

**Name** : KAUTA MARVIN

**Registration Number** : S20B23/204

**Faculty/School** : SCIENCE AND TECHNOLOGY

**Course** : BACHOLERS OF SCIENCE IN COMPUTER SCIENCE

**Lecturer** : ***mr SIMON LUMomba***

**TECHNICAL PLAN**

**A MOBILE BASED APPLICATION TO FOR HELPING THE VISUALLY IMPARED USING NATURAL LANGUAGE PROCESSING AND ARTIFICIAL INTELLIGENCE**

**Introduction**

The world for normal human being is far different than visually impaired, due to either lack of vision or no vision. The difficulties in their daily routines can be minimized with help of technological support which is usually aids that can be used for travelling. Computer vision a field of artificial intelligence provides the assistance for helping impaired.

Information that peoples collect from their surroundings and the outside world is obtained through sight. visually impaired people suffer inconveniences in their daily and social life because of lack of sight. losing sight is one of the most unfortunate things that can happen to someone.

With the rapid increase in the total population, the number of visually impaired people is also increasing significantly.

Given the growth in the numbers of visually impaired (VI) people in low-income countries, the development of affordable electronic travel aid (ETA) systems and devices, sensors, and apps embedded in ordinary smartphones make a potentially cost-effective in terms of affordability allowing many to have these devices and systems.

While several systems and innovations like e-cane, guide dog, infrared-based cane, laser-based walker, Batcane , Guidecane, Lasercane, Mowat sensor, the Sonic guide, The VeDi system provides another showcase for indoor and outdoor navigation by integrating vision-based with pedestrian localization systems.( P. Chippendale,2014), and the Polaran (Gonzalez-Delgado,2016), the white cane, have been designed in developed countries like Colombia, Russia, United States of America, Few of these systems have been designed in Uganda like a smart walking cane which was made by a Ugandan engineer Zablon Mulero in Moroto regional referral hospital as by the National Drug Authority as reported by the observer newspaper by URN on November 13, 2019.

**SOLUTION**

**A mobile based application to for helping the visually impaired using natural language processing and artificial intelligence**

**TARGET GROUP: Partially blind and completely blind individuals**

1. The general objective of the suggested system for visually impaired people using voice recognition and image processing system is to make blind people more independent and become aware of their surroundings easily
2. Designing the compact size device that works on real time for visually impaired During navigation and daily activities
3. Cost effective system for understanding scenery.

**USER REQUIREMENTS CONCERNING IMPLEMENTATION.**

**GENERAL STRUCTURE OF THE PROPOSED MOBILE APPLICATION.**

This application will have three modules that are catering for specific groups of people in society namely;

* The partially blind
* The completely blind
* The helpers for the VIPs

**FUNCTIONALITIES FOR EACH OF THE GROUPS**

|  |  |  |
| --- | --- | --- |
| **PARTIALLY BLIND** | **COMPLETELY BLIND** | **THE HELPERS FOR THE VIPS** |
| This group of people will have both functionalities that suit the completely blind and the partially blind which include   1. Text recognition – ranges from reading articles, books, and other items that have text on them 2. Voice Assistant – this will help in the fast execution of instructions like sending messages and navigation through the app. 3. Object recognition features 4. object description features – helps in describing the surrounding of a person. 5. Speech-to-text features. These will help in speeding up interaction with the app. | 1. Text recognition – ranges from reading articles, books, and other items that have text on them 2. Voice Assistant – this will help in the fast execution of instructions like sending messages and navigation through the app. 3. Object recognition features 4. object description features – helps in describing the surrounding of a person. This uses captions for an area that are read out as speech feedback 5. Speech-to-text features. These will help in speeding up interaction with the app in the areas of sending messages plus accepting feedback during an interaction. 6. Speed dial features. – this involves contact for emergencies and sending messages to the contacts saved in the app 7. Wireless pairing functionalities to help them use other devices like Bluetooth to headphones and cameras to extended displays 8. Route navigation in the surrounding -helps the blind person know how if traffic is in the area nearing their location. | 1. User location features- these are used to see where exactly the user is and what is around them. 2. Directions and chat features 3. Monitoring movement features 4. Recording surrounding of user 5. Voice feedback texts. |

**DEVELOPMENT APPROACH**

**Rapid Application Development (RAD).**

In the RAD approach, we shall

* break overall system into a series of versions
* Each version has Analysis, Design, and Implementation
* Have Output from one version to be the input to the next
* Incorporate ideas, issues, lessons learned in one version into the next version

