Final Report: ENGR 3700 Engineering Practicum (13.5 unit)

By Caleb Ebert (FAN: eber0042, Student No.: 2261560)

Industry Supervisor: Edwin Foo, Academic Supervisor: Nasser Asgari

Start Date: 2/9/2024, Finish Date: 31/1/2025

Report Submitted on the [ADD SUBMITION DATE HERE]

TODO:

Document should be between 30 to 40 pages. Report needs to be within this threshold.

The report should contain the following work:

* Details of the technical project work undertaken by the student during the will placement, should have a strong technical bias.
* All Decisions should be justified and explained.
* Report should clearly indicate the project significance and how the student’s WIL project and work contributed to the objectives of the host organisation/company.

*Yes, your understanding of a strong technical bias aligns with the concept. It means presenting information in a way that assumes the audience possesses a high level of technical expertise or familiarity with the subject matter. This often results in:*

*Detailed Explanations: Extensive elaboration on how things work, the technical mechanisms involved, and the rationale behind design decisions.*

*Use of Jargon: Frequent use of specialized terminology without layman explanations.*

*Focus on Mechanics: Emphasis on the "how" rather than the "what" or "why" from a non-technical perspective.*

*Limited Contextualization: Minimal effort to translate technical concepts into broader or simpler terms for a general audience.*

*This bias can be appropriate for audiences with technical backgrounds, such as engineers or developers. However, for broader audiences, it might require balancing with simpler explanations or summarizing key points at a high level before diving into the specifics.*

# Executive Summary

This final report details the project undertaken at Nanyang Polytechnic (NYP) in Singapore as part of the Engineering Practicum course at Flinders University. The five-month placement focused on developing the capabilities of the Temi robot to enhance student outreach activities. Serving as the sole developer, I acted as the lead for the project, navigating a flexible and open-ended scope that allowed for significant creativity and initiative.

The primary objectives included creating a quiz application, implementing Bluetooth Low Energy (BLE) connectivity, replacing Temi's existing text-to-speech system with a ChatGPT-based plugin, and developing a Software Development Kit (SDK) to enable customizable tours. Comprehensive user documentation was also produced to facilitate replication by future developers. Additional goals added through the project was the develop of an emotion detection system, creating inter-device communication between Temi robots, and expanding the robot's overall functionality.

A significant portion of the placement was devoted to developing two Android applications: the quiz application and the tour SDK. The quiz application, based on a prior project, was rebuilt from scratch. It improving flexibility for adding new questions and introduced a number of new features. The SDK enabled Temi to conduct customizable tours, incorporating advanced features such as proximity detection, directional awareness, motion tracking, speech integration with ChatGPT, inter-device communication, and automated tour interruptions and resets based on user presence.

Key achievements include:

* Successfully developing a quiz application with BLE integration and enhanced customization.
* Designing a comprehensive SDK for the tour system, significantly advancing Temi’s capabilities and improving user engagement.
* Collaborating on an emotion detection system and inter-robot communication to broaden Temi’s utility for interactive outreach.

This placement allowed the development of essential skills in software development, particularly in Android Studio, which was outside my primary focus on electronics and robotics. Valuable experience was gained in managing and organizing code, creating user-friendly systems for developers with minimal technical expertise, and learning how to acquire new technical skills effectively.

Overall, this placement was a transformative experience. This enabled developments that meaningfully contributed to Temi's functionality while enhancing my own capabilities in application development and project management.

# Acknowledgement

I would like to express my sincere gratitude to Edwin Foo, my industry supervisor at Nanyang Polytechnic (NYP), for his valuable input and guidance throughout the project. His willingness to provide feedback and serve as a sounding board for ideas greatly contributed to the development process.

I would also like to extend my thanks to Heng Junxiang for sharing his code that integrated the Temi SDK. His assistance significantly reduced the time and effort required to implement the SDK effectively. Additionally, I am grateful for his collaboration on the Emotion Detection system. While I provided insights and contributed to specific aspects of the system, he conducted the majority of the research and development, demonstrating exceptional expertise and dedication. His involvement will be expressing in \*\*Will do this later

\*\*I need to find the github for which the model I got was collected from.

Table of Contents

[Executive Summary 2](#_Toc188621231)

[Acknowledgement 3](#_Toc188621232)

[List of Tables 5](#_Toc188621233)

[List of Figures 5](#_Toc188621234)

[Introduction 7](#_Toc188621235)

[Temi Tour SDK (5 weeks) 8](#_Toc188621236)

[Introduction 8](#_Toc188621237)

[Planning and Design 9](#_Toc188621238)

[Research on Temi’s mapping system 9](#_Toc188621239)

[Exploring Existing Tools: TemiScript 10](#_Toc188621240)

[Design Process 11](#_Toc188621241)

[Features and Functionality 12](#_Toc188621242)

[Development Process 1](#_Toc188621243)

[Moving Forward with the Temi SDK 1](#_Toc188621244)

[Testing and Validation 3](#_Toc188621245)

[OH2024 and Library Application development (4 weeks) 6](#_Toc188621246)

[Introduction 6](#_Toc188621247)

[Planning and Designing 7](#_Toc188621248)

[Features and functionalities 8](#_Toc188621249)

[Development Process 8](#_Toc188621250)

[Testing and Validation 9](#_Toc188621251)

[User handbook (2 weeks) 10](#_Toc188621252)

[Introduction 10](#_Toc188621253)

[Planning and Designing 11](#_Toc188621254)

[Features and Functionalities 11](#_Toc188621255)

[Testing and Validation 12](#_Toc188621256)

[Transferable Skills Reflection 13](#_Toc188621257)

[Recommendations 1](#_Toc188621258)

[Conclusion 3](#_Toc188621259)

[Appendices 6](#_Toc188621260)

[Flow Diagrams 15](#_Toc188621261)

[OH2024 and Library Application development Flow Diagrams 16](#_Toc188621262)

[Code Archive: 18](#_Toc188621263)

[Code Sample 1 Bluetooth communication between Temi (Client End for V2) 18](#_Toc188621264)

[Code Sample 2 Bluetooth communication between Temi (Server End for V3) 19](#_Toc188621265)

[Code Sample 1 Bluetooth communication between Temi (Bluetooth Manger Class) 23](#_Toc188621266)

[Resource 1: Storyboard for Tour 1](#_Toc188621267)

[Resource 2: Flow Diagram for tour 1](#_Toc188621268)

[Resource 3: Explanation of how the cross Temi commination works 1](#_Toc188621269)

# List of Tables

[Table 1 key milestones for the Tour SDK 8](#_Toc188621270)

[Table 2 features developed for the temi tour SDK 1](#_Toc188621271)

[Table 3 Tool used to develop the Temi Tour SDK 2](#_Toc188621272)

[Table 4: Key Milestones for the OH2024 Application 7](#_Toc188621273)

[Table 5: Key Milestones for the User Handbook 10](#_Toc188621274)

[Table 6 list of all skill groups from Table 7 with reflection on each based on the WIL experience 1](#_Toc188621275)

[Table 7: Metrics used in the Transferable skills survey. 11](#_Toc188621276)

[Table 8: Temi Interaction and Functionality Feature Objectives for OH2024 and Library Application development 12](#_Toc188621277)

[Table 9: Script for video for Main Application 13](#_Toc188621278)

[Table 10 outlined development process for the Temi Tour SDK 1](#_Toc188621279)

# List of Figures

[Figure 1 pictures of the Temi Robots from Nanyang Polytechnic (NYP). Left image is version 2 and the right is version 3. 7](#_Toc188621280)

[Figure 2 example of the Temi Centre UI for altering saved LIDAR maps. The red circle highlights an example of a location point that can be placed on the map. 9](#_Toc188621281)

[Figure 3: A visual representation of the View-ViewModel-Model (VVM) architecture, illustrating the distinct roles of each component in managing the user interface (UI), logic, data flow, and core functionality (vvmv). 1](#_Toc188621282)

[Figure 4 transferable skills assessment radar graph 13](#_Toc188621283)

[Figure 5: image of LIDAR present on the Temi robots. Same model is used across version 2 and 3. 6](#_Toc188621284)

[Figure 6: exert from media developed for the Temi Tour SDK to illustrate how to talk to temi. 7](#_Toc188621285)

[Figure 7: Home screen for the Main application. 7](#_Toc188621286)

[Figure 8: Example of the map system. The pointer indicates the Temi’s location and the red dot the location desired to go to. 7](#_Toc188621287)

[Figure 9: Home screen for the Intro application. 8](#_Toc188621288)

[Figure 10: Home screen for the Library application. 8](#_Toc188621289)

[Figure 11: Story board for the Intro Application. The story boards are read from left to right and top to bottom. 9](#_Toc188621290)

[Figure 12: Comparison of slide 3 in the story board from Figure 11 to the developed scene for the video in Intro Application. 10](#_Toc188621291)

# List of Flow Diagrams

[Flow Diagram 1: Main Application - Greeting Module (Greet Mode) Flow Diagram 16](#_Toc188621790)

[Flow Diagram 2: Main Application - Button Prompt System (Button Display Prompt) Flow Diagram 17](#_Toc188621791)

[Flow Diagram 3: Main Application - Q&A Button (Q&A) Flow Diagram (Button 1) 17](#_Toc188621792)

[Flow Diagram 4: Main Application - Directions/Locations Button (Directions/Locations) Flow Diagram (Button 2) 18](#_Toc188621793)

What stuff did I get done?

