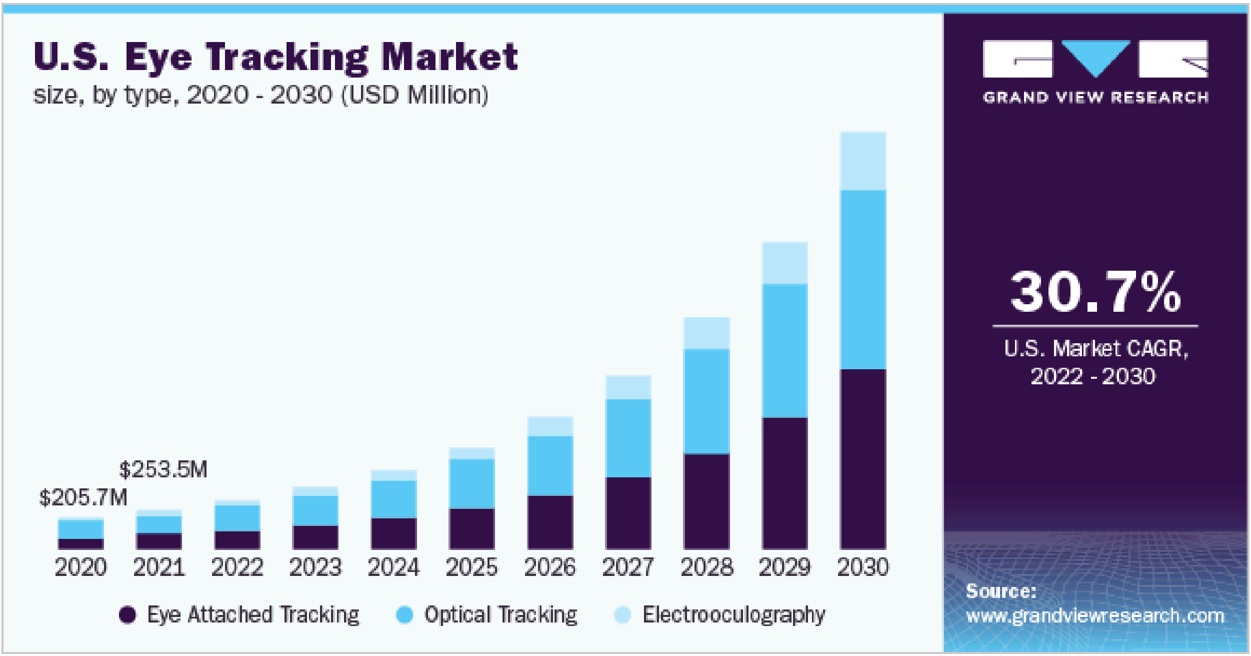
The North American region dominated the market with a revenue share of more than 48.00% in 2021 and is expected to retain its dominance over the forecast period.

The U.S. accounts for the largest revenue share in the regional market owing to the wide adoption of smart sensors and contactless biometric systems.

In terms of type, the optical tracking segment led the market with a revenue share of more than 52.00% in 2021.

Based on component, The hardware segment dominated the market with a revenue share of 60.01% in 2021.

On the basis of location, The remote segment led the market with a revenue share of more than 62.00% in 2021.



Eye Tracking Product Technology Insights

1. Diverse application domains

   Eye tracking is not just for accessibility; it's used in UX / interface testing, advertising and marketing (attention measurement), retail/shopper research, automotive (driver monitoring & operator attention), healthcare & assistive technology, AR/VR / extended reality, neuroscience / cognitive research, etc.

   Eye Tracking Uncovers Insights in Consumer and UX Research - Tobii

   North America Eye Tracking Solutions Market Size & Share Analysis - Industry Research Report - Growth Trends

2. Hardware + software / analytics layers
   - The "component" segmentation typically divides hardware (cameras, IR sensors, optics, head trackers) vs. software / analytics, with hardware often commanding a major share (i.e. many forecasts show hardware leading)

3. Type of eye tracking
    - Remote (screen-based) systems
    - Wearable / head-mounted
    - Eye-attached / clip-on
    - Also non-optical methods (e.g. electrooculography, appearance-based, etc.)
      Straits Research+2Data Bridge Market Research+2
      Many forecasts project that **remote / optical tracking** will maintain a large share given its applicability in UX, consumer research, etc.

4. Competition & key players
    - Tobii
        i. Provider of eye-tracking software and hardware. The technology enables one to detect eye orientation, and thus the object of the gaze. Using innovations in technology, the company's mission is to bring eye tracking into broader use in applications such as eye control interfaces for computers, design testing, and medical diagnostics.
    - Smart Eye
        i. AI-powered driver monitoring system for vehicles. The platform offers features like measurements of the drivers head pose, gaze direction, eyelid closure, image analysis, stimulus recording, and video analysis. It offers services for automobiles, airplanes, trucks, trains, simulators, control rooms, and more.
    - Varjo
        i. Provider of virtual reality headsets. It enables users to experience virtual and augmented reality content and train, create and learn in superimposed or virtual environments. It caters to simulation, architecture, 3D model designing, medical, and other applications.
    - NovaSight
        i. Provider of eye-tracking-based solutions for vision assessment and treatment. It offers medical devices to prevent vision impairments among pediatric patients using eye-tracking-based technology in novel diagnostic, therapeutic, and preventive solutions. Its products include CureSight an amblyopia treatment system, designed to replace traditional eye patching, TrackSight a software-based vision health monitoring and myopia prevention system, and Eye swift an eye-tracking-based vision diagnostics system.
    - OmniVision
    - iMotions
        i. Provider of human insights software. Across its desktop and online applications, iMotions allow users to collect and analyze human behavior data through eye tracking, galvanic skin response, facial expression analysis, EEG, EMG, and ECG on one convenient platform. Backed by

science, this multi-modal software suite is the most comprehensive human insights tool currently available on the market.

5. Challenges, limitations & cautions
    - Prediction / simulation vs real tracking: some “predictive eye tracking” tools (i.e. algorithmic heatmap estimation) are criticized for overpromising, and are less reliable than real gaze data in some contexts.
    - Interpretation: eye fixation / gaze data are proxies, not direct measures of intent, so must often be combined with qualitative or cognitive methods.
    - **Calibration, noise**, **glasses** / occlusions, **ambient lighting, head motion** — these technical constraints still limit where and how reliably gaze systems can be used.
    - Costs: high-precision eye trackers remain expensive, particularly in research-grade domains, which limits adoption in cost-sensitive segments.

PCEye & Tobii in the Ecosystem, and Comparative Positioning

PCEye (Tobii Dynavox PCEye)

- **Description / positioning**
    - PCEye is a compact / clip-on eye tracker designed primarily for assistive / accessibility use. It allows users with physical disabilities to control a Windows computer using gaze, offering features like dwell-based clicking, zoom, and cursor control. Sensory Guru+3Tobii Dynavox Global+3Tobii Dynavox Global+3
    - The PCEye 5 is one of the latest models; it is marketed to work across screen sizes, indoor/outdoor settings, etc. Tobii Dynavox Global+1
    - Price point: For example, in UK/Europe it’s listed ~ £1,995 to £2,440 (i.e. relatively high cost). Sensory Guru
    - Use cases: designed to enable communication, computing, independence for people with severe motor impairments, rather than general UX or advertising research.
- **Strengths / market niche**:
    - Strong built-in value in the assistive / accessibility domain, which is a stable, mission-driven market.
    - Because it’s specialized, it can command premium pricing in that niche.
    - As accessibility demand increases (legal, regulatory, inclusive design trends), demand could grow.
- **Limitations / challenges**:

    - Compared to research-grade trackers, the precision, sampling rate, and latency might be more constrained (f**or many assistive use cases, "enough precision" is more important than ultra high fidelity**).
    - The market beyond assistive use is competitive and has different requirements (e.g. **UX or marketing researchers may demand analytics, integrations,** etc.).
    - The overall size of the assistive eye-gaze subset is much smaller than the entire eye-tracking / gaze market.