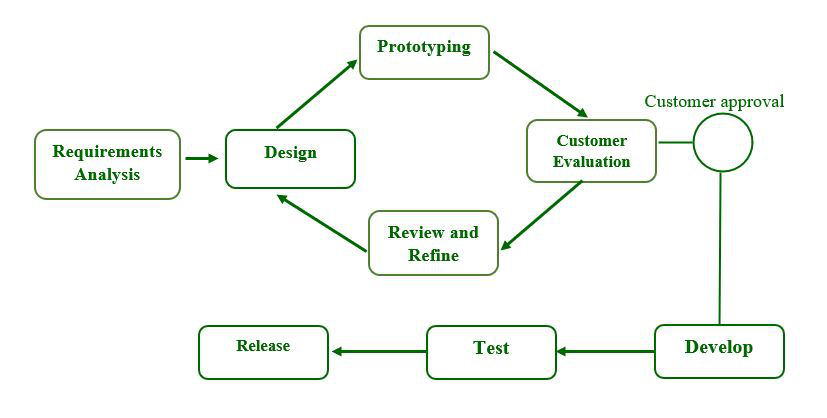
With this approach, the main functions of the system are tested first while the user gets a system to be using as its still in development.

**Prototyping under RAD approach.**

we will use system prototyping for our system

* Analysis, Design, Implementation are performed concurrently
* Start with a "quick-and-dirty" prototype
* Provides minimal functionality
* Repeat the process, refining the prototype each time
* Stop when the prototype is a working system

The diagram below explains how our prototyping approach will be implemented.

[[1]](#footnote-1)

*Figure 5 prototyping structure --* [*https://www.geeksforgeeks.org/software-prototyping-model-and-phases/*](https://www.geeksforgeeks.org/software-prototyping-model-and-phases/)

**SELECTED METHODOLOGY OR PROCESS MODEL;**

**REQUIRED SOFTWARE TOOLS, DESIGN AND DEVELOPMENT TOOLS**

These are sets of technological equipment used to develop a system. This section covers the different tools, both software, and hardware that are required to develop the visually impaired system using voice and image processing technologies.

**Design Tools**

1. Use Case Diagrams
2. Flow charts Diagrams
3. Sketch diagrams

**PROGRAMMING LANGUAGES TO BE USED.**

1. **Frontend and backend**
   1. flutter dart framework
   2. Django with python for ML and AI (for web based)
2. **Database**
   1. Firebase and MySQL
3. **Other technologies**
   1. **Deep learning**
   2. **Machine learning**
   3. **Algorithms**
   4. **Cloud services**

The languages, frameworks, and libraries to be used in the development of the visually impaired system using voice and image processing technologies include but are not limited to the following;

1. **API[[2]](#footnote-2) and ALGORITHM** – these algorithms and API are embedded in mobile phones to do the object detection for us in real-time some of these algorithms include

**Most[[3]](#footnote-3)** important one-stage object detection algorithms

1. **YOLO (2016) (You Only Look Once)**
   1. As a real-time object detection system, YOLO object detection utilizes a single neural network. The latest release of Image AI v2.1.0 now supports training a custom YOLO model to detect any kind and number of objects
2. **SSD (2016) Single-shot detector**
   1. SSD is a popular one-stage detector that can predict multiple classes. The method detects objects in images using a single deep neural network by discretizing the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location.
3. **RetinaNet (2017) Region-based Convolutional Neural Networks**
4. **YOLOv3 (2018)**
5. **YOLOv4 (2020)**
6. **YOLOR (2021)**
   1. YOLOR is a novel object detector introduced in 2021. The algorithm applies implicit and explicit knowledge to the model training at the same time. Herefore, YOLOR can learn a general representation and complete multiple tasks through this general representation