* Created an application for a quiz that allows mild adaption of the quiz application
* Created a method for using Bluetooth
* Created a user handbook (still needs to be worked on as it is not completed)
* Created an emotional detection system with collaboration with someone else
* Created a template for setting up the system for use in the SDK
* Created a greet mode for… (I forgot their name I think it was Kajima something) <- I have a video of this
* Created a tour application
* Created a method to allow the temi robots to communicate using Bluetooth
* Created a method to use ChatGPT
* Created a method to use ChatGPT to communicate via Bluetooth
* Created an application for OH2024
* Created a system for the library to use

# Introduction



Figure 1 pictures of the Temi Robots from Nanyang Polytechnic (NYP). Left image is version 2 and the right is version 3.

This report summarizes the enhancements made to the Temi robot during a placement at Nanyang Polytechnic (NYP) in Singapore. As part of the Engineering Practicum course at Flinders University, the project aimed to broaden the functionalities of Temi to support NYP's student outreach efforts. NYP utilized two devices: Temi V2 and Temi V3 (see Figure 1 for images).

Temi is an AI-powered personal robot designed to combine mobility, communication, and interaction capabilities. With autonomous navigation, speech recognition, and a high-resolution touchscreen, Temi enhances user engagement across various contexts. These include personal assistance to corporate and educational applications. In this project, Temi was positioned to improve NYP’s outreach by engaging with potential students, parents, and stakeholders. Its autonomous navigation and conversational abilities made it a valuable tool for communication and event navigation.

The project's primary goals were to develop applications that would make Temi more engaging and versatile for outreach scenarios, including:

* Creating a quiz application that interacts with a Bluetooth-enabled chocolate dispenser to distribute rewards based on quiz results.
* Replacing Temi’s default text-to-speech system with an advanced ChatGPT-based plugin for enhanced communication.
* Developing a Software Development Kit (SDK) to enable customizable guided tours using Temi’s capabilities.
* Producing comprehensive user documentation to assist future developers in replicating and building upon the work.

**The significance of this project lies in its ability to demonstrate NYP’s technological innovation while providing a unique and interactive experience for potential students and visitors. By enhancing Temi’s functionality, the project positions the robot as a key tool in showcasing NYP’s commitment to cutting-edge technology and student engagement.**

Throughout the course of the project, some adjustments were made to the original objectives. While the primary goals remained, additional functionalities were developed, including an emotion detection system in collaboration with another intern, inter-robot communication capabilities, and expanded interactive features. These changes addressed broader use cases and ensured the robot could meet dynamic outreach needs.

This report focuses on detailing the project work completed during the placement. It assumes that the reader has the necessary technical background to fully comprehend the functionalities and implementations of the systems developed. The order in which each project goal is presented corresponds to the amount of time dedicated to its development. Therefore, the report will first cover the Software Development Kit (SDK) combined with the ChatGPT text-to-speech system, followed by the quiz application, and finally, the creation of user documentation.

# Temi Tour SDK (5 weeks)

## Introduction

The Temi SDK was developed to overcome the limitations of existing solutions like TemiScript, offering a more advanced and flexible system for creating guided tours. The primary purpose of the SDK is to transform Temi into a dynamic and responsive guide capable of adjusting to its environment and user inputs. Through the addition of innovative features such as inter-Temi communication, user tracking, and misuse detection, the SDK equips the robot with tools to deliver tailored experiences that go beyond the static, scripted behaviours of previous systems. This development plays a pivotal role in achieving project goals, as it allows for seamless tour customization, improved engagement, and the ability to showcase interactive technology in a way that captivates audiences. The SDK lays a strong foundation for future enhancements, ensuring scalability and versatility for various applications. All key millstones for the deliverable have been outlined in Table 1. Priority has been added to indicate the importance to the overall importance of features and which features should be focused on first.

Table 1 key milestones for the Tour SDK

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Purpose | Alignment with Objectives | \*\*Importance |
| T1: Location Navigation | Enables the Temi robot to navigate to specific locations. | Necessary to be able to get to tour locations. | Needed |
| T2: Voice Interaction | Allows Temi to speak and interact verbally with users. | Necessary to explain information to users. | Needed |
| T3: Simultaneous Movement and Speaking | Enables Temi to move while speaking. | Improves tour flow and reduces idle time when Temi is going to a new location. | High |
| T4: Directional Movement Control | Allows Temi to switch between moving backwards and forwards. | Allows temi to be facing the tour members when going to locations. | Medium |
| T5: User Detection | Enables Temi to detect users’ presence and movements. | Enhances interaction by responding to user proximity. | High |
| T6: Voice Recognition and Response | Allows Temi to listen to and respond to user commands. | Facilitates personalized and interactive tours. | Medium |
| T7: ChatGPT Integration | Enables Temi to use ChatGPT for advanced conversational capabilities. | Enhances depth and relevance of interactions. | Low |
| T8: Volume Control | Allows users to adjust Temi’s volume. | Improves user experience with customizable audio levels. | Low |
| T9: Question Answering | Enables Temi to answer user questions. | Enhances educational aspect of tours. | Low |
| T10: Image Display | Allows Temi to show images during tours. | Provides visual aids to enrich tour content. | Medium |
| T11: Music Playback | Enables Temi to play music. | Adds an auditory dimension to tours. | Low |

\*\* **Importance Scale:** Needed, High, Medium, Low

## Planning and Design

### Research on Temi’s mapping system

The development of the Temi SDK has primarily focused on enhancing the tools available for creating tours. While there was an existing system in place, it had significant limitations in terms of customization and flexibility, which hindered its effectiveness. The primary goal of the SDK was to improve upon these tools and allow for more versatile and efficient tour creation.

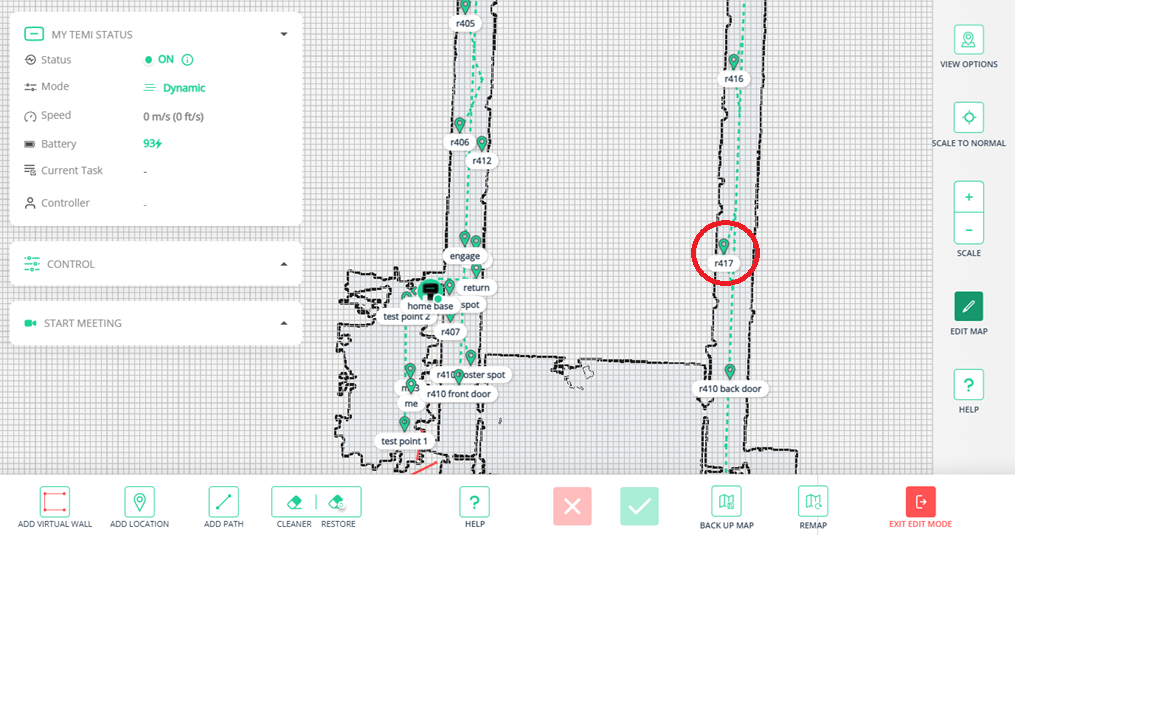


Figure 2 example of the Temi Centre UI for altering saved LIDAR maps. The red circle highlights an example of a location point that can be placed on the map.