Tobii (Eye Gaze / Tobii's Research / Consumer / UX side)

- Tobii is one of the leading names in eye tracking, with strong presence in research-grade, UX / usability, consumer insight, and attention analytics.
- They also offer consumer / gaming eye trackers (e.g. Tobii Eye Tracker 5 for gaming / head & eye tracking) and platforms that integrate gaze with experiences.
- In the marketing / consumer insights space, they emphasize how eye tracking can "see through the eyes of consumers" to assess attention on packaging, ads, in-store signage, etc.
- They also provide consulting / research services (Tobii Insight) to help organizations integrate gaze data into business decisions.

Positioning / strategic advantages:

- Broad product portfolio: from research-grade systems, consumer/gaming trackers, to integrated services and software.
- Established brand, research credibility, academic visibility.
- Strong relationships in UX / marketing / consumer research.

Challenges / competitive risks:

- In high-volume low-cost segments (e.g. simple web gaze tracking), there may be downward pricing pressure.
- Maintaining differentiation in analytics, ease-of-use, and integration with other sensors (EEG, biometric, VR) is critical.
- Competitors in automotive (Smart Eye, Seeing Machines) and niche players in assistive tech (like PCEye) are strong in their domains.

Technical Constraints in Assistive Eye-Gaze Systems

In assistive technology (AT) contexts — e.g., users with motor impairments, ALS, cerebral palsy, or spinal cord injuries — eye-gaze systems like **Tobii PCEye**, **EyeTech DS**, and **EyeControl** enable computer or communication control. However, these setups face unique **environmental and physiological constraints** that limit reliability and inclusivity.

Constraint	Description	Impact in Real Use
Ambient lighting	Sunlight or strong reflections can overwhelm infrared (IR) sensors used in most gaze trackers. Glass reflections, windows, and screens all affect performance.	Loss of pupil detection, signal dropouts, or total tracking failure outdoors. Tobii explicitly warns that strong sunlight can “blind” the tracker.
Calibration difficulty	Users must fixate on calibration points, but many users (esp. with motor impairments) can’t keep a steady head position or prolonged gaze.	Poor calibration leads to cursor drift and mis-selections. Frequent re-calibration causes frustration.
Noise and drift	Sensor noise, head tremor, or micro-movements cause instability in gaze point estimation.	Pointer jitter, inaccurate dwell selection, or latency.
Glasses / occlusion	Reflections from eyeglasses or curved lenses distort IR light. Droopy eyelids or frequent blinking add further occlusion.	Tracking interruptions or complete loss of one eye.
Head movement	IR systems assume limited head movement. Excessive motion changes reflection geometry and breaks tracking.	Cursor jumps off screen, forcing users to reset position.

Uncontrolled posture & environment	Users may operate in bed, on wheelchairs, or tilted setups — so camera angle and eye distance vary constantly.	Tracking accuracy drops sharply if the camera's relative angle changes.

Directions of Improvement

User Journey

User Journey/Flow

https://www.figma.com/board/hL43yuOGjHFStjaK0Ekmn3/Jessica-Tobii-User-Flow-Pain-point?node-id=0-1&t=TFXqxbWgdXg2aPpH-1

Midterm Script

iSee Intro:

Yi: Hi, we're iSee Group. Our project explores an **eye-gaze control system** from an **assistive technology perspective**, focusing on our main client, **Jessica**, who has **limited upper-body mobility**. I'm Yi, I'm responsible for interaction design. My partners Bev is for software development and Jo for hardware design.

Eyegaze Tracking Research

Yi: Eye-tracking technology records where and how long a person looks using cameras that follow eye movements in real time. In assistive contexts, this allows users like Jessica to **control digital interfaces hands-free**, promoting **independence and accessibility** in daily life. The image shows a person using a Tobii eye-tracking system on a computer monitor. The screen displays a product comparison interface with a green-to-red heat map overlay showing where the user's gaze is focused, mainly on a cereal image, a nutrition label, and a price tag. The Tobii eye tracker is mounted below the monitor, connected by a USB cable.

Eyegaze Tracking Research

Yi: We analyzed existing eye-tracking systems such as Tobii Dynavox, PCEye and so on. Most provide reliable gaze control, but they struggle with outdoor lighting, precise calibration, and user comfort.

User Journey & User Flow

Yi: We created user journey and workflow maps to understand Jessica's real-world experience with the Tobii eye tracker.

Yi: This is a horizontal user journey map for Jessica using eye tracker. It shows six stages while using the tech includes feelings, actions, and design opportunities for improvement. You can use the link on the slide to explore the full details.

Yi: This is a user flow mapping Jessica's interaction with the Eye Tracker in six stages. Link on the slide to explore the full maps in detail.

Yi: It's a detailed user map with pain point we will focus on. On sunlight interference with developing adaptive brightness. On small icons which require high precision such as enlarge hit zones.

Yi: This is a chart show key moments from setup to daily use. Please use the link on the slide to explore the details too.
These are our current product and user research.
I'm going to pass it to Jo for prototypes of hardware.

Project Goal:

Yi: Thanks to Jo and Bev for the insight of prototyping.

So, our project goal is to enhance comfort, accuracy, and adaptability of gaze interaction in real-world conditions. After mapping Jessica's full user journey, we finalized the key pain points to focus on. Our **top priorities** are improving **sunlight adaptability**, since outdoor glare breaks calibration, and **interface usability**, where small icons make gaze control difficult. Additionally, we identified **fatigue reduction**, **auto-calibration**, and **adaptive dwell time** as nice-to-have improvements for comfort and long-term use.
I'm going to pass it to Bev for our next steps.

Yi Timeline

Topic: Small icons → require high precision → cause gaze fatigue

Timeline

Oct 28-Nov 6th – User Research & Problem Framing

Goal: Understand when and why small icons cause fatigue and inaccuracy.

1. Analyze Tobii gaze data (heatmaps) to identify precision issues.
2. Conduct research: about accuracy, lighting, and comfort.
3. Map user pain points through the gaze journey.
   Deliverable: Visual pain-point map + hypotheses (e.g., icon size vs. fatigue).

Nov 4th- Nov 13th – Design Exploration (Low-Fi Concepts)

Goal: Generate multiple UX strategies to address precision.

1. Enlarged Zone: Icons expand or highlight when gaze approaches.
2. Magnet Effect: Gaze cursor "snaps" to the nearest target.
3. Cluster Redesign: Fewer, larger grouped icons instead of dense layouts.
4. Progressive Reveal: Longer fixation = gradual enlargement or glow.
   Deliverable: 3–4 concept sketches or storyboards (Figma/hand-drawn).

Nov 12th- Nov 20th – Prototype Simulation 1

Goal: Create an interactive mockup to test gaze-based interactions.

1. Build a **Figma prototype** or simple **web/Unity demo** simulating gaze hover (using mouse as substitute).
2. Experiment with icon size, spacing, and feedback types.
3. Coordinate with Jo & Bev to connect future hardware input (gaze coordinates).
   Deliverable: Clickable prototype showing different gaze-control strategies.

