**Periphery Deliverable 5**

Isaac Engler, Olivia Pooler, Jack Bisson, Markus Hiyazaki, Ebban Pudden

University of Ottawa, Department of Engineering

GNG 1103, Group 5

Professor Knox

October 27, 2024

**Table of Contents:**

[Design Drawing: 2](#_Toc467596466)

[Plan and Schedule 5](#_Toc910077240)

[Significant Project Risks and Associated Contingency Plans 5](#_Toc1198741250)

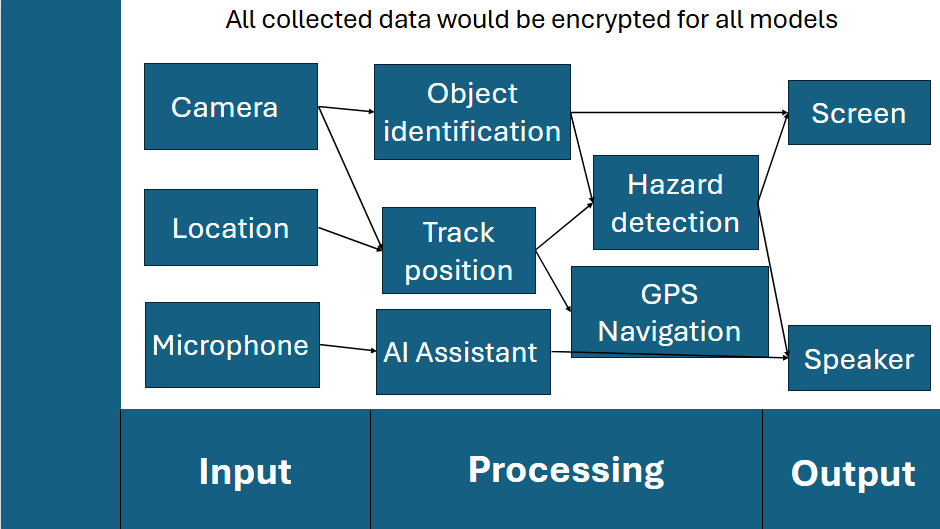
[Budgeting 7](#_Toc2035568497)

[Prototyping Test Plans 8](#_Toc1090465971)

# **Design Drawing**

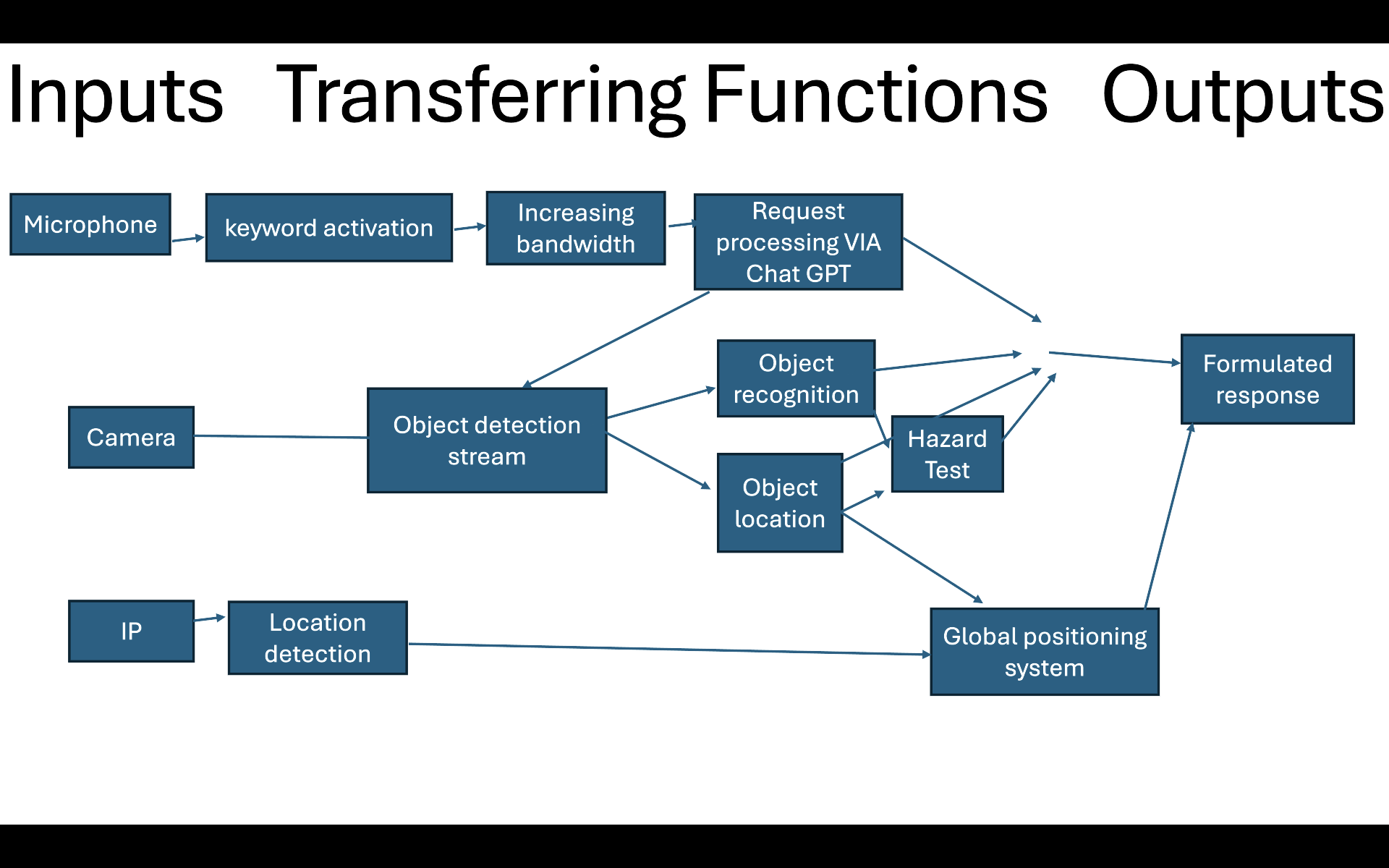
Our first design drawing is a flowchart that demonstrates the mode of input our device will take, how it will be processed and the mode of output the user will receive. This is our original, general design drawing:

**Drawing 1:** General design (older version)

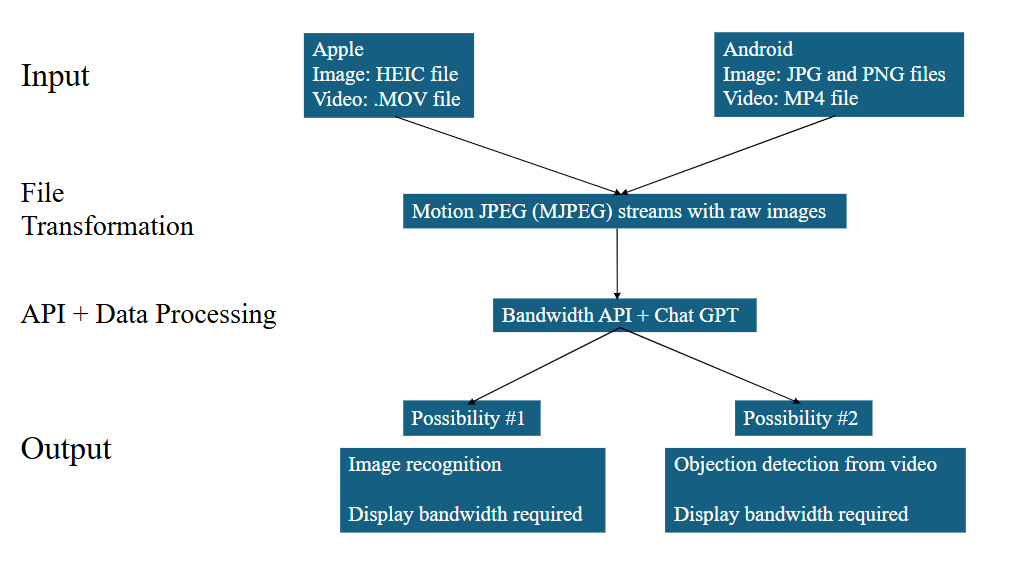


This drawing represents how we plan on using a microphone, camera, and IP inputs and return a formulated response. The microphone acts as a primary input and upon processing requests, the system will deduce whichever other inputs are needed.