Before developing the SDK, it was crucial to understand the mapping process used by the Temi robot. In order to deploy a Temi into a location, the robot first needs to create a map of the environment. This process is initiated by placing the Temi into a "mapping mode," during which it moves through the environment and collects data using a LIDAR module located at the bottom of the robot (see Figure 5 in the appendices for image of the LIDAR). Figure 2 is an example of a generated map. These maps are then saved and can be accessed via the Temi Center platform, where further modifications can be made.

The core feature of the mapping tools is the ability to add locations to the map, which can be seen in Figure 2. When adding a location, three key pieces of data are required: the location’s name, the direction the Temi should face upon arrival, and the tilt angle for the Temi’s screen. These locations are essential for the robot’s navigation, as it uses them as waypoints to move around the environment. However, there are limitations. For example, if an obstacle blocks the Temi’s ability to get to a location, the robot will be unable to adapt to this. Hence, requiring intervention to resolve the issue. This issue becomes more problematic when the robot is expected to operate autonomously, as opposed to being monitored by an operator.

To address navigation challenges, virtual walls and paths can be defined within the mapping system. The LIDAR system has difficulty detecting transparent objects like glass walls, which can lead to navigation issues. Virtual walls create barriers that the robot cannot cross, helping to restrict access to certain areas. The path tool, on the other hand, specifies a preferred route for the robot to follow. Though it is not always strictly enforced. If an obstacle obstructs the path, the Temi may deviate from the defined route. Another critical aspect is that the Temi’s mapping system does not allow it to operate outside of mapped areas; if the robot strays from the designated map, it will be considered lost and will need to be recovered.

### Exploring Existing Tools: TemiScript

Prior to the development of the SDK, TemiScript was used as an alternative for creating tours. Developed by an independent third party (Robo Solutions), TemiScript allowed for basic control over the Temi robot. This included movement, basic interaction with the environment, and multimedia features. However, TemiScript had significant limitations, particularly in its ability to provide in-depth customization and interactivity for more complex tour scenarios. The lack of features such as variables and conditional statements made it difficult to create dynamic, interactive tours. As a result, tours created with TemiScript were typically linear and lacked the flexibility needed for more advanced use cases. A basic tour was created using TemiScript, showcasing the robot’s ability to visit predefined locations, speak, display images, and play audio. Although these features were sufficient for a simple tour, several limitations of TemiScript became apparent.

One major limitation of TemiScript was its inflexibility when developing tours for multiple Temi robots. \*\*3 TemiScript relies on an application that needs to be installed on each device used for tour creation. There are several methods for developing a tour, including a UI interface available on a dedicated website or the device’s application. However, the system uses a cell-based approach where functions are added in sequence, one cell at a time. This structure makes it difficult to adjust or add new steps in the middle of a sequence. \*\*3 While deleting cells is possible, adding a new one in the middle requires removing all subsequent cells and recreating the sequence. This process can be time-consuming and inefficient, especially when planning more lengthy tours. The alternative, using a text file to upload the script in a text file format circumvented this limitation.

Additionally, transferring a script from one device to another posed challenges. TemiScript allowed for customizing buttons to trigger specific scripts, but issues arose with the layout of buttons and their associated scripts. This made it cumbersome to move a completed sequence from one device to another, particularly when developing tours for multiple robots.

**\*\*3 should get an image of temioscript to help explain the issue that I had.**

### Design Process

The development of the SDK began with the decision to create a tour as a core use case. This approach ensured that all potential features required for designing and executing interactive tours could be implemented and validated. By focusing on a fully realized tour as the test case, the SDK could be tailored to meet the project's goals, serving as both a functional medium for testing and a robust platform for validating its capabilities. Conducting trial runs of the tour with target users provided valuable insights, allowing for iterative adjustments to make the system both functional and appealing to its intended audience. Additionally, this process ensured that the SDK’s methods were intuitive and easy to understand.

A sequential approach was used to implement features. Each feature was designed, integrated into the SDK, and then validated to confirm its intended functionality. This modular development method simplified the identification and resolution of bugs, as each system component could be tested in isolation. A key focus during the design process was to ensure that creating tours using the SDK was straightforward. To achieve this, functions were designed to handle the primary workload, requiring only simple inputs to activate and manage the system’s core tasks.

To support the SDK’s development, two key resources were created: a storyboard and a flow diagram. Both resources have been provided in the appendices as resource 1 and resource 2 respectively.

* The storyboard helped visualize the tour and identify desired features. It provided a rough framework of how the tour would look and allowed for a structured breakdown into smaller, manageable sections. By segmenting the tour into states, it was possible to focus on developing and debugging each part individually. Each storyboard panel corresponded to a specific state within the SDK, ensuring a logical flow and facilitating modular development. This approach also made it easier to replace or refine sections of the tour as needed. For example, the initial plan was to have the Temi start at the home bay, which was replaced with the current start in the story board.
* The flow diagram, which will be discussed in detail later, was used to map out the logical structure and interactions within the tour. This ensured that the SDK's logic flow was consistent, coherent, and adaptable to different scenarios.

By combining these tools, the development process became more efficient, enabling precise feature implementation and testing. This methodology not only improved the robustness of the SDK but also ensured that it met the project’s objectives of usability, functionality, and flexibility.

**\*\*Want to create a map and use that to explain the planning and design.**

**\*\*I have started created the flow diagram but still need to do more work in it.**

## Features and Functionality

The Temi SDK enhances the existing tools, such as TemiScript, by offering a more flexible and robust solution for creating tours. While TemiScript was suitable for basic tours, it lacked the advanced features required for more complex, interactive, and adaptable behaviours. The SDK enables developers to create tours that are not only customizable but also interactive, making it easier to deploy the same tours across multiple Temi robots.

As outlined in Table 1 of the "Temi Tour SDK - Introduction," the key milestones were used as a foundation for the features listed in Table 2. These features enhance tours by adding dynamic and responsive behaviour. This is achieved through features like position and motion tracking, text-to-speech with basic text analysis for understanding user responses, and the integration of ChatGPT for interactive question-and-answer segments. These improvements lead to more engaging tours, increasing student interest and enhancing the outreach efforts of institutions like Nanyang Polytechnic.

One of the primary benefits of the SDK is its flexibility. Since the SDK is developed in Android Studio, it allows other developers to easily set up and modify the code. The tours are written in code, which provides greater control and the ability to add new features that were not originally planned. While the SDK supports feature expansion, it also includes pre-built functions to simplify the process for developers. For example, it offers functions for managing text-to-speech, controlling movement and speaking, and implementing systems for tracking user interactions and resetting the system if necessary. This makes it easier for other developers to build on the existing features and add new functionalities.

A notable innovation is the "interrupt system." This system monitors the presence of users and allows the tour to stop if users are not detected for an adjustable amount of time, if the Temi is being misused (e.g., being picked up or manually moved), or if a user closes. The interrupt also works well with a reset system, which ensures that the system resets if users leave the tour midway.

The main highlight of the SDK is the communication system between two Temi robots. The official Temi SDK does not support communication between Temi devices, so a new system was developed from scratch. In this system, one Temi acts as the "server" and the other as the "client." The client uses Bluetooth to detect active devices and connects to the server using a unique identifier called a UUID. UUIDs function as identifiers for communication ports, specifying the type of data transfer and interaction that can occur. A custom UUID was used for communication purposes.

For the communication to take place, the server creates a "server socket" to exchange information using the custom UUID channel. The server is programmed to allow connections only at specific parts of the tour, while the client continuously searches for the connection to the server.

The communication between the two Temis is managed using "flags" to indicate when one device has finished speaking. For example, when one Temi finishes a line of dialogue, it sends a flag with the message "END" to the other Temi. If no data is passed through the communication channel for a set period, the Bluetooth connection can enter a sleep state, causing noticeable pauses in the dialogue exchange. To ensure that the channel stays active, an "IDLE" flag is sent periodically, ensuring the conversation remains fluid without interruptions.

There are two types of conversations programmed: a scripted conversation and a semi-dynamic one using ChatGPT to generate responses. The scripted conversation uses the flag system mentioned earlier. The second type of conversation is triggered when a user asks a question at the end of a tour segment. In this case, the server sends a "GPT" flag to the other device, signalling it to switch to using ChatGPT for the conversation.

During the tours, both version two and version three Temis were used. The version two Temi is unable to use the ChatGPT API. To address this, the server (the version 3 Temi) generates the conversation and stores it as variables. This allows the version 3 Temi to pass the dialogue through the Bluetooth channel, enabling the version 2 Temi to speak the lines.