Nov 19th- Nov 26th – Prototype Simulation 2 with User Testing

Goal: Evaluate how design changes affect accuracy and fatigue.

1. Design short interaction tasks (e.g., select icons).
2. Record error rate, dwell time, and user fatigue rating.

3. Visualize results with heatmaps or bar charts.
   Deliverable: Usability report + data visualization comparing conditions.

Nov 26th- Dec 5th – Refine and Prepare final

Goal: Synthesize insights into practical recommendations.

1. Summarize findings into a **“Gaze-Friendly UI Guideline.”**
   Example categories:
   - Minimum icon size
   - Activation delay timing
   - Feedback cues (visual, auditory, or haptic)
2. Present before/after mockups.
   Deliverable: Final design poster + documentation for presentation.

Small icon research

Small Icons → High Precision → Gaze Fatigue

Objective

Understand when and why small UI icons cause inaccuracy and fatigue in gaze interaction

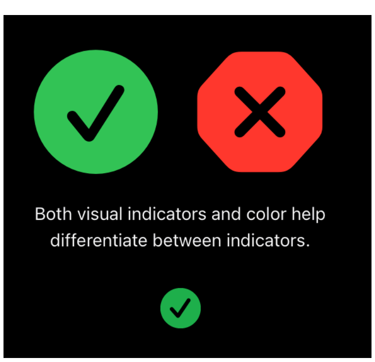
Similar Ideas and Products

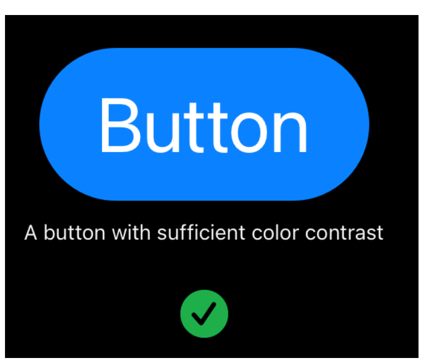
https://apps.apple.com/us/app/eye-gaze-communication-board/id6446154016

https://developer.apple.com/design/human-interface-guidelines/app-icons#Specifications
iOS, iPadOS
Default size 17 pt
Minimum size11 pt

Strive to meet color contrast minimum standards. To ensure all information in your app is legible, it's important that there's enough contrast between foreground text and icons and background colors. Two popular standards of measure for color contrast are the Web Content Accessibility Guidelines (WCAG) and the Accessible Perceptual Contrast Algorithm (APCA). Use standard contrast calculators to ensure your UI meets acceptable levels. Accessibility Inspector uses the following values from WCAG Level AA as guidance in determining whether your app's colors have an acceptable contrast.

Text size	Text weight	Minimum contrast ratio
Up to 17 pts	All	4.5:1
18 pts	All	3:1
All	Bold	3:1

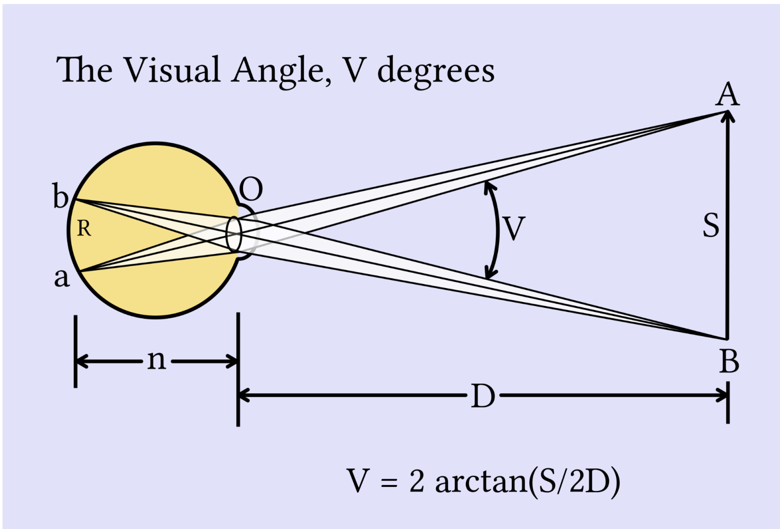




Related Concept

1. Icon size/Visual angel(°)

   Visual angle is the angle formed by two rays of light drawn from the extreme points of an object to the center of the eye, measuring the apparent size of an object as perceived by the viewer. It is a key concept in vision science because it is independent of the object's actual size or the viewer's distance, which means a small object far away can have the same visual angle as a large object close up.



   0.6° → Very small ICONS (similar to small buttons on the toolbar)
   1.0° → Common desktop icon sizes
   1.6° → The recommended lower limit for barrier-free design
   2.4° → Large ICONS, obvious and easily eye-catching

2. Contrast Ratio Standard in WCAG

Element Type	Minimum Contrast	Recommended (Enhanced)
Normal text	≥ 4.5 : 1	7 : 1
Large text (≥18 pt)	≥ 3 : 1	7 : 1
Icons / non-text graphics	≥ 3 : 1	Ideally 7 : 1

Testing High vs Low contrast allows us to observe:

- whether high contrast (7:1) improves gaze precision and reduces fatigue;
- how low contrast (3:1) under bright light makes icons harder to focus on;

- interaction effects with icon size and lighting. e.g., low contrast + small icons cause the most mis-selections.

3. Cheatsheet

- 1° visual angle ≈ **1.05 cm** at 60 cm distance.
- Icon size in pixels = (° × cm/°) × PPI / 2.54.
- Example @ 110 PPI: **1.6° ≈ 1.68 cm ≈ 73 px**.

Spacing

- Recommend **≥0.5° edge-to-edge** between small icons.
- Snap zone **≥ icon size** for assistive dwell.

Research Questions

- RQ1: How does **icon size** affect selection accuracy and dwell time?
- RQ2: How do **spacing** and **neighbor interference** impact mis-selections?
- RQ3: How do **lighting conditions** (glare/sunlight vs indoor) and **contrast** affect gaze precision and comfort?
- RQ4: Which **selection method** (dwell vs switch/confirm key) yields less fatigue for small targets?
- RQ5: How do **session length** and **icon familiarity** influence fatigue over time?

Experimental Design

Participants: 6–10 low-vision + 6–10 sighted controls (or start with pilot N=6).
Device: Tobii (Windows), IR on; distance ≈ 60 cm; single 15–24" display.

Factors (manipulated):

- Icon size (°): **0.6°, 1.0°, 1.6°, 2.4°**
- Spacing (center-to-center): **+0.4°, +0.8°, +1.2°** beyond icon edge
- Contrast: **High (WCAG~7:1)** vs **Low (~3:1)**
- Lighting: **Indoor (~300–500 lux)** vs **Bright (~2,000+ lux)** or simulated glare
- Selection: **Dwell (600/800/1000 ms)** vs **Confirm key**

Tasks: Serial target selection (randomized icon grid); block length 30–60 trials; rest between blocks.
Counterbalancing: Latin square for factor order; calibrate before each block.

Area of Interests (AOIs): Each icon + boundary zones; peripheral “empty” AOIs for off-target fixations.

Initial Hypotheses (to validate)

- **H1 Size**: Targets **<1.0°** yield ≥2× error vs **≥1.6°**.