**TARGET HARDWARE/SOFTWARE ENVIRONMENT.**

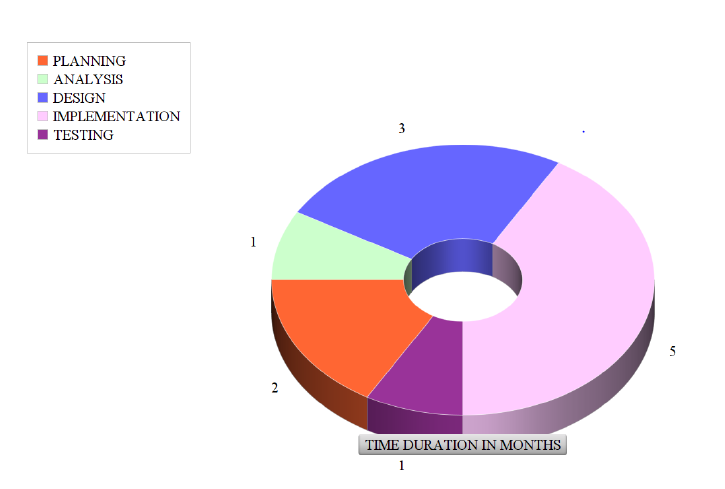
**HARDWARE TOOLS TO USE IN THE DEVELOPMENT PROCESS**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Type of hardware** | **Use** | **Quantity** |
| 1 | Laptop computer | To research, design, develop and  document the system. | 1 |
| 2 | 1TB hard disk | To transfer files | 2 |
| 3 | Printer | To print documents | 1 |
| 4 | 2TB hard disk | For offline backup of the system | 1 |
| 5 | Smart phone | For research and testing the system | 2 |
| 6 | Modem / router | Alternative for internet connection | 1 |

**SOFTWARE TOOLS**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Software’s** | **Use** | **Description** |
| 1 | MS office word | Documentation | MS office 2016 is used for writing documentation starting from proposal up to the end. |
| 2 | Draw.io | Documentation | Visual paradigm was used to draw the diagrams  (context diagram, flow chart, Data flow diagram, entity relationship diagram) |
| 3 | Android studio | Development and  testing | A compiler development and testing of the application |
| 4 | Visual Studio Code | Development | For actual development of the frontend of the web  Application version |
| 5 | Git Bash | Version Control | This application is used for pushing, pulling and  all the other git operations |
| 6 | Java Eclipse | Development and  testing | This application is an integrated Development Environment. |
| 7 | Postman | For testing of REST Applications | This is an application used to make requests to the server and testing the responses. |
| 8 | Viso.ai | Referencing the algorithms | Area for referencing algorithms and API integrations |

**IMPLEMENTATION PLAN**



The graph numbers represent number of months

**FINANCIAL BUDGET- WILL BE USED TO PRODUCE COSTINGS.**

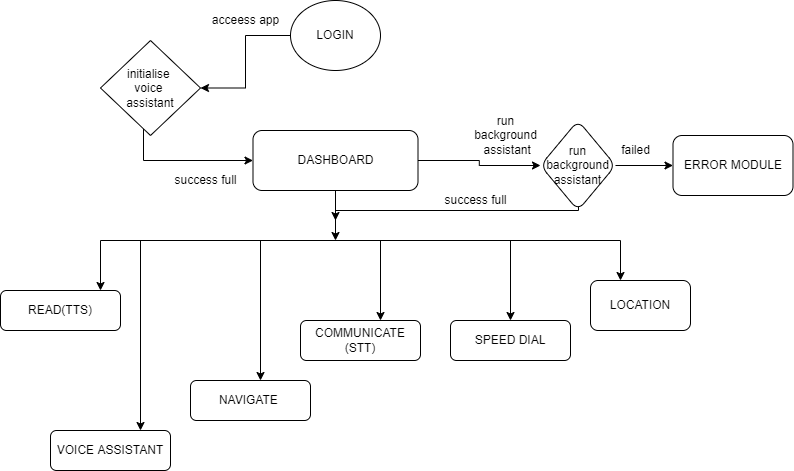
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Type of hardware** | **Use** | **Quantity** | **Unit price** | **total** |
| 1 | Laptop computer | To research, design, develop and document the system. | 2(refurbished) | 4000000 | 8000000 |
| 2 | 1TB hard disk | To transfer files,  For offline backup of the system | 2 | 150,000 | 300000 |
| 3 | Printer | To print documents | 1 | --------- | --------- |
| 4 | Smart phone | For research and testing the system | 2 | 300000 | 600000 |
| 6 | Modem | Alternative for internet connection | 1 | 50000 | 50000 |
| 7 | Algorithm subscriptions | For accesing API keys and webhooks  For the algorithms | 3 | 150000/month | 450000 |
| 8 | Webhook and cloud servies | To access cloud services like voice recognition algorithms online | 1 | 1000000 | 1000000 |

**PROJECT WORKFLOWS AND ACTIVITIES**

These will have an effect on the schedule duration and overall project effort.