**Drawing 2:** General design (updated version)

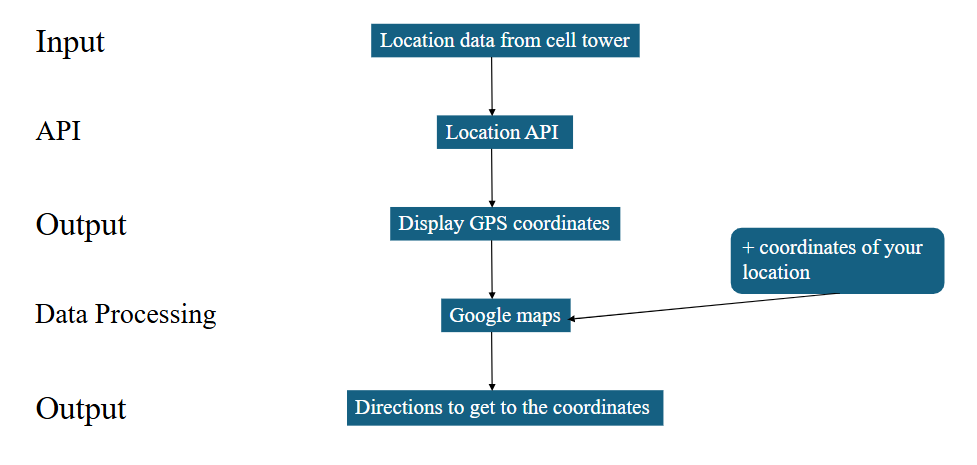


**Drawing 3:** Object detection stream (Bandwidth API)

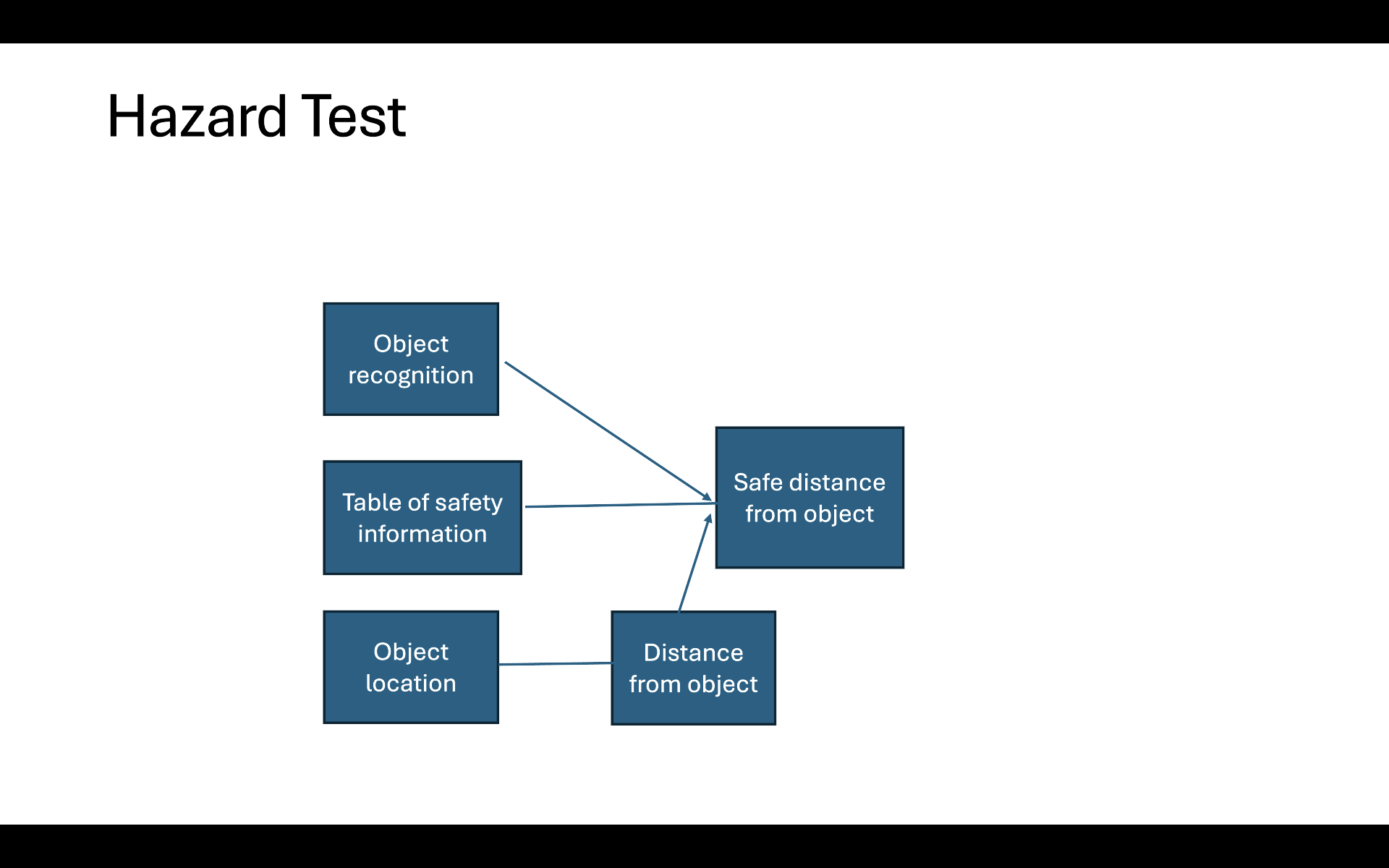


Our second design drawing is a more specific flowchart that contains the data our device will accept as an input, how it will be transformed via processing and the data that will be outputted.

**Drawing 4:** Object detection stream (Location API)

Using google Gemini AI to compare an object to pre-determined "dangerous" criteria (i.e. being too close to a fire), as inputs, shown by the two arrows in the general sketch, it also includes inputs from our generated pool of dangerous items. As an output, it returns a safe distance from the requested objects. If it is deemed safe, an indicator (green box) will be returned instead.

**Drawing 5:** Hazard test sub system



**Plan and Schedule**

**\*\*Table 1:** Plan and a schedule for prototyping and testing their solution to the client’s needs by the end of the semester

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Description** | **Estimated Duration** | **Tentative Dates** | **Team member responsible** |
| 1 | Figure out how to access the sandbox | 1 hour | 10-29-24 | Markus |
| 2 | Deliverable F - Prototype #1 | 6 hours | 11-3-24 | Everyone |
| 3 | Deliverable G - Prototype #2 | 6 hours | 11-10-24 | Everyone |