By incorporating ChatGPT, the SDK enables dynamic, real-time conversations between the two Temis. This creates a more interactive and engaging experience, while still maintaining the structure of the planned tour. For information on how the cross temi communication work, please see resource 3 and code samples 1 to 3 in the appendices for more information.

Table 2 features developed for the temi tour SDK

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Description | Details | Alignment with Objectives |
| Music Player | Allows controlling of audio played by the Temi SDK. | * Start and stop music that plays in an indefinite loop. * Adjust music volume independently from the Temi's system volume. * Configure the left and right speaker volume (not utilized in the current application). * Release the audio player resource when playback is complete to save system resources. * Play a track once and release the resource, suitable for sound effects. | Allows adding music and sound effects to make the tour more customisable and engaging |
| GIF Player | Play GIFs on screen. | * Play a GIF on the screen, such as an idle animation when the device is not in use. * Change the displayed resource to a different GIF or static image as needed. * Play a sound effect when the screen is pressed while a specific GIF is displayed (e.g., on V2 Temi, pressing a dog GIF triggers a bark sound). | Makes the tour more engaging using visual elements such as images and GIFs to portray information. |
| Volume Control | Programmatically control Temi's volume. | * Adjust the Temi's system volume programmatically. | Provides method to alter the volume of the temi to adjust based on the environment it is in. |
| Misuse Detection | Check for misuse state of Temi. | * Detect if the Temi is being picked up or dragged. (Not utilized in the tour application.) | Ensures that when the robot it not handled correctly there is a method in place to create a response. |
| Constraint Follow | Enhanced user tracking and orientation. | * Operates only when the Temi is not in a misuse state. * Set a default yaw angle for idle mode when no users are detected. * Track a user within a configurable angle range. * Handle overlap when the angle boundary spans between positive and negative π. * Turn in the direction where the user was last seen if the user is missing. * Track the user’s position and movement. | Provides a system for Temi to idle on when not in tours. The idea is to provide a method to try to get users to engage with the Temi. |
| Position and Motion Detection | Track user position and movement. | * Track user position in the y-direction (far, mid-range, close, missing). * Track user position in the x-direction (left, right, or straight ahead, missing). * Detect user motion in both x and y directions to inform directional adjustments in Constraint Follow. | Enhances user interaction by accurately tracking movements. |
| System Interrupt | Interrupt system operations based on conditions. | * Interrupts Temi's movement or speech based on specific conditions, such as:   + User missing.   + Misuse case triggered (e.g., Temi being picked up or moved).   + User standing too close. * Triggers can be assigned per action, providing precise control over what actions can be interrupted. * Retains the interrupted action, allowing the system to resume once the interrupt condition is resolved. * Includes a missing person interrupt system:   + Temi prompts the user to return for a set number of times, with adjustable timing between prompts.   + If the threshold is exceeded, the tour is cancelled, and Temi returns to the starting position. * Enhances tour reliability by automatically resetting in cases where users abandon the tour, offering an improvement over previous methods. | Ensures smooth operation and resets tours if users leave midway. |
| Tour Creation | Organize and control tour states. | * Organizes the tour into states, each represented as a case in a switch-case structure, with a separate thread managing the state flow. * Enables easy modification, testing, and addition of new states to the tour. | Simplifies tour customization and improves development efficiency. |
| Conversation Management | Control conversations with users. | * Stops ongoing AI-driven conversations after receiving a user response, preventing the Temi AI model from overriding the response handling. | Ensures that conversation systems work as intended and is not interrupted by built in system in Temi. |
| Behavior Control Functions | Listen, ask questions, control movements. | * **Listen**: Captures user responses. * **Ask Question**: Prompts the user with a question and generates an AI-driven response. Displays a "thinking" GIF and plays music with a randomized delay (7–15 seconds) to simulate thinking time. * **Goto**: Directs Temi to a specified location on its LIDAR map with configurable settings (e.g., speech, facial expressions, movement direction, interrupt handling). Automatically retries aborted movements caused by obstacles. * **Speak**: Allows Temi to speak text, managing interruptions by breaking text into sentences and repeating interrupted lines. * **Tilt Angle**: Adjusts Temi's head tilt angle. * **Stop Movement**: Halts all Temi movements. * **Move Direction**: Commands Temi to move in a specific direction. * **Control Speed**: Adjusts Temi's movement speed. * **Extract User Name**: Processes user input to identify names. * **Text Processing**: Interprets user responses for positive or negative confirmations. * **Yaw Control**: Adjusts Temi’s yaw (rotational) angle. | Provides flexible and dynamic control over Temi’s behaviours. |
| Bluetooth System | Manage Bluetooth connections. | * Scans for Bluetooth Classic connections and connects with other Temi devices. * Sends flags between connected Temis to synchronize actions (e.g., knowing when the other has finished speaking). * Integrates ChatGPT for dynamic dialogue, allowing flexible conversations while adhering to scripted paths. | Enhances communication and coordination between multiple Temis. |

## Development Process

### Moving Forward with the Temi SDK

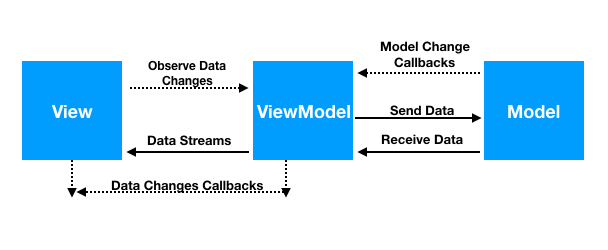


Figure 3: A visual representation of the View-ViewModel-Model (VVM) architecture, illustrating the distinct roles of each component in managing the user interface (UI), logic, data flow, and core functionality (vvmv).

Before developing the Tour SDK, it was essential to first understand how to effectively use and set up the SDK. To organize the code, a view-model architecture was adopted, which provides a clear separation of concerns. In this structure, the view is responsible for managing the user interface (UI), the view model handles the logic and data flow, and the model contains the core functionality. An illustration of this can be seen in Figure 3.

A key component of the system is the Robotemi SDK from the Temi Center, which was integrated using the RobotController class. This class offers access to the essential functions and listeners necessary to operate the Temi robot. By structuring the code this way, only the relevant functions are handled, simplifying the logic and minimizing complexity. The use of listeners allows the system to respond to specific events, such as robot movement or user detection, without unnecessary processing.

The RobotController follows a singleton pattern, ensuring that only one instance of the class exists, which helps prevent multiple listeners from causing redundant updates. Additionally, mutable state flows were utilized to manage real-time data updates, eliminating the need for constant polling loops and improving efficiency.

This framework provided the necessary foundation for the development of the Tour SDK. It was used for all subsequent developed deliverables, except the Quiz Application. The development process focused on breaking down the tour application into smaller sub-sections. Each of these sub-sections was developed, tested, and validated as they were integrated into the main application. The initial features added to the application focused on core functionalities, such as user detection, navigation to different locations, and speech capabilities. By developing these basic features first, more complex functionalities could later be built upon them. This approach not only minimized debugging but also accelerated development, as existing features could be reused and adjusted when necessary, rather than creating entirely new systems.

For example, core features like speech recognition, keyword processing (e.g., detecting “yes” or “no”), and speaking were developed first. These were followed by a ChatGPT-based response system using the ChatGPT API for generating replies to hardcoded queries. Subsequently, a query system was created to enable Temi to ask questions and process user responses. Combining these systems resulted in a functional Q&A system.

To highlight the timeline of development, a table has been created and stored Table 3 (Appendices). This table includes two main elements: a list of completed objectives with their respective completion dates and the specific points in time when milestones were achieved. It is important to note that the milestones are based on Table 1, with T2, T8, T10, and T11 excluded. The order of deliverables in the table reflects the time spent on each task, rather than the chronological order of completion. Prior to finalizing the Temi Tour SDK, the quiz application and a greeting application for Robosolutions were developed. These projects utilized the Temi robot, and many of the core features, such as voice interaction, were reused across deliverables. Consequently, several milestones were achieved during earlier deliverables. Hence, all milestones for this deliverable were developed and no changes to them were done.

Table 3 outlines the tools used in the development of the Temi Tour SDK. Required tools include L1, L2, L3, L4, and L8. These what are needed to be able to create the application and upload it on to Temi. The remaining tools supported aesthetic aspects of the tour and could be substituted with similar software. Familiarity with these tools (e.g., Microsoft PowerPoint) or their cost-effectiveness influenced their selection.