**DATA FLOW DIAGRAMS FOR THE APPLICATION**

**APPLICATION DASHBOARD**

****

**EXPLANATION**

Once the user has been logged in to the app they will gain access to the dashboard. In the background, a voice assistant will be initialized to help the user navigate through the app.

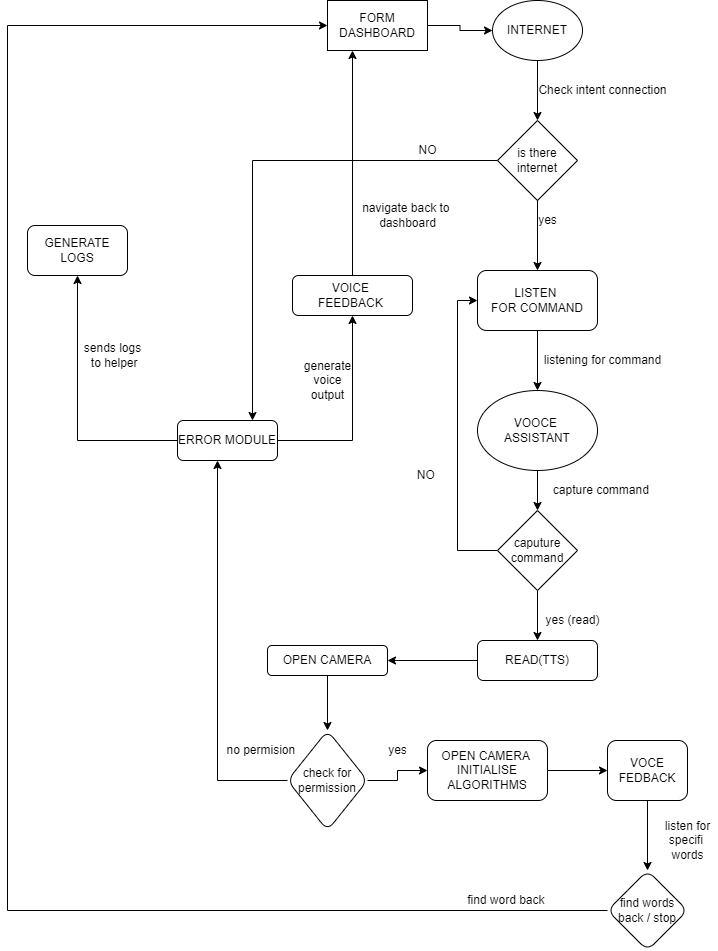
The voice assistant will listen for major or keywords like **start, end, call, read, navigate, location, communicate, and dial** words to perform a corresponding action for the user. Also, in addition, an option for voice assistant will be available to help the user navigate through the app and perform functionalities. In case of any errors during the running of the app, they will be rendered as voice feedback from the error module.

The dashboard has tiles with labels that have corresponding identifiers that will be used during the instruction execution.

**OPERATION**

After the user is logged in, a screen reader and a voice assistant will help the user select desired functionality by either listening to the screen reader or the voice assistant.