- **H2 Spacing**: **Edge‑to‑edge <0.5°** increases neighbor mis‑selections by ≥30%.
- **H3 Lighting/Contrast**: Low contrast or bright glare increases TTS by ≥20% and fatigue ratings by ≥1 pt.
- **H4 Selection Method**: **Confirm‑key** reduces accidental selections vs dwell ≤800 ms on small targets.
- **H5 Learning**: Familiar icons reduce TFF over sessions but not spacing‑related errors.
- **H6 Adaptive Enlarge**: **Gaze‑triggered enlargement (×1.5–2.0)** cuts errors by ≥25% at ≤1.0° sizes.

Metrics & Instruments

Performance

- Accuracy (% correct selections)
- Time to First Fixation (TFF) on target (ms)
- Time to Select (TTS) (ms)
- Number of corrective saccades (#)
- Off‑target fixation time (ms)
- Mis‑selection distance (°) from target center

Physio/Behavioral

- Pupil diameter change (Δ) / Blink rate (if available)
- Head movement count (mm or inferred)

Subjective

- Perceived accuracy & effort (7‑pt Likert)
- Fatigue (short NASA‑TLX or 5‑item custom scale)
- Comfort notes (free‑text)

Visual Pain-Point Map

Small Icons Pain-Point Map

	Find (Search / Locate)	Acquire (Refine / Align)	Confirm / Select	Recover (Undo / Repeat)
Pain points	Small icons (<1.0°) often missed or confused in visual clutter.	Tight spacing causes gaze drift and "neighbor pull."	Dwell time too short → accidental selections.	Re-aiming after an error increases time and eye strain.
Signals (from gaze data)	↑ TFF (time to first fixation) Dispersed heatmap density.	↑ Corrective saccades. Off-target dwell time ↑	↑ Mis-selections. Short off-target fixations.	↑ TTS (time to select). ↑ Pupil Δ, ↑ blink rate.
Design Opportunity	Increase icon size / contrast. **Add preview magnification** or **highlight-on-gaze.**	**Add spacing (≥0.5°).** Implement gaze-sticky snap zone for fine control.	**Use confirm-key** or adaptive dwell. **Enlarge-on-gaze** for visual assurance.	Provide **undo / repeat affordance.** Enlarge repeated targets automatically.

Figma

Designing the Eye-Gaze App Interface (for Jessica)

Designing Ideas the Eye-Gaze App Interface (for Jessica)

Goal & User Context

The app is for **Jessica**, who uses **eye-gaze control** due to low vision or mobility limitations. The interface should let her:

- Use **eyes only** to navigate and select content (no mouse or touch).
- Avoid **fatigue, mis-selections, or re-aiming**.
- Receive clear **visual and audio feedback** after every action.
- Feel **comfortable and confident**, not strained.

Design Principle we must have

Principle	Meaning	In the UI
Large target size (≥1.6°)	Small icons (<1°) cause fatigue and inaccuracy	Buttons ≥70–80 px, spacing ≥0.5°
High contrast (≥7:1)	Supports low vision & glare environments	Dark background + bright icons
Clear feedback	The user knows what's selected	Gaze-triggered highlight or zoom
Two-step selection	Prevents accidental clicks	Gaze to highlight → gaze again to confirm
Minimal layers	Avoids complex navigation	Only 3–5 main buttons on home screen
Rest mode	Reduces eye strain	Pause gaze tracking automatically after idle

Layout (4:3 or 16:9)

Visual feedback examples

Gaze state	Visual cue
Start gazing	Soft blue halo appears
Holding gaze	Halo shrinks over 600–1000 ms (countdown)
Confirmed	Icon enlarges (×1.2) + sound cue
Canceled / error	Red highlight or tone
Rest mode	Dim background or “Take a break” overlay

Features

Home Screen:
3–5 large buttons (Music, Video, Communication, Settings).
Assistive Toolbar:
Magnifier, contrast switch, pause tracking.
Settings:
Adjust dwell time (600–1000 ms), icon size, spacing, color theme.
Feedback & Logs:
Show dwell time, errors, and fatigue level (for later tuning).

Visual Style

Lo-Fi ideas

Lo-Fi Prototype

Figma: [Figma Prototype]

Core Design Principles

Principle	Meaning	UI Implementation Specification	Rationale
Large Target Size	Small icons cause fatigue and inaccuracy.	Buttons ≥ 160px (on a standard iPhone 16Pro screen) with generous spacing. The main 2x2 grid buttons are approximately 160px x 200px.	Maximizes the area for the eye-gaze tracker to register intent, significantly reducing mis-selections and eye muscle strain.
Minimal Layers	Avoids complex navigation and cognitive load.	Home Screen: Only 4 main buttons in a simple 2x2 grid. Flat, direct navigation hierarchy.	Reduces the need for complex eye movements and memory load, minimizing cognitive and visual fatigue.
High Contrast	Supports low vision and glare environments.	Contrast Ratio ≥ 7:1 (WCAG AAA standard). Dark Background (#0A0A0A) with Bright White/Neon Blue elements.	Improves readability and minimizes light emission, which is a major contributor to visual fatigue during prolonged use.
Clear Feedback	The user must know what is being selected.	Gaze-triggered highlight (Soft Blue Halo) and Icon Enlargement on confirmation. Distinct Audio Cues for all state changes.	Provides immediate, unambiguous confirmation of the system's state and the user's intent, building user confidence.
Two-Step Selection	Prevents accidental clicks.	Step 1 (Highlight): Gaze on target. Step 2 (Confirm): Hold gaze for a specified dwell time (600–1000 ms).	Separates the act of looking from the act of selecting, requiring deliberate intent and preventing the "Midas touch" problem.
Rest Mode	Reduces eye strain and prevents accidental	Automatic Pause: Gaze tracking pauses after a period of user-defined idle time. Visual Cue: Dim background or "Take a break" overlay.	Encourages necessary breaks and ensures the interface remains stable when the user is not actively

	input during breaks.		engaging.

3. Interface Structure and Component Specification

The interface is structured into a primary Home Screen and a persistent Assistive Toolbar.

3.1. Home Screen Layout (Wireframe)

The Home Screen uses a simple 2x2 grid to maximize the size and spacing of the primary targets.

Element	Dimensions (Width x Height)	Target Size (Minimum)	Notes
Main Buttons (4)	Approx. 160px x 200px each	160px x 200px	Music, Video, Communication, Settings. Icons are centered and large.
Horizontal Spacing	16px	N/A	Clear separation between targets.
Assistive Toolbar	375px x 84px	N/A	Located at the bottom, outside the main content area.

3.2. Assistive Toolbar

The toolbar provides essential controls that can be accessed quickly without navigating away from the main screen.

Button	Function	Target Size	Rationale
Magnifier	Toggles on-screen magnification.	80px x 80px	Aids in viewing fine details or small text in sub-screens.
Contrast Switch	Toggles between color themes.	80px x 84px	Allows the user to instantly adjust the visual environment for comfort.
Pause Tracking	Manually triggers Rest Mode.	80px x 84px	Provides immediate control to stop accidental input when the user needs to look away.

3.3. Visual Feedback System (Gaze-Selection Loop)

The feedback system is crucial for the two-step selection process, providing continuous status updates to the user.