Table 3: Tool used to develop the Temi Tour SDK

|  |  |  |
| --- | --- | --- |
| Category | Tool/Resource | Purpose |
| Hardware | L1: Temi V2 and V3 | Primary devices for testing and deploying the SDK. |
|  | L2: Computer with a Local Wi-Fi Network | Facilitated a stable connection between Temi devices and the development environment. |
| Software | L3: Platforming Tool | Connected Temi devices to the computer for development and debugging. |
|  | L4: Temi Center Website | Used for creating and editing maps for navigation. |
|  | L5: Adobe Firefly | AI-based tool for generating images to enhance the visual aspects of the tour. |
|  | L6: Microsoft PowerPoint | Designed sprites and animations for the application. |
|  | L7: GIF Creation Tools (e.g., GIFHY) | Generated and managed GIFs for animations within the application. |
| API / Online Services | L8: ChatGPT API Key | Integrated AI-driven functionality, enhancing user interaction capabilities. |

## Testing and Validation

As briefly mentioned in the “Development Process” section, each feature was tested as it was developed to ensure the robustness of the application. The primary focus of testing was to verify that each feature functioned as intended and did not cause the application to crash. Android Studio offers tools for running and testing applications on virtual machines. However, these tools are designed primarily for testing applications on phones and tablets, which limited their utility for testing the application on Temi robots. As a result, most testing was done by directly uploading test versions of the application onto the Temi devices.

It is important to note that the operating system provided by Temi Center (the creators of the Temi robots) and the hardware capabilities of the Temi version 2 and version 3 robots differ. Consequently, uploading a test application to a version 2 Temi robot takes between 1 to 2 minutes, while on a version 3, it takes only 1 to 5 seconds. The differences in operating systems also led to compatibility issues between the applications developed for version 2 and version 3. While this was not a significant issue in the development of the Temi Tour SDK, it did present challenges for other deliverables, which will be discussed later.

During the development of features, each element was tested as it was created, whenever possible. However, not all features, unlike the audio player, provide clear or transparent results during testing. For instance, in the initial development of the user detection system, the detection results are stored in a global state flow variable, which is assembled by other functions within the application. However, this does not produce a visible cue during testing.

To address this, Android Studio provides a feature called Logcat, which displays logs generated by the device while it is running. This feature can be used in the application with a function called Log, which allows allocating a tag to help search for specific logs in the Logcat and a data entry in the form of a string. This capability proved to be extremely helpful for testing and debugging, as it provided access to data at specific points within the application and enabled tracking of the application’s logical state.

The tags were particularly useful for isolating logs related to a specific feature being developed. Samples of these logs can be found in the code examples in the appendices.

While this process was effective for the development of simpler features, larger and more complex features were not as straightforward. The main challenge arose from features with more intricate logic systems or those that depended on multiple other features. Since there was no systematic method to test every possible state a feature could be in when integrated into the main code, there were occasional instances where certain features did not function as intended or caused crashes which were not identified during initial testing. However, it was rare for this process to overlook bugs that resulted in major issues such as crashes. In most cases, the features in the application displayed unusual behaviors. While these oddities were not ideal, they did not render the overall application unusable; they simply made it more inconvenient.

To address this, the application underwent validation once it reached a relatively stable version with sufficient improvements. During this phase, the application was tested in as many real-world scenarios as possible to ensure its functionality. Any bugs observed were recorded, along with the conditions that led to them, and were subsequently patched. Once the issues were resolved, the next set of features was implemented, and the testing cycle repeated.

Due to the nature of the project, the completion of all features and milestones did not necessarily indicate that the deliverable was finished. Since the project's primary goal was student outreach, there was a significant subjective component to its success. As a result, the final version underwent trial runs with the intended user group. A total of five trials were conducted using the developed tour with the Temi Tour SDK: one with the academic supervisor of the project, two with staff at NYP, and two with groups of students visiting NYP for a tour. The main objective of these trials was to observe how users interacted with and responded to the tour, as well as to gather their feedback.

The feedback and notes from the trials were as follows. Some recommendations were made to change certain lines of dialogue in the tour, as they were found to be unclear or confusing in their intent.

Users participating in the tour were pleasantly surprised and delighted when engaging with the interactive segments, particularly the communication between the Temi robots. Many users actively recorded these encounters on their phones, which was a positive indication of engagement. However, some users experienced difficulty engaging with the text-to-speech segment. This issue appeared to stem from the fact that many users were unfamiliar with robotic devices such as Temi. To address this, it was decided that a media piece should be developed to better illustrate how to interact with Temi. Given the lack of a budget for licensed software, Microsoft PowerPoint was chosen for this task, as it was already available on the project’s allocated PC, and the developer had prior experience creating animations with it. Figure 6 (Appendices) provides an excerpt from the animation created. The implementation of this media improved interaction with the speech section of Temi, although there were still minor issues with the timing of when users were meant to speak. Additionally, there was a noted hesitation among users when interacting with Temi, which often led to the Q&A section of the tour not being utilized.

During one of the tours, it was observed that the tour stopped midway. Users inquired why the Temi robot did not stop when they moved away. In response, a new system was developed for the SDK to detect when users were not in front of the Temi. If no user was detected, the tour would be paused until the user returned. If the user did not return, the system would reset the tour.

In one student tour, the majority of participants were from China and spoke only Mandarin. This highlighted an issue with the SDK’s lack of support for languages other than English. As a result, a new method was implemented into the SDK to allow easier editing and support for additional languages.

For the overall Temi Tour SDK to be considered a success, it needed to fulfil the following criteria: expand upon features provided by other products, enable further expansion of features, actively engage students during the tour, and function without issues.

In terms of the first two points, the deliverable successfully achieved all outlined milestones, and its development format allows for straightforward expansion. The source code and all necessary resources (e.g., PNG and MP3 files) have been uploaded to a publicly available GitHub repository. However, one identified limitation is the learning curve associated with using the Temi Tour SDK. Utilizing the SDK requires a solid understanding of Android Studio, proficiency in Kotlin, and familiarity with the Temi Center library. This trade-off between flexibility in development and ease of use was deemed necessary, as these two aspects are often at odds with each other.

Trial runs demonstrated that the tour developed with the SDK was engaging for students. While some users experienced initial difficulty understanding how to interact with the Temi robots, this was largely attributed to their unfamiliarity with such devices in everyday life.

The overall application performed reliably, with nearly all features functioning as intended. The only notable issue involved the communication system between the two Temi devices, which occasionally failed to establish a proper connection. When the application is first launched for the tour, some troubleshooting may be required to ensure the devices can connect. Nevertheless, the tour includes a contingency method that allows it to bypass the Temi communication section if a connection cannot be established. While there are still improvements that can be made, the derivable is to a high enough stand to be considered a success.

# OH2024 and Library Application development (4 weeks)

## Introduction

The OH2024 and Library Application Development involved the creation of three distinct applications built upon the same foundational framework. Two of these applications were designed specifically for use at Nanyang Polytechnic's (NYP) Open House 2025 (OH2024), while the third was tailored for deployment within the library at NYP. Despite their similarities in structure, each application had unique functionalities and purposes. An open house is an event hosted by Nanyang Polytechnic (NYP) to provide prospective students, parents, and other interested parties with the opportunity to explore the campus facilities, interact with staff and students, and learn more about the courses and programs offered. The first application, referred to as Main, consisted of three sections, with its home screen shown in Figure 7 (Appendices):

1. **Q&A Section**: This section allowed users to choose from a range of pre-answered questions. Additionally, users could verbally ask a question, and the application would use ChatGPT to generate an answer. Each selected question included the option to display an accompanying image, and subtitles were automatically generated along with the response.
2. **Mapping System**: This feature enabled the Temi robot to navigate to four predefined destinations. Upon selecting this section, four buttons were displayed, each corresponding to a destination. When a button was pressed, a map of the relevant area appeared, highlighting two key elements with an example shown in Figure 8 (Appendices):
   * A pointer showing the Temi’s current position.
   * A marker indicating the destination.
3. **Tour Section**: This feature was intended to provide a guided tour but was not fully implemented during development.

The second application, referred to as Intro, had a single functionality. It played an introductory video for one of the exhibits featured at OH2024. The home screen is shown in Figure 9 (Appendices). The third application, referred to as Library, was developed for NYP’s library. It shared several capabilities with the Main application, albeit tailored to the library’s environment and requirements. The home screen is shown in Figure 10 (Appendices).

It is important to note that these applications were not part of the initial project scope. Consequently, the specification report created during the early stages of the project did not account for these deliverables or include corresponding milestones. For the OH2024 applications, a list of specifications was later provided by a team at NYP, outlining the required features for the Temi robots to be used during the event. These specifications subsequently served as the milestones for the project.