**ON SELECTION OF: READ (TEXT TO SPEECH )**

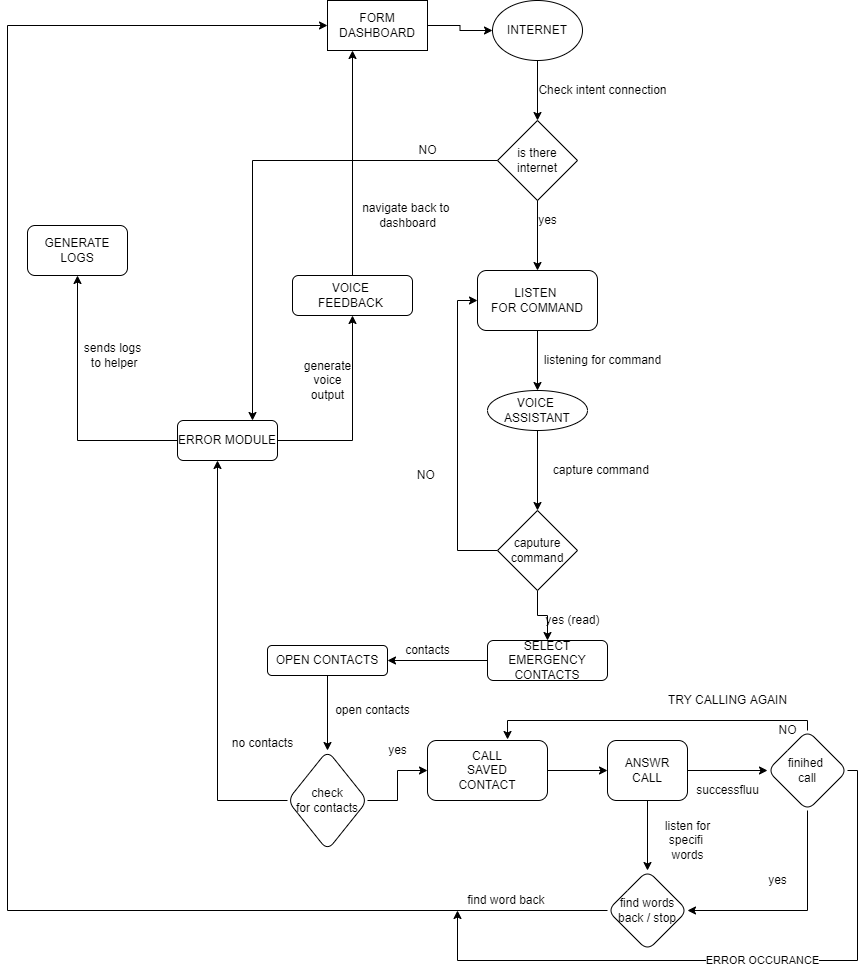
****

**To** select the **READ functionality**, a user will either say read or once u select read, from the dashboard the ap will check if an internet connection is present, if true the voice assistant will keep listening In the background for key works like stop end start according to the stage of instruction execution.

The camera will be opened and the Text To Speech algorithms will be loaded in the background. Once camera permissions are granted to the app the camera will open and load the algorithms to detect text in what the user is trying to view. Text-to-speech (TTS) is a type of assistive technology that reads digital text aloud. It's sometimes called “read-aloud” technology. With a click of a button or the touch of a finger, TTS can take words on a computer or other digital device and convert them into audio.

These words are then rendered as feedback to the user in audio form. While this is happening the voice assistant is listening and waiting for keywords. Once they are detected, the corresponding action will be performed back to the dashboard for a new instruction. In case of any errors, they are tracked back to the error module and rendered as error logs back to the VIP helper for review.