State	Visual Feedback	Audio Feedback	Timing
Start Gazing	Soft Blue Halo appears around the target icon.	Soft, low-frequency 'tick' sound.	Immediate upon gaze detection.
Holding Gaze (Dwell)	Halo shrinks (visual countdown) over the dwell time.	Continuous, subtle 'hum' or 'pulsing' sound.	600–1000 ms (user-defined dwell time).
Confirmed Selection	Icon enlarges (×1.2) briefly.	Clear, high-frequency 'chime' or 'click' sound.	Instantaneous upon dwell time completion.
Canceled / Error	The Target icon flashes a Red Highlight briefly.	Distinct, low-frequency 'buzz' or 'tone'.	Instantaneous upon gaze moving away or system error.

Report: Eye Tracking Communication Apps for iPhone

Report: Eye Tracking Communication Apps for iPhone

Primary Requirements

User: Jessica - Tobii EyeGaze quadriplegia user
Priority: Accuracy above all else
Device: iPhone with eye tracking capability

Core Features Needed

1. **Keyboard communication board**
2. **Dedicated storage area** for:
   - Words
   - Sentences
   - Paragraphs
3. **System-wide functionality** across:
   - Video recording
   - Phone calls
   - FaceTime
   - Zoom
   - Other apps

Research Categories

1. iOS Eye Tracking Capabilities

- Native iOS eye tracking features (iOS 18+)
- Hardware requirements
- Accuracy benchmarks vs. Tobii EyeGaze
- Calibration process

2. AAC Communication Apps Compatible with iOS Eye Tracking

Search for apps with:

- Full keyboard access
- Text storage/phrase banks
- System-wide access/integration
- Eye tracking optimization

Candidates to investigate:

- Proloquo4Text
- TD Snap
- Predictable
- CoughDrop
- TouchChat
- Grid for iPad
- Others with iOS eye tracking support

3. System-Wide Access Solutions

- iOS accessibility features for cross-app functionality
- Third-party utilities for universal access
- Integration with native iOS apps (Camera, FaceTime, Phone)

4. Accuracy Comparison

- User reviews: iOS eye tracking vs. Tobii
- Clinical/professional assessments
- Environmental factors affecting accuracy
- Calibration and customization options

5. Storage & Retrieval Features

- How each app handles:
  - Quick phrases
  - Frequently used sentences
  - Paragraph templates
  - Custom vocabulary organization

Research (Yi)

KEY FINDINGS ON ACCURACY

Critical Accuracy Comparison: iPad eye tracking showed a mean absolute error of 3.2° ± 2.0° compared to Tobii's 0.75° ± 0.43° PubMed Central - this means iOS eye tracking is currently **about 4x less accurate** than Tobii EyeGaze systems.

Important Considerations:

- iOS 18 eye tracking has accuracy issues that can make it frustrating to use, with recommendations to experiment with dwell time and sensitivity settings Spokenaac
- Glasses impact the accuracy of Apple's eye tracking, likely due to screen reflection Spokenaac
- There's variability in how well eye tracking functions across different apps Spokenaac

Device Requirements:

- iOS 18 eye tracking only works on iPhone 12 or newer, or iPad 10 or newer Spokenaac

---

SYSTEM-WIDE FUNCTIONALITY

Good News:

- Eye Tracking works across iPadOS and iOS apps and doesn't require additional hardware or accessories Apple
- Eye Tracking works seamlessly across iPadOS and iOS apps without needing additional hardware Ultralytics
- This means Jessica can use eye tracking in FaceTime, Zoom, Camera, Phone, and other native apps

How It Works:

- Eye Tracking uses Dwell Control to activate elements by holding your gaze on buttons or areas Apple Support
- Eye Tracking enables scrolling and navigation by moving eyes to desired areas, with a dwell timer activating when focusing on buttons iGeeksBlog

---

AAC APPS WITH KEYBOARD + PHRASE STORAGE

Based on the research, here are the top candidates that match Jessica's needs:

1. Proloquo4Text (Strong Match)

Keyboard: ✅ Full keyboard
Phrase Storage: ✅ Extensive storage system

- Quick Blocks for saved phrases
- Categories for organization
- Sentence prediction from history

Eye Tracking Support: Compatible with native iOS 18 eye tracking and Made for iPad certified eye tracking cameras AssistiveWare

System-Wide Use: Supports multitasking mode for iPad, allowing communication while using another app AssistiveWare

Additional Features:

- Single-screen layout with word and sentence prediction AssistiveWare
- Can switch between conversations without losing typed text App Store
- Over 150 voices available
- Works on iPhone, iPad, and Apple Watch

Cost: Premium app (one-time purchase, typically $119-149)

---

2. TouchChat HD with WordPower

Keyboard: ✅ Customizable keyboard
Phrase Storage: ✅ Custom pages and buttons with phrases

Eye Tracking Support: Works with head tracking on iPad Pro and iPhone X or beyond by enabling it in TouchChat settings Aaccommunity

- Note: This source mentions head tracking; iOS 18 native eye tracking should also work

Features:

- Full-featured AAC solution with customizable vocabulary files containing words, phrases, and sentences SimpliHere
- Can import/export vocabulary files

Cost: Expensive (typically $299.99)

---

3. Spoken - Tap to Talk AAC

Keyboard: ✅ Keyboard available
Phrase Storage: ✅ Storage system for frequent phrases

Eye Tracking Support: Designed to work with iOS 18 eye tracking [Spokenaac]

Features:

- Specifically optimized for iOS eye tracking
- Feedback from developers suggests good compatibility

Cost: More affordable option

---

4. APP2Speak

Keyboard: ✅ Text-to-speech functionality
Phrase Storage: ✅ Pre-set and user-defined messages with customizable phrase banks

Eye Tracking Support: Works seamlessly with AAC apps like APP2Speak for hands-free communication using iOS Eye Tracking [APP2Speak]

Features:

- Customizable communication boards with phrases connected to personally selected photos [APP2Speak]
- Simplified setup compared to traditional AAC devices

Cost: Lower cost option

RECOMMENDED RESEARCH ACTION PLAN

Phase 1: Accuracy Testing (CRITICAL) Since accuracy is Jessica's top priority, and iOS is significantly less accurate than Tobii:

1. **Borrow or test** an iPhone 12 or newer with iOS 18
2. **Calibrate carefully** in good lighting without glasses if possible
3. **Test keyboard typing** for at least 15-30 minutes to assess if accuracy is sufficient
4. **Compare to her current Tobii experience**

Phase 2: App Trials If accuracy is acceptable:

1. **Start with Proloquo4Text** - Best match for her needs
    - Free trial available through app evaluations
    - Strong phrase storage system

    - Single-screen keyboard layout
2. **Test Spoken or APP2Speak** - More affordable alternatives
3. **Try TouchChat** - If budget allows and more robust features needed

Phase 3: System Integration Testing

- Test eye tracking while:
    - Recording videos
    - Making FaceTime calls
    - Using Zoom (may require testing)
    - Making phone calls

POTENTIAL CONCERNS & SOLUTIONS

Concern 1: Accuracy Gap

- **Issue:** iOS is 4x less accurate than Tobii
- **Solutions:**
    - Adjust sensitivity and dwell time settings
    - Use larger keyboard layouts
    - Enable "Zoom on Keyboard Keys" feature
    - Ensure optimal positioning (1.5 feet from screen)

Concern 2: Glasses

- **Issue:** Glasses impact eye tracking accuracy [Spokenaac]
- **Solution:** Test with and without glasses; may need contact lenses for best results

Concern 3: App Compatibility

- **Issue:** Need to verify system-wide access in Zoom
- **Solution:** Since iOS eye tracking is system-wide, should work, but test before committing

---

REVISED APP RECOMMENDATIONS

Since ALL AAC apps work hands-free with iOS 18 eye tracking, here are the best matches for Jessica's needs:

TOP RECOMMENDATION: similar with Proloquo4Text

Why it's best for Jessica:

- ✅ **Full keyboard layout** (similar to her current Tobii board)/Honebomb types?
- ✅ Extensive phrase storage with "Quick Blocks"
- ✅ Clean, uncluttered single-screen design
- ✅ Word prediction to reduce eye movements
- ✅ Works system-wide (FaceTime, Zoom, etc.)
- ✅ **Completely hands-free with iOS eye tracking + dwell control**

How she'd use it:

1. Looks at letters to type → dwells to "click"
2. Looks at saved phrases → dwells to select
3. Looks at "Speak" button → dwells to speak
4. Can switch to other apps and keep typing in background

Research (Bev)

Accessible iOS Communication Apps

The research identified several key Augmentative and Alternative Communication (AAC) apps that meet Jessica's criteria, with Proloquo4Text emerging as the most promising candidate.