Table 4: Key Milestones for the OH2024 Application

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Purpose | Alignment with Objectives | Importance |
| Main | | | |
| Greeting Module | Automatically greets visitors within a 2-meter radius, creating a welcoming and interactive experience. | Enhances visitor engagement by providing a friendly and approachable first impression. | Needed |
| Randomized Movement | (Optional) Allows Temi to move randomly within the defined area to increase visibility and interaction opportunities. | Improves visibility and interaction opportunities by making Temi more dynamic and noticeable. | Low |
| Button Prompt System | Provides users with an interactive button-based interface for selecting functionality. | Central to offering user-friendly interaction, enabling access to key functionalities like Q&A, directions, and tours. | Needed |
| Button 1: Q&A | Enables users to ask or select questions about Temi’s capabilities, with responses provided through preloaded data or text-to-speech. | Educates visitors about Temi’s capabilities, showcasing its interactive features and potential. | High |
| Button 2: Directions/Locations | Offers navigation to specific locations within S Block with verbal guidance and a button-based selection of four destinations. | Facilitates efficient navigation, improving the visitor experience by guiding them to key areas. | Needed |
| Button 3: Tours | Provides guided tours of S Block, including commentary about four key locations. | Enhances visitor engagement by showcasing the area in an informative and interactive manner. | High |
| Intro | | | |
| ****Introductory Video (Intro App)**** | Plays an introduction video about RoboMasters. | Provides event-specific content to engage visitors. | Needed |
| Library | | | |
| ****Library Navigation**** | Guides Temi to specific locations in the library (e.g., Collections, Facilities, Spaces). | Meets the library's operational requirements. | Needed |
| ****Scheduled Start and Patrol**** | Automatically starts at 8:30 AM and moves to entrance to start and transitions to patrols at 11:00 AM. | Aligns with the library's daily schedule and operational flow. | High |
| ****Entrance Alignment Adjustment**** | Ensures Temi faces straight at the library entrance. | Enhances the user experience and professionalism. | Medium |

## 

## Planning and Designing

Before planning, the first thing that was done was to try and develop an idea of what each stakeholder wanted. As there were two main projects being developed, there were two main stakeholders that were not connected to each other. The stakeholders concerned with the development of the OH2024 application will be referred to as OH2024, and the stakeholders for the Library will be referred to as the Librarians.

The introduction of developing features for OH2024 and the Library was not done simultaneously. The OH2024 application was developed first, with no prior understanding that an application would also later be requested for the Library. Either way, the process of planning and designing was not overly difficult. Before the development of the OH2024 application, a list of desired features was provided, which can be seen in Table 4. Based on this, a meeting was held with the stakeholder to gain a clearer understanding of their requirements and ideas. From this meeting, each feature was broken down into objectives, which served as a guide for what each feature needed to accomplish.

It should also be noted that most of the features required for the OH2024 application were also needed for the Library application. By the time the Library application was requested, the OH2024 application was nearing completion, allowing the reuse and repurposing of features already developed for OH2024. The primary reason for this overlap is that the needs and desires of both stakeholders, while slightly different, were overall very similar. The Librarian wanted the Temi to be able to guide users to locations within the library and answer any questions they had about the library in general.

Table 8 shows the list of objectives for each feature provided. These objectives were decided based on discussions with the relevant stakeholders to gain a better understanding of their requirements. Some important insights emerged from these discussions:

* For the OH2024 main application, it was determined during the meeting that implementing a tour feature was not necessary. Several reasons were cited for this decision. The primary concern was that only one Temi robot was available to handle all features of the main application, and adding a tour feature would occupy the robot for extended periods. Additionally, there were concerns about noise levels affecting the ability of users to hear Temi effectively. The primary reason for excluding the tour was that a feature already existed for Temi to guide people to specific locations, which addressed a significant portion of the functionality the tour application would have offered.

After the discussions, it was possible to design a flow diagram to better outline the logic within the application. These flow diagrams can be seen in the appendices as Flow Diagram 1, Flow Diagram 2, Flow Diagram 3, and Flow Diagram 4.

For the Intro application, the stakeholder provided a script that needed to be included in the video. This script can be seen in Table 9 (Appendices). From this, discussions took place to design a basic storyboard to guide video development. The resulting storyboard can be seen in Figure 11(Appendices).

As seen in Figure 11, the storyboard is messy. There was no need for a polished storyboard as its primary purpose was to help the stakeholder visualize their ideas. Additionally, the short development timeline for the video reduced the need for thorough documentation of the story. Since there was only one developer for the video—who was also responsible for the storyboard—there was no need to communicate ideas with others, allowing the process to remain simple and streamlined.

## Development Process

### Main Application

The development process used for creating the Main and Library applications closely mirrored the approach detailed in the section “Temi Tour SDK”, specifically regarding the software development process. For a more detailed description of the methods used, refer to that section. The only notable variation was that the developed product was presented directly to the stakeholders, who provided input on the changes they wanted. This instead of collecting feedback from testers. Further details on this process will be expanded upon in the “Testing and Validation” section.

### Intro Application

For the development of the Intro video, the script provided in Table 9 (Appendices) was used to create the dialogue for the video. This dialogue was then divided into sections, with each section aligning with an element of the storyboard outlined in Figure 11. This process ensured proper timing of the video elements with the corresponding dialogue.

Each resource required for the video was either created or sourced individually, focusing on one scene at a time. Once a scene was completed, the same method of resource collection was repeated for the next scene. Basic animations were created using a combination of scene transitions available in Microsoft Clipchamp and custom animations designed in Microsoft PowerPoint.

An example of how a storyboard element (Figure 11) was transformed into a final video scene can be found Figure 12 (Appendices). The tools used for video creation, as outlined in Table 6, included W5, W6, W7, W8, and W9. The final video had a total runtime of 1 minute and 21 seconds.

### Library Application

For the Library Application, adapting the application developed from the Main Application posed some difficulties. This was primarily due to the difference in the versions of the Temi robots used. The Main Application utilized one version, while the Library Application used version 2. More details on the differences between these two versions can be found in the “Testing andValidation” section of “Temi Tour SDK”. The main issues arose from the reduced capabilities of version 2.

Since the Library Application required the ability to detect users and greet them, it needed to utilize dynamic mode. However, dynamic mode on version 2 contains a bug that causes the robot to randomly track users, which overrides the movement actions of the Temi. This can result in major issues, especially when trying to get the Temi to reach a specific location successfully.

To mitigate this issue, a workaround was implemented by repeatedly calling the destination for the Temi. The way the Temi determines whether it has reached a location is by using a listener that provides an enum indicating the state of the “go to location” action. The two main states used are complete and aborted. When the bug causes an override, it leads the listener to return an abort state instead of a complete state. By tracking this, the application can determine if the Temi has completed the “go to location” command. Hence, when an abort case occurs, the application is designed to repeat the “go to location” command until it receives a complete state from the listener.

There is another issue present with version 2. The Temi is unable to retrieve its current coordinates, which leads to two main problems. The first issue occurs at 8:30 AM every day, when the Temi needs to wake up and go to the entrance. While it is possible to use a “go to location” command to navigate the Temi to the entrance, tracking whether the Temi stays there after reaching the location is another matter. Without the ability to track the Temi’s current position, if it is moved—whether intentionally, accidentally, or by the bug causing it to track users—the Temi may deviate from its default position. The only way to return the Temi to the entrance is by calling the “go to location” command at frequent intervals. However, this introduces an issue: if the Temi is already at the location when the “go to location” command is called, it causes the Temi’s head to “nod,” which is seen as the screen tilting down and up. The more frequently this “go to location” command is executed, the more frequently the nod occurs. While this is not inherently harmful, it is undesirable from a visual standpoint as requested by the stakeholders and it also causes more wear on the servo motors that control the screen tilt, leading to higher costs down the line.

The second issue arises from the direction/location system in the Main Application, which used the Temi’s coordinates to generate a pointer on the map based on its actual location. Since version 2 cannot retrieve its coordinates, this feature had to be removed. As a result, only the position of the desired destination is shown on the map. Further issues with the direction/location system stem from the fact that version 2 cannot perform text-to-speech due to an outdated OS in Temi Centre. This means the system, which previously asked the user if they would like to be shown and required a verbal response, had to be altered. In the Library Application, the system now asks the user if they would like to be shown, and the user confirms their choice by selecting one of two buttons for “Yes” or “No.”

Another major issue arose from the outdated OS, which prevented the ability to open the application at set times through the Temi Centre Website. This functionality is important because it allows the application to open as long as the Temi is awake, even if the Android device is powered off, turned on, or reset. The Temi Centre Website allows the creation of "sequences," a more visual way of telling Temi what to do, which does not rely on the Android OS. Without this method, there was a reliance on trying to open the application through the Android OS. Methods such as broadcast receivers and alarm managers, designed to set alarms at specific times and listen for external events, were tested. However, it is important to note that the Android OS is designed for phones and tablets, where applications can be distributed via the Play Store to a broad range of users. To prevent malicious elements from being added to applications, Android restricts certain functions from running in the background or when the app is closed. Through testing various methods, it became clear that it is not possible to have an application open itself at set times, as this feature can pose significant security concerns for most users. This is one of the main disadvantages of developing applications on the Android OS.