**ON SELECTION OF: EMERGENCY (SPEED DIAL)**

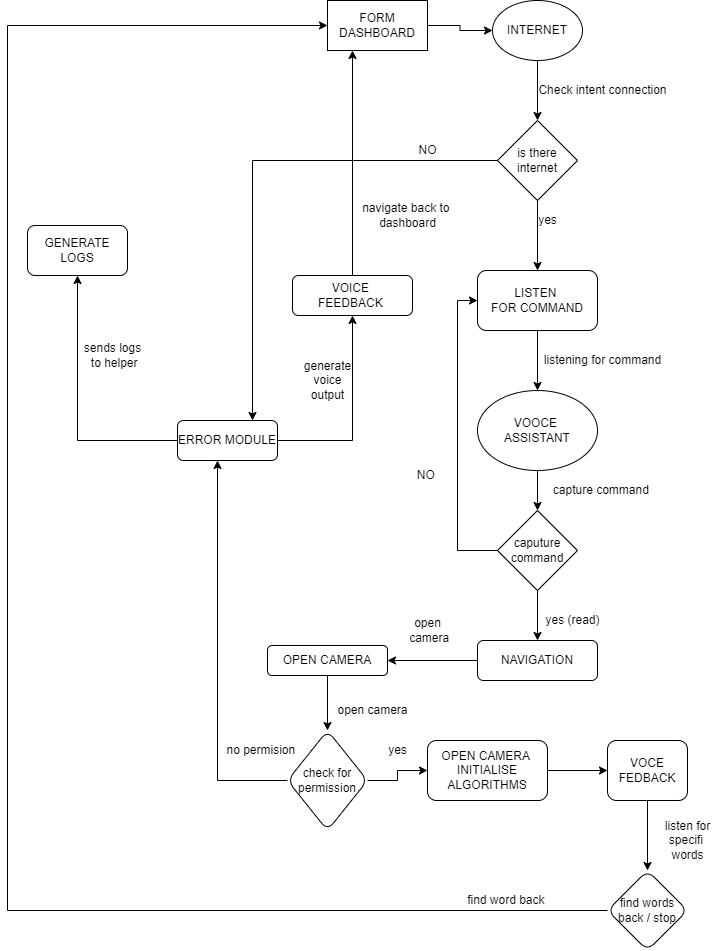
****

**To** select the **EMERGENCY / SPEED DIAL functionality**, a user will either say speed dial, or once u select emergency, from the dashboard the app will check if an internet connection is present, if true the voice assistant will keep listening In the background for key works like stop end start according to the stage of instruction execution.

The app will open the speed dial contact list and read out which contact the user would like to call, once the contact is selected, the cal will begin to ring. In case an error occurs it's tracked and rendered as feedback to the user in audio form. While this is happening the voice assistant is listening and waiting for keywords. Once they are detected, the corresponding action will be performed back to the dashboard for a new instruction they are also tracked back to the error module and rendered as error logs back to the VIP helper for review.

In case of a successful call, the calls will be recorded and later used for other purposes while the voice assistant waits and listens for keywords to perform specific actions on the dashboard

**ON SELECTION OF: NAVIGATION**

****

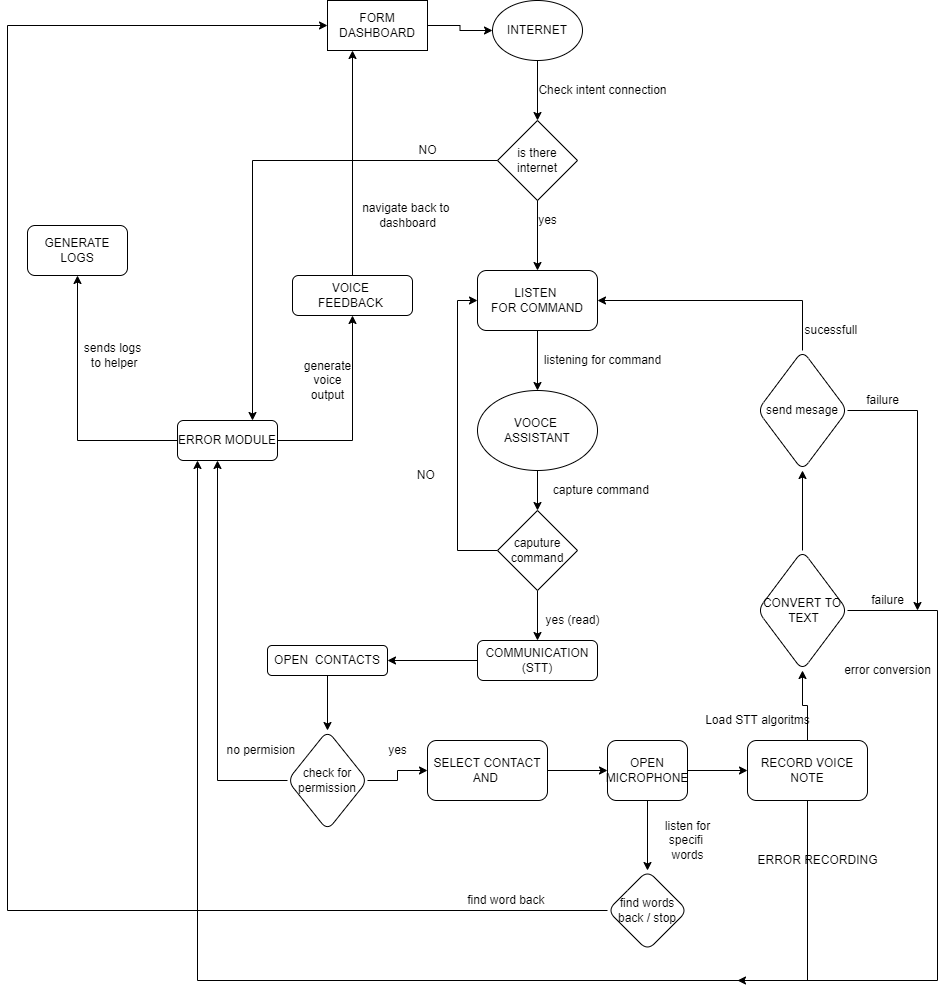
**To** select the **NAVIGATION functionality**, a user will either say navigate or once u select navigate, from the dashboard the app will check if an internet connection is present, if true the voice assistant will keep listening In the background for key works like stop end start according to the stage of instruction execution.