| 4 | Deliverable H - Prototype #3 | 6 hours | 11-17-24 | Everyone |
| \*5 | Code for Bandwidth API | 8 hours | 11-17-24 | Everyone |
| \*6 | Code for Location API | 8 hours | 11-17-24 | Everyone |
| 7 | Presentation for Client Elevator Pitch | 3 hours | TBD | Jack + Ebban |
| 7.5 | Practice Presentation for Client Elevator Pitch | ½ hour | TBD | Everyone |
| 8 | Client elevator pitch | 3 minutes | TBA | Everyone |
| 9 | Deliverable I – Design Day Presentation Material | 6 hours | 11-27-24 | Everyone |
| 10 | Deliverable J – Project Presentations | 6 hours | TBA |  |
| 11 | Practice Presentation for Design Day |  | 11-30-24  12-2-24 | Everyone |
| 12 | Deliverable K – User and Product Manual | 6 hours | 12-3-24 | Everyone |

\*Estimated time duration for coding portions are rough estimates as no team members are experienced coders therefore it is difficult to estimate how long coding process will take. \*\*For each deliverable, the team spends 4 hours on the rough copy (first iteration) and 2 hours on the final draft (second iteration) to improve it.

# **Significant Project Risks and Associated Contingency Plans**

The most technical part of this project will be putting together the code, so it runs with error and is able to work for the intended use cases depicted in the detail design drawings.