In addition, there was a change in how the greeting module works: instead of moving, the Temi now remains stationary. Along with this change, the button prompt system was replaced by a poster that outlines the basic functions of the application. The poster can be seen in Figure 13(Appendices).

Table 6: Tool used to develop the OH2024 and Library Applications

|  |  |  |
| --- | --- | --- |
| Category | Tool/Resource | Purpose |
| Hardware | W1: Temi V2 and V3 | Primary devices for testing and deploying the SDK. |
|  | W2: Computer with a Local Wi-Fi Network | Facilitated a stable connection between Temi devices and the development environment. |
| Software | W3: Platforming Tool | Connected Temi devices to the computer for development and debugging. |
|  | W4: Temi Center Website | Used for creating and editing maps for navigation. |
|  | W5: Dzine | Software for changing the look of images; used to achieve the cyberpunk design. |
|  | W6: Microsoft PowerPoint | Designed sprites and animations for the application; also used to create templates for Dzine. |
|  | W7: GIF Creation Tools (e.g., GIFHY) | Generated and managed GIFs for animations within the application. |
|  | W8: Microsoft Clipchamp | Used to make videos for the application. |
|  | W9: YouTube | Sourced royalty-free sound effects and music for the application. |
| API / Online Services | W10: ChatGPT API Key | Integrated AI-driven functionality to enhance user interaction capabilities. |

## Testing and Validation

# User handbook (2 weeks)

\*\* should get a screen shot of the GitHub once it is all done to use as reference.

\*\* Need to address which milestones where achieved when document completed

## Introduction

One of the key goals of this project is to create documentation that supports and guides future developers working on the Temi platform. The User Handbook serves as a resource, providing information on how to set up a workstation for developing applications for the Temi robot. It outlines the processes required to create applications, details current known issues with the Temi platform and their solutions, highlights the core functions available for development, explains how to structure and layout a project file, and provides an analysis of the code developed during the project.

This deliverable addresses a critical gap in the resources available for Temi development. Currently, documentation for utilizing the Temi SDK is limited. While there are general guides for Android Studio, the specifics of working with the Temi SDK provided by Temi Centre are not well-documented. The primary sources of information are the Temi website and a GitHub repository, which offer only general descriptions of the SDK’s functions without in-depth implementation guidance. As a result, through the course of the project significant challenges in understanding how to effectively utilize the SDK occurred. This often led to relying on trial and error to achieve objectives. This process was proven to be time-consuming and inefficient. By creating this handbook, the aim is to provide future developers with a structured and accessible resource that consolidates knowledge, reduces onboarding time, and enhances development efficiency.

The User Handbook is designed to not only assist with current development efforts but also serve as a living document that can be expanded upon and handed down to future teams. Its purpose is to streamline the development process, reduce the learning curve, and ensure that the foundational work done in this project can be built upon effectively. Table 5 outlines all milestones desired for this deliverable.

Table 7: Key Milestones for the User Handbook

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Purpose | Alignment with Objectives | \*\*Importance |
| UH1: Workstation Setup Guide | Provides step-by-step instructions for setting up a workstation for Temi development. | Ensures future developers can start working with minimal setup time. | Needed |
| UH2: Application Creation Process | Outlines the process for creating new applications for the Temi platform. | Streamlines development and reduces errors in application creation. | Needed |
| UH3: Known Issues and Solutions | Documents common issues with the Temi SDK and their solutions. | Saves developers time by addressing frequently encountered problems. | High |
| UH4: Core Function Documentation | Provides descriptions and examples of core functions available for Temi applications. | Facilitates effective use of the Temi SDK and encourages code reusability. | High |
| UH5: Project Structure Guidelines | Details best practices for organizing and structuring project files. | Ensures projects are maintainable and understandable for future teams. | Medium |
| UH6: Code Analysis | Includes an analysis of the code developed during the project. | Helps developers understand design decisions and implementation methods. | Medium |
| UH7: Expansion and Customization | Provides guidance on expanding or customizing existing functionality. | Encourages innovation and iterative development. | Low |
| UH8: Feedback and Updates Section | Allows future teams to document additional findings or improvements. | Ensures the handbook evolves with the needs of future developers. | Low |

\*\* **Importance Scale:** Needed, High, Medium, Low

## Planning and Designing

The development of the User Handbook required a clear understanding of its purpose and the outcomes it was meant to achieve. From the outset, it was established that the handbook needed to serve as a comprehensive resource for developers working on future projects involving the Temi robot. However, the initial phase lacked a clear structure and specific steps to accomplish this goal.

To address this, time was dedicated to outlining a plan that would guide the creation process and ensure the deliverable met its objectives. The first step was identifying the core milestones seen in Table 5, which acted as a framework for organizing the content. These milestones were designed to define what the handbook needed to include for it to be a useful source.

Another critical aspect of planning involved determining the style and layout of the handbook. The design needed to balance readability and professionalism, ensuring that the content was accessible to both experienced developers and those new to the Temi platform. Considerations were made for formatting, use of diagrams or visuals, and how information would be logically grouped to maximize usability. The milestones and design approach provided a foundation for creating a user-friendly and informative handbook that could be expanded upon as future projects evolved.

## Features and Functionalities

Due to the nature of the deliverable being a document, there are limited features and functionalities. However, several key considerations were made to ensure the document could be easily expanded and maintained for future developers. The primary concern was how to allow other developers to access and contribute to the document easily while maintaining version control. This led to the decision to host the document on GitHub, which provided several benefits:

1. **Collaborative Access**: GitHub allows developers to independently contribute to the document, ensuring seamless expansion and collaboration.
2. **Version Control**: GitHub’s versioning system provides a simple way to revert to older versions if needed, safeguarding against undesirable changes.
3. **Centralized Repository**: Storing the documentation and resources in one location simplifies access for developers, ensuring they have all current information on the Temi SDK in one spot.

Another key feature was the creation of a formatting guide, which provides a standardized method for developers to integrate content into the main document. The goal was to create a universal approach that multiple generations of developers could follow to add content in a way that maintains consistency and ensures the document remains easy to navigate.

## Testing and Validation

For testing and validation, the focus was on ensuring that the document was accessible to its target audience—developers with a basic understanding of coding, but no prior experience with Android Studio or the Temi SDK. The key elements of the document tested were how to set up a development workstation, how to format an application, and how to upload an application to a Temi. In line with the milestones from Table 4, this testing focused on **UH1** and **UH2**.

The testing process involved providing the user handbook to a tester and instructing them to set up the workstation, format an application, and upload it to the Temi. The tester was informed that they could not ask for help during the process, simulating a real-world scenario where users must complete tasks independently. Afterward, the tester provided feedback on their experience, which was then reviewed to identify any issues or suggestions for improvement. Based on the feedback, necessary changes were made to improve the document.

While it would have been ideal to test the effectiveness of the sections on **code analysis** and **future content integration**, this would have required more time for testers to become familiar with Temi development. Given the limited time available for testing, it was not possible to include this more in-depth validation. However, the testing that was conducted confirmed that the essential setup and application creation steps were clear and accessible to the target audience.

# Transferable Skills Reflection

Figure 4 transferable skills assessment radar graph

Due to the reflective nature of this section, I will be using first-person perspective to discuss my skills and development over the course of my placement. One of the expectations for the placement was to complete three Transferable Skills surveys, which encouraged personal reflection on growth. Table 7 (Appendices) contains all the skills used for reflection. As there are numerous skills, it was decided that for the purpose of this report, similar skills would be grouped into broader categories called skill groups. These groups will then be used to provide a more general overview of my skill development. It is import to note that the vast majority of the project development only involved me with some input from other parties.

Each metric is scored from 0 to 10, reflecting my confidence in each skill group and how prepared I feel for entering the industry. A score of 0 represents no confidence in the skill, 5 indicates the minimal expectation I have for entering the industry, and 10 signifies that I feel confident in the skill with no further development needed. A reflection on each of the skill groups has been done and stored in Table 7 with metric values in Figure 4.