Proloquo4Text (AssistiveWare)

- **Non-Traditional Sized Keyboards/Enlarged Icons:**
    - Offers a Large Icons option in its Appearance settings to make buttons larger and easier to touch and see.
    - Supports changing font size and color and background color on the Text Pad and Quick Blocks for high contrast.
    - The companion accessible keyboard, Keeble, offers an Extended QWERTY (with a numbers row) or a QWERTY with larger buttons on iPad Pros, and allows for custom color themes to control contrast.
- **Keyboard Communication Board and Phrase Storage:**
    - It is a text-to-speech app designed for rapid, flexible, text-based communication, laid out on a single screen to reduce typing effort.
    - Features fully customizable Phrase storage (known as Quick Blocks). Users can type and save any phrase, sentence, or collection of sentences for fast access in conversation.
    - The Quick Blocks can be used as a communication board, with dedicated areas for saving and storing words, sentences, and paragraphs.
- **Compatibility with Video Calling/Recording:**
    - The app is explicitly designed to be used during phone and FaceTime video calls.
    - Users can simply start the call (phone or FaceTime), launch Proloquo4Text, and use it to speak. The app's speech output is routed through the call.
    - This functionality strongly suggests compatibility with other video conferencing apps like Zoom when used on an iPhone or iPad.

Other Relevant Apps

1. **TouchChat HD - AAC With WordPower**: Includes integrated head tracking (with applicable iPad/iPhone models) and a range of touch access features.
2. **TD Snap**: A full-featured communication app for individuals with cerebral palsy, among other conditions.

3. Speech Assistant AAC: Allows users to create categories and save phrases on buttons for quick access and easy communication, and also allows typing messages using the iOS keyboard.

App Option	**Pros (Meets Jessica's Requirements)**	**Cons (Potential Limitations)**
Proloquo4Text	**Excellent Model for Custom App:** Text-based AAC with a focus on typing. **Meets all core needs:** Dedicated, customizable phrase storage (Quick Blocks). **Video Call Compatible:** Specifically designed to route speech output through FaceTime and phone calls. **Accessibility:** Offers Large Icons and high-contrast options.	**Cost:** Typically a premium, one-time purchase app. **Learning Curve**: While text-based, the Quick Blocks system requires initial setup and organization.
TouchChat HD - AAC With WordPower	**Access Methods:** Includes integrated Head Tracking and a range of touch access features, which could be an alternative to eye-gaze. **Comprehensive:** Uses the widely-respected WordPower vocabulary for symbol-based communication.	**Symbol-Based:** Primarily a symbol-based app, which may not align with Jessica's preference for a keyboard communication board (though text-entry is possible). **Cost:** High-end, premium app.
Speech Assistant AAC	**Simplicity & Storage:** Allows creation of categories and saving phrases on buttons for quick access (meeting the storage/communication board need). **Cost:** Generally more affordable or has a subscription model.	**Basic Keyboard:** Relies on the standard iOS keyboard for typing, which lacks the non-traditional sizing and enlarged icons of a dedicated AAC keyboard like Keeble (a companion to Proloquo4Text). **Video Call Compatibility:** Less explicit documentation on routing speech during video calls compared to Proloquo4Text.
Native iOS Eye Tracking (The tool Jessica currently finds inaccurate)	**Free & Built-in:** No additional cost or installation required. **Device Control:**	**Accuracy:** Low Precision compared to dedicated eye-gaze devices (like Toby),

	Allows control of the entire iOS device, not just one app.	leading to the "looking at one thing, selecting another" issue Jessica described. **Not Optimized for AAC:** It is a general control feature, not specifically designed for the rapid, high-stakes communication of an AAC app.

Dec 2nd Prototype demo

iSee demo.mov

https://www.figma.com/design/djyWW2wGiASZ8g0Iuf5ggn/Jessica-Tobii-User-Journey-Map?node-id=0-1&t=YUIJZ8LzAU88t6iM-1

https://www.figma.com/proto/djyWW2wGiASZ8g0Iuf5ggn/Jessica-Tobii-User-Journey-Map?page-id=0%3A1&node-id=166-12118&viewport=-5363%2C-3074%2C0.5&t=QPMU1ypYFYMqyMtb-1&scaling=scale-down&content-scaling=fixed&starting-point-node-id=50%3A249

📚 Midterm

LEAVE BLANK

📚 Midterm Task Breakdown

Task Delegation

Divide the project into segments that each person handles to be stored here ?
Create a timeline to establish when certain tasks are done here or wherever convenient.

Thurs, Oct 9

DUE:

5. Updates on research + communication with anyone (hardware / software / design spaces) = any leads??
6. Yi: Email Jessica + market research on existing similar prototypes / products

TODO:

6. Jo: some sort of 3D model done/ scope out things that need to be printed
7. Bev: a pretty good idea on what software works with what hardware + the technical feasibility of implementing with the timeline we have
8. Yi: User Journey + Competitive Analysis (market research)

Thurs, Oct 16

DUE:

5. Report on your TODOs from last week
6. Jo: coming up with model + drawing
7. Bev: look more in depth into software (green)
8. Yi: user journey + competitive analysis (market research)

TODO:

5. Edit Slides Here
6. Write script (2 mins per person) for your part
7. Record slides + Add your own captions + A transcript of your part (**DUE SATURDAY**)
    a. **Midterm Instructions Slides - CHECK THIS BEFORE RECORDING**
    b. END WITH "I'm going to pass it on to x" etc.
    c. Record through zoom with captions (QC IT FIRST)
8. Edit Recordings + Final Edits (**DUE SUNDAY**)
    a. Bev

Tues, Oct 21 ➡️ Midterm !!