**Table 2:** Code for Bandwidth and Location API Contingency Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Scenario | Trigger | Response | Who to Inform | Key Responsibilities | Timeline |
| Leaked code | A team member leaked source code from the NetAware API (for example, placed code into ChatGPT). | Identify what code has been leaked, then the project manager (not Zak) will inform Shabodi staff and the lab project manager (Zak) of the situation and what the best course of action to take. | Lab project manager (Zak) and Shabodi staff. | Team member who leaked the code and project manager. | After every session with the NetAware API, team members will list what they worked on.  If a code leak is suspected, the project manager will inform the lab project manager and Shabodi staff within the hour (preferably). |
| Absent team member | A team member is abruptly absent without notice. | The project manager will attempt to contact the absent team members via their phone. If there is no response, a vote will be held on who will fill in the absent team members' role. | N/A | Project manager and all team members. | A team member is absent from a meeting and is not responding to messages on  Microsoft teams or email. The project manager will attempt to call the absent team members. If they don’t pick up and nothing is heard from them in over a day, a vote will be held on who will fill in for the roll of the absent team member. |
| Issues with API and/or code | One or more lines of code aren’t working as intended or are causing errors within the API. | Any code related issue will be directed towards Shabodi or if permitted, will instead be brought to the staff at the (ITI 1120) student mentoring center. | Shabodi or staff at the 1120 student mentoring center. | Engineer and/or whichever team member discovered the issue first. | A team member discovers an error or issue within the API and mentions it to the rest of the team. Upon further examination no one can resolve it, thus the engineer will mention to either Shabodi or student mentoring staff at a time most convenient for them. |

# **Budgeting**

The table and spreadsheet below contain information on how we intended to spend our twenty-five-dollar budget. Included are additional software, such as ChatGPT and Gemini AI, that we will need to purchase subscriptions for as well as all the materials required for the prototypes and final project.

**Table 3**: A criterion of how we will spend our budget.

|  |  |  |
| --- | --- | --- |
| Hardware/Software | Purpose | Cost |
| ChatGPT | Image processing and recognition. | $20 monthly - ChatGPT Plus (includes image analysis and file uploads). |
| Gemini | Navigation and image identification. | Free – However the number of requests will be limited to 1,500 per day. |
| Google Maps | Navigation. | $10 per every 1000 request – Places SDK (part of Goggle Maps API) includes geolocating and geocoding. |

**Spreadsheet 1:** Bill of Materials

|  |  |  |  |
| --- | --- | --- | --- |
| **Material** | **Type** | **Purpose** | **Cost** |
| Shabodi's Location API | API | Location | Free |
| Shabodi's Bandwidth API | API | Bandwidth rate adjuster | Free |
| Mozilla API | API | Keywords | Free |
| Google Gemini | API | Navigation | 26.99$ + First month free |
| ChatGPT | API | Object detection + Voice assistant | 20$/month, or 25$/month for team version |
| Paper | Drawings | Prototyping | Already have |
| Pen | Drawings | Prototyping | Already have |
| Android Device | Device | Testing | Already have |
| IOS Device | Device | Testing | Already have |

# **Prototyping Test Plans**

Below are our prototyping test plans which include a detailed summary of the purpose of each prototype, what we are measuring, what we are testing in the system, what type of protype we are creating and the estimated time duration for each.

The first prototype will explore the use and capabilities of Shabodi’s Bandwidth API, highlighting its ability to transmit data efficiently within the network.

**Table 4: Prototype #1 – Focused**

|  |  |
| --- | --- |
| **Functionality of Bandwidth API** | |
| Propose of prototype | Obtain numerical data for the bandwidth required for different size and types of files |
| Measurements needed | Bandwidth (bps) |
| Testing the whole system or a component | Component |
| What type of prototype | Numerical |
| Estimated time duration | 1 hour |

The second prototype will explore the use and capabilities of Shabodi’s Location API, highlighting its ability to locate the user with a high degree of accuracy in an efficient manner.

**Table 5: Prototype #2 – Focused**

|  |  |
| --- | --- |
| **Functionality of Location API** | |
| Propose of prototype | Test accuracy and compare coordinates given by Shabodi’s API versus 2 other credible sources (Google earth, GPS coordinates) |
| Measurements needed | Coordinates |
| Testing the whole system or a component | Component |
| What type of prototype | Numerical |
| Estimated time duration | 30 minutes |

The third prototype will explore the use and capabilities of integrating Shabodi’s Bandwidth API with ChatGPT, testing Shabodi’s ability to be integrated.

**Table 6: Prototype #3 – Comprehensive**

|  |  |
| --- | --- |
| **Integration of Bandwidth API and ChatGPT** | |
| Propose of prototype | Take controlled outputs from GPT without being connected to Shabodi and with being connected |
| Measurements needed | Latency measurement for time between input and output (ms) |
| Testing the whole system or a component | Half of the system |
| What type of prototype | Numerical |
| Estimated time duration | 30 mins (if software is set up beforehand) |