Table 8 list of all skill groups from Table 7 with reflection on each based on the WIL experience

|  |  |
| --- | --- |
| Skill Group | Reflection of WIL Experience |
| Collaboration and Teamwork | During my placement, I had the opportunity to consult with my Industry Supervisor to discuss ideas and receive guidance. I also briefly collaborated with an intern at another company, Robosolutions, on a project. These interactions provided exposure to teamwork and collaboration in a professional setting. However, most of my work involved independent tasks, such as planning workflows, brainstorming solutions, and designing deliverables. As a result, while I gained some experience in collaboration, my development in this area was somewhat limited. |
| Communication Skills | A key element of my placement was focused on report creation and documentation, particularly developing user guides, which emphasized written communication. Through this work, I had multiple opportunities to build, test, and refine my communication skills. Additionally, I received valuable feedback, especially given that the project was intended for student outreach. One important insight I gained was learning to analyze feedback deeply rather than accepting it at face value. However, due to the isolated nature of the project, there were limited opportunities for verbal communication or public speaking, areas I would have liked to develop further during the placement. |
| Cognitive and Analytical Skills | Throughout the placement, I developed and honed my cognitive and analytical skills. To complete each deliverable, I needed to conceptualize the final product and break down the steps required to achieve it. This was especially important when using systems, I had no prior experience with, such as Bluetooth, creating UIs, and handling the ChatGPT API. I also had to evaluate various online resources to implement these systems into the application, which was not always straightforward and required extensive research. Additionally, my skills in analyzing and diagnosing problems were tested repeatedly, as there were many instances where bugs emerged in the application. Identifying and fixing these bugs greatly improved my skills in this area. |
| Efficiency and Productivity | This placement provided a lot of autonomy in determining how each deliverable would be done and the elements involved. There was no allocated budget, meaning there was no need to address costs associated with actions taken. The explorative nature of the project also allowed for flexible scheduling. As a result, the timing of tasks was largely up to me, which required effective time management to ensure that milestones were met. While I sometimes felt the quality and efficiency of my work could have been better—especially when dealing with unfamiliar features, such as developing Bluetooth connectivity—I was able to complete the project on schedule and within the required period. |
| Creativity and Innovation | The freedom provided for development and the fact that few people had gone as deep into Temi application development as I did offered a unique opportunity. When issues arose, there were often no existing resources to consult. This required me to identify the causes of problems and create solutions. Additionally, I developed systems for the Temi that were previously not possible, such as cross-Temi communication using the ChatGPT API. The project challenged me to think innovatively and create solutions that helped achieve the project's goals. |
| Personal Development | The project required a set of skills that I initially lacked, such as understanding application development, managing APIs, and integrating complex features like facial recognition. It was daunting at first, but overcoming challenges and developing new skills was highly rewarding. While the placement didn't offer many opportunities for more formal work settings, I gained greater confidence in my ability to acquire new skills and solve problems independently. |
| Stress and Emotional Regulation | The most stressful aspect of the project occurred at the beginning, when I faced a steep learning curve to develop features for the Temi. However, the lack of strict deadlines and the freedom to complete tasks at my own pace made managing stress easier. The project allowed me to balance my time and focus on problem-solving without the pressure of constant deadlines. |
| Social Responsibility and Ethics | There was minimal focus on social responsibility and ethics due to the project’s nature as a student outreach initiative. The main way this was address was focusing on how students interacted with the Temi and considered the best methods for integration to ensure responsible and ethical use. Peoples willingness to interact with the Temis was not high was surprising, but makes sense as the implementation of cobot like Temi is limited making it alien to most people. |
| Self-Confidence | Given the progress I made during the placement, including developing the application and documentation, I am satisfied with my achievements. Although I still have much to learn, the soft skills I gained will be valuable in future endeavours. In terms of technical skills, I believe the knowledge acquired during the placement will be of great use in the direction I want my career to take. |

# Recommendations

During the course of this project, the Temi Version 2 robot presented significant challenges that hindered its effectiveness in development and deployment. The primary limitations include:

**Performance Issues**

* **Slow Upload Times**: Applications, even small ones, experienced upload times ranging from 1–2 minutes, leading to inefficiencies during testing and iteration.
* **Timeouts for Large Applications**: The hardware struggled with handling larger applications, often resulting in frequent timeouts and disrupted workflows.
* **Unresponsive UI**: Navigating through the in-built system on the Version 2 was sluggish, further complicating the testing process.

**Compatibility Challenges**

* **Outdated OS**: The Version 2 runs an outdated operating system provided by Temi Centre, which restricted its functionality. Applications developed on the Version 3 often faced compatibility issues when deployed on the Version 2. Specific limitations included:
  + Inability to perform text-to-speech.
  + Inability to use the ChatGPT API for natural language processing.
  + Lack of access to local coordinate systems for navigation.

These limitations severely impacted the Temi's ability to engage effectively with its environment and users, which is critical for successful student outreach.

**Dynamic Mode Bug**

* A critical bug in the Version 2's *Dynamic Mode*, a system designed for user detection, caused the robot to randomly start tracking users. This behaviour often overrode other system commands, including navigation.
* When the robot was in motion during outreach events, this bug increased the likelihood of collisions and reduced its ability to perform its intended tasks reliably.

**Recommendation**

Upgrading the Temi Version 2 to Version 3 is strongly recommended for the following reasons:

1. **Improved Development Efficiency**:
   * The Version 3 resolves all performance-related issues, including upload times, timeouts, and UI responsiveness.
   * A faster and smoother development process allows for greater focus on enhancing application functionality.
2. **Expanded Functionality**:
   * With a modern OS, the Version 3 unlocks advanced capabilities, including reliable text-to-speech, ChatGPT API integration, and robust navigation systems.
   * These features enhance the robot’s interactivity, making it more engaging and versatile for outreach programs.
3. **Increased Research Opportunities**:
   * Deploying multiple Version 3 units would enable the exploration of advanced functionality, such as cross-communication between Temis for synchronized tasks and collaborative behaviours.
   * For example, multiple Temis could work together to provide dynamic tours, interact with groups, or share tasks in real-time.

It is also recommended to allocate more time to the development of the quiz application and the Kit-Kat dispenser. While these systems are functional, their reliability leaves much to be desired. The primary issue lies in the unstable connection between the Bluetooth module controlling the dispenser and the Temi robot. Currently, there is no mechanism in place to automatically reconnect in the event of a disconnection. Additionally, the motor controlling the dispenser tends to go slack intermittently, the cause of which remains unknown. These issues significantly reduce the autonomy of the system, as constant monitoring is required to address faults, which is far from ideal. Future enhancements should focus on the following:

1. **Code Restructuring**: Refactor the existing codebase to improve readability and simplify future expansions.
2. **Connection Reliability**: Develop a robust system to handle Bluetooth reconnections seamlessly and minimize disruptions.
3. **Hardware Stability**: Investigate and resolve the root cause of the motor slack to ensure consistent operation.

Addressing these issues will not only enhance the reliability of the Kit-Kat dispenser but also align its functionality with the overall goal of creating an autonomous, user-friendly system.

# Conclusion

DICTIONARY

Process: this is a set actions done by the temi to be able to complete a certain action

Sequence: this is a method of creating basic tours on the temi center.

The goal was to develop a similar product to what they created but improved capabilities.

[An Executive Summary (no longer than a page) that describes the main work undertaken, the achievements, and other significant outcomes (\*)]

What are the projects that you have gotten done during the entirety of the project?

* Created a quiz
  + Features locked behind password system.
    - System has a cancel and a confirmation.
      * Cancel will allow exit of the password prompt.
      * Ok will allow confirmation of password entered. Will do nothing if incorrect password added. If correct, it will close and procced onwards.
    - Allows inputting of the password using a touch keyboard.
  + Scoreboard
    - Pressing on a name five times will allow a password protected prompt to remove the name. All names that are similar are removed and replaced with empty.
    - All attempt is ranked from highest to lowest, with each record containing the user name and their score.
    - When there are too many user names on the board, the system will create a method to allow scrolling. Allows viewing of all names in the board.
    - When entering the score board, it will say a line of dialogue introducing the user to the scoreboard.
  + Multiple language types
  + Data method
  + Clear method
  + Bluetooth selection
  + Idle state
  + Anti-misuse system
* Created an emotional detection system with collaboration with Heng Junxiang

Section 1: Software Development Kit (SDK)

1.1 Introduction

Briefly introduce the purpose of the SDK.

Explain its significance to the project goals (e.g., enabling customizable guided tours, enhancing user interaction with Temi).

Outline the main objectives of the SDK development.

1.2 Planning and Design

Describe the initial planning process, including any research conducted (e.g., SDK documentation, existing implementations).

Detail the design goals and considerations (e.g., flexibility, ease of use, scalability).

Include diagrams or flowcharts, if applicable, to illustrate the SDK’s architecture.

1.3 Features and Functionality

List the key features of the SDK and explain their purpose.

Describe how these features align with the project’s objectives (e.g., enhancing outreach activities).

Highlight any innovative aspects of the SDK.

1.4 Development Process

Provide a step-by-step overview of the SDK development process.

Tools and Technologies: Outline the software and hardware tools used.

Challenges: Discuss any difficulties faced during development and how they were addressed.

Milestones: Highlight major development milestones and their timelines.

Mention collaboration with other team members if applicable.

1.5 Testing and Validation

Explain the testing methodologies used to validate the SDK’s functionality (