The camera will be opened and the object detection algorithms like CNN, YOLO, and other algorithms will be loaded in the background. Once camera permissions are granted to the app the camera will open and load the algorithms to detect objects and give descriptions according to the surrounding of the user's view.

Object detection is a computer vision technique for locating instances of objects in images or videos. Object detection algorithms typically leverage machine learning or deep learning to produce meaningful results. These results will be rendered as voice feedback to the user. During the object detection, google maps will automatically generate a map view of the user's location and give insights into where the user is and where they would like to go. In case they select a destination location, route suggestions will be given in audio form according to the best route to take and which route has fewer motor vehicles

These words are then rendered as feedback to the user in audio form. While this is happening the voice assistant is listening and waiting for keywords. Once they are detected, the corresponding action will be performed back to the dashboard for a new instruction. In case of any errors, they are tracked back to the error module and rendered as error logs back to the VIP helper for review.

**ON SELECTION OF: COMMUNICATION**

****

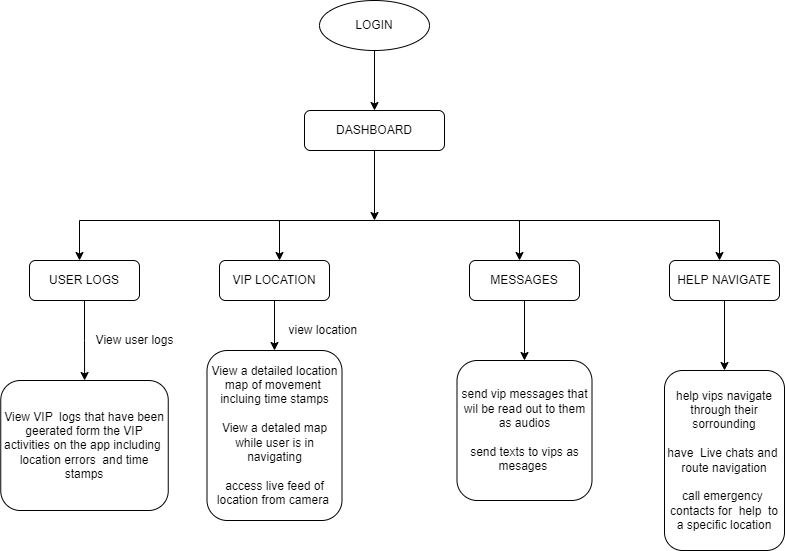
**To** select the **COMMUNICATION functionality**, a user will either say communicate, or once u select communicate, from the dashboard the app will check if an internet connection is present, if true the voice assistant will keep listening In the background for key works like stop end start according to the stage of instruction execution.

The app will open the contact dial contact list and read out which contact the user would like to send a message to, once the contact is selected, the microphone will open and ask for permission to record audio. In case an error occurs it's tracked and rendered as feedback to the user in audio form. While this is happening the voice assistant is listening and waiting for keywords. Once they are detected, the corresponding action will be performed back to the dashboard for a new instruction they are also tracked back to the error module and rendered as error logs back to the VIP helper for review.

In case of a successful message capture, the message will be recorded and later used for other purposes while the voice assistant waits and listens for keywords to perform specific actions on the dashboard. The voice commands are translated into real words that are later processed to the desired destination.

These results will be rendered as text feedback to the desired user in message form. During the process, google maps will automatically generate a map view of the user's location and give insights into where the user is and this data is attached to the message being sent. While this is happening the voice assistant is listening and waiting for keywords. Once they are detected, the corresponding action will be performed back to the dashboard for a new instruction. In case of any errors, they are tracked back to the error module and rendered as error logs back to the VIP helper for review.

**THE HELPER MODULE**

****

**RISKS AND UNCERTAINTIES OF THE PROJECT**

Along development and testing time estimated for around 2yrs.

1. <https://www.geeksforgeeks.org/software-prototyping-model-and-phases/> [↑](#footnote-ref-1)
2. <https://viso.ai/deep-learning/object-detection/> [↑](#footnote-ref-2)
3. <https://viso.ai/deep-learning/object-detection/> [↑](#footnote-ref-3)