Presentation Flow:

Yi - Interaction Design (2 mins)

- User Flow/ Pain points/ User Journey

- Competitive Analysis

Jo - Hardware (2 mins)

- Hardware concepts / cheapest options / CAD design of a prototype
- Timeline/ next steps for final prototype (hardware side)
- Speak about people you've reached out to + most recent updates

Bev - Software (2 mins)

- Software POC demo + next steps/ timeline + how software connects to hardware

Yi (2 mins)

- Finalize/ prioritize the user pain points that we want to focus on
    - Sunlight (1), Small icons (2) - high priority
    - 2-3 nice to haves

Bev then Jo (1 min each)

- Explain next steps on what our proof of concepts will resolve

---

Jo - Hardware

3. Hardware concepts / cheapest options / CAD design of a prototype
4. Reach out to relevant companies
    a. TobiEyeGaze
    b. Meta (Matthew - Accessibility Lead)
        i. Let's schedule a call all together after the Oct 9's meeting.
    c. Apple

Bev - Software

3. Software concepts + algo / existing options / scope of technical feasibility
4. Reach out to relevant professionals
    a. TobiEyeGaze
    b. Apple Accessibility (Accessibility Support: https://support.apple.com/en-us/111749)

Yi - Interaction Design

4. Email Jessica and keep her up to date with our talk with faculty Oct 7, 2025
    a. Expectations on end deliverables
    b. Our midterm presentation date and what we plan on accomplishing from today till our midterm date
    c. Is she okay with that?
5. Reach out to the experts working on vision and user interface (Jessica cannot use devices outside in the sunlight: high contrast etc.)
6. Research/Reach out to the users who use eye gazing

Midterm Deliverables/ Approx Content We Will Cover

4. Who we've reached out to + who did not respond
5. What we learned from people (user, designers, stakeholders) we spoke to
6. What we learned from our research - hardware / software / interaction design

✒️ Midterm Transcript

Full Transcript

[0:00:00] **Yi:** Hi, we're iSee Group. Our project explores an eye-gaze control system from an assistive technology perspective, focusing on our main client, Jessica, who has limited upper-body mobility. I'm Yi, I'm responsible for interaction design. My partners Bev is for software development and Jo for hardware design. Here is our product research and user research.

[0:00:25] **Yi:** Eye-tracking technology records where and how long a person looks using cameras that follow eye movements in real time. In assistive contexts, this allows users like Jessica to control digital interfaces hands-free, promoting independence and accessibility in daily life. The image shows a person using a Tobii eye-tracking system on a computer monitor. The screen displays a product comparison interface with a green-to-red heat map overlay showing where the user's gaze is focused. Tobii eye tracker is mounted below the monitor, connected by a USB cable.

[0:01:03] **Yi:** We analyzed existing eye-tracking systems such as Tobii and PCEye. Most provide reliable gaze control, but they struggle with outdoor lighting, precise calibration, and user comfort.

[0:01:16] **Yi:** We created user journey and workflow maps to understand Jessica's real-world experience with the Tobii eye tracker.This is a horizontal user journey map for Jessica using eye tracker. It shows six stages while using the tech includes feelings, actions, and design opportunities for improvement. You can use the link on the slide to explore the full details.

[0:01:23] **Yi:** This is a user flow mapping Jessica's interaction with the Eye Tracker the slide to explore the full maps in detail. It's a detailed user map with pain point we will focus on. On sunlight interference with developing adaptive brightness. On small icons which require high precision such as enlarge hit zones.

[0:02:00] **Yi:** This is a chart show key moments from setup to daily use. Please use the link on the slide to explore the details too.These are our current product and user research.

[0:02:09] **Yi:** I'm going to pass it to Jo for prototypes of hardware.

[0:02:14] **Josephine:** Thank you Yi for that wonderful overview on user research. In the next 4 slides, I will focus on the hardware side, turning those insights into our compact phone base eye tracking pen.

[0:02:28] **Josephine:** Our community partner, Jessica, has been using the Toby PCEye tracker for years depicted in the top right hand corner. In order to meet her wants, we are

exploring a smarter- sleeker design that resembles the size of a pen depicted in the bottom right hand corner.

[0:02:46] **Josephine:** The CAD model demonstrates how easy it'll be to modify the dimensions, such as the camera diameter and the wall thickness without redrawing the entire thing. The image shows a simple base geometry that's going to support the small camera module, microcontroller and IR LEDs.

[0:03:10] **Josephine:** We want to accomplish three things out of this physical prototype: a smaller design to reduce bulk, multi-device compatibility with the addition of the USB Type C cable, and most importantly - something that has an anti-reflective coating to reduce glare and improve pupil detection outdoors.

[0:03:30] **Josephine:** In this cost breakdown table we have limited the design to five components, all under $9 each. The provided enclosure and microcontroller will be absolutely free.

[0:03:42] **Josephine:** Now over to Beverly for the software.

[0:03:45] **Beverly:** Thanks, Jo, for the insight on the hardware components. Building on her work on hardware fabrication, my software prototype explores how users interact with the system, from calibration to gaze-based reading and feedback. My goal was to create a lightweight, browser-based eye-tracking solution for mobile, letting us test gaze interaction before hardware integration.

[0:04:06] **Beverly:** The prototype has three screens: First, users set up their webcam to lock onto and track their facial features. Second, they calibrate by following a 5x5 grid of red dots, holding each gaze for 3 seconds, ensuring accuracy and stability. Third, users gaze at blue word circles. Holding their gaze 1 to 2 seconds simulates using a keyboard with eye-gaze technology: word turns green, dot disappears, the word you successfully selected gets read aloud, giving users audio and visual feedback.

[0:04:37] **Beverly:** The prototype uses WebGazer.js: video captured, eye landmarks detected,coordinates smoothed, and dwell detection triggers feedback. The calibration section trains the model, simulating real gaze-controlled interfaces. Smoothing is also added to reduce jitter for accurate tracking.

[0:04:54] **Beverly:** Currently, this simulates gaze in the browser. Our next steps include connecting to hardware like ESP32, OpenMV, or Jetson, optimize calibration, and to test this on phones.

[0:05:07] **Beverly:** Next, I'll pass it to Yi, who will discuss prioritizing user pain points.

[0:05:10] **Yi:** Thanks for Jo and Bev for the insight of prototyping.

[0:05:15] **Yi:** So, our project goal is to enhance comfort, accuracy, and adaptability of gaze interaction in real-world conditions. We finalized the key pain points to focus on.

[0:05:24] **Yi:** Our top priorities are improving sunlight adaptability, and interface usability, where small icons make gaze control difficult. Additionally, we identified fatigue reduction, auto-calibration, and adaptive dwell time as nice-to-have improvements for comfort and long-term use.

[0:05:43] **Yi:** I'm going to pass it to Bev for our next steps.

[0:05:47] **Beverly:** Building on the pain points we identified, our software and hardware prototypes tackle these challenges by simulating eye-tracking on a phone.

[0:05:54] **Beverly:** For harsh sunlight, the software tests gaze accuracy under different lighting. For small mobile icons, it simulates adjustable target sizes. This lets us refine circle size, spacing, and dwell duration, ensuring users can reliably select items with their eyes in a controlled, low-risk environment before integrating hardware.

[0:06:12] **Josephine:** We decided to build this dedicated design instead of relying only on iPhone software, just because it gives us more control over accuracy and usability. The built in phone cameras aren't really optimized for eye-gaze tracking, especially in brighter variable lighting. But with our hardware, we can add infrared illumination, better optics and design for multiple devices, not just an iPhone. Now this is what has influenced the next steps in terms of the hardware. We're going to combine all those components mentioned earlier into a compact unit. We're gonna calibrate the gaze tracking for
[0:06:50] **Josephine:** accuracy under varied lighting. Then we're just going to test the performance in sunlight with a simulated gaze interaction task. And it's with this test that we will refine our software mapping and hardware alignment based on results

[0:07:06] **Josephine:** Well, I would like to thank you for listening to our presentation and coming along with us on this journey.

[0:07:012] **Josephine:** We look forward to presenting the final prototype. Thank you.

Midterm Feedback

Midterm Presentation Feedback

Group Name: iSEE: Eye-gaze tracker

Jessica_DAT Fall 2025, Midterm Feedback (Responses)

Project URL: https://wp.nyu.edu/ap_classes_dat_f25/isee/

Members: Yi Wang, Josephine Odusanya, Beverly Yip

Primary Project: iSEE

Grading Rubric:

	Excellent (great depth and breadth)	Good (strong start)	Fair (start evident but missing important factors)	Insufficient (needs extensive work to meet midterm expectations)	Group Score
Summary of Goals	15/20	/20	/20	/20	15
Timeline	13/20	/20	/20	/20	13
Work So Far (Literature Review)	17/20	/20	/20	/20	17
Presentation Style	20/20	/20	/20	/20	20
Clarity and Accessibility of content posted to website	20/20	/20	/20	/20	20

Grade (total score): 85

Feedback:

When you discuss this project, we think that you should start by talking about it in terms of use with the iPhone since that's where the problem is, unless things changed as you talked with Jessica. It wasn't clear why you were spending so much time with eye gaze in general.

It was a great idea to think about charging too. Not necessarily for you to find a solution but to acknowledge it with Jessica and to start thinking about directions it might go in.

Goals:

- Improve eye gaze in real-world situations. It was confusing that you talked about eye gaze separately from the phone. Is using the phone still the goal? It should be included in the goal. We see the goal as facilitating mobile phone use but modifying or developing an eyegaze system that controls access to the phone's features.

Timeline:

- Great idea to include the user journey and user flow. We don't see a timeline representing your work plan for the next few weeks in your presentation. You did a good job on the website.

Work so far:

- Good job presenting eye gaze technology. This section should also include information about how other, similar people use their phones. Any scientific literature supporting your decisions.

Presentation Style (including verbal presentation and visual aids):

- The slides showing the user maps were very difficult to see. Consider separating the information onto multiple slides so there is less information on each slide and so that the text can be larger. Our recommendation is to make the text as large as possible on the slides. Use the space you have to increase the text size on all of your slides. The prototype overview was also hard to see. We suggest you make everything as big as possible on the slides for your final presentation.
- Your presentation styles were very clear and natural. Well done.

Clarity and Accessibility of Website:

- GREAT Work. These are all minor comments:
- Your project mentor should be listed as Jessica. Jessica should also be added as a group member and she should share her description if she wants to.
- Your website is full of great information. As you move forward, please remember to update and revise your descriptions.
- Rather than using terms like "next week" or "2- 3 days", but the actual dates in your timeline. You can include a planned date and an actual date if you think it would be helpful.
- Try to keep up with alt text for image descriptions.
- Yi- your website is beautiful. As you learn about accessibility, consider applying it to your personal website too- consider alt text for image descriptions and color contrast for easier visual interpretation. I noticed that the light gray print on the beige background is especially hard to read. You can use a contrast checker to help you decide on color choices.

Suggestions--

- find out how other people access their phones. Contact the eyegaze makers and see what their suggestions and plans are.
- Consider adding charging for the phone in your design with the holder.

⚫ Archived

LEAVE BLANK

✉️ Follow up Email Jessica Oct 7th

To	jefrew01@gmail.com
Cc	Beverly Yip Josephine Odusanya Anita Perr Daniel Johnston
Bcc	Person
Subject	Quick update & midterm plan

Hi Jessica,

Following our faculty check-in today, I wanted to share our current plan and progress for the eye-tracking project and confirm that this direction aligns with your expectations.

Midterm Deliverables / Content Estimate (Oct 21)

- Presentation deck (6–8 slides) covering:
    - Research framing and objectives
    - Key insights from stakeholders (designers) interviews
    - Summary of technical exploration (hardware / software / interaction design)
    - Next steps toward prototype development
- Supporting notes or visuals (diagrams, early sketches, reference tests).

Who We Plan to Reach Out To

- Potential interviewees: accessibility researchers, assistive-tech users, designers with experience in adaptive interfaces.
- We're currently drafting outreach messages and identifying who to contact within the next week.

What We Aim to Learn (People: Users, Designers, Stakeholders)

- How people with limited mobility or other accessibility needs might use eye-tracking for control or interaction.
- What current challenges designers face when integrating gaze-based input.
- Stakeholder perspectives on feasibility, comfort, and ethical concerns.

What We Plan to Explore (Hardware / Software / Interaction Design)

- **Hardware:** Low-cost webcam setup vs. dedicated eye-tracking module.
- **Software:** Testing open-source frameworks for gaze detection.

- **Interaction Design:** Early concept sketches for gaze-triggered feedback or interface navigation.

We wanted to share this update to make sure we're aligned in our direction. Please let us know if you have any feedback or concerns about our current scope and midterm deliverables.

Best,
Yi, Bev, Jo

✉️ Email Jessica Oct 24th

Hi Jessica,

Hope you've been doing well! I wanted to share our midterm progress update with you, and would like to have your feedback.

We decided to build a dedicated hardware design instead of relying only on iPhone software, since this gives us more control over accuracy and usability. Our solution builds directly on the existing Tobii Eye Tracker setup and Windows-based control environment, ensuring compatibility and minimal disruption.

Hardware:

With our custom hardware, we can add infrared illumination, improved optics, and cross-device flexibility, making the system more reliable for low-vision users. The built-in phone cameras aren't optimized for eye tracking, especially in bright or variable lighting. With our custom hardware, we can add infrared illumination, improved optics, and cross-device flexibility, making the system more reliable.

Software:

We're testing eye-tracking performance under harsh sunlight to address calibration and visibility issues you've experienced outdoors. A gaze-interaction simulation tool helps us prototype adjustable target sizes and study how different visual contrasts affect usability. Through ongoing integration and calibration tests, we'll fine-tune how the overlay and hardware assist work together.

UI:

We're also working on the small icon selection problem, trying to create an enlarged touch zone to make it easier and more accurate to pick targets.

Progress Present:

Here's our presentation video and slides for detailed reference:
https://www.youtube.com/watch?v=rIoDeIn3XJw
iSee_Assistive Technology_Midterm Presentation

We'd really appreciate your feedback on both our current direction and next steps.

Best,
Yi, Bev, Jo

💌 Final Follow up with Jessica

Hi Jessica,

I hope you're doing well and having a restful end to the year.

I wanted to follow up and properly close the loop on our project together. First, thank you again for the generosity, openness, and insight you shared with us. Your lived experience shaped every major decision we made, and the project would not exist in the same way without your guidance.

As we reach the end of this semester, our work is concluding at the research and early design stage. Our documentation captures the core insights, design rationale, and potential future directions, particularly around camera placement, mounting flexibility, and potential user interaction with iPhone. These materials are intended to serve as a foundation that could be built upon in future iterations with additional time, resources, and technical support.
Final Project:https://wp.nyu.edu/ap_classes_dat_f25/isee/final-project/
Final presentation: iSee - Final Presentation Slides
Final presentation video:
https://stream.nyu.edu/media/iSee+Group_DAT+Final+Presentation/1_ydz5a8zk
Website:https://wp.nyu.edu/ap_classes_dat_f25/isee/
Weekly update:https://wp.nyu.edu/ap_classes_dat_f25/isee/isee-weekly-updates/

At this moment, we are not in a position to continue active development beyond this phase as a team. That said, the work is not meant to be an ending, but a handoff point. We truly hope the research and concepts can be useful to you or to others who may continue exploring mobile eye gaze solutions.

Thank you again for trusting us with your story and your needs. It has been an honor to learn from you, and we deeply appreciate the chance to work alongside you.

Wishing you all the best, and hoping our paths cross again.

Warmly,
Yi, Bev and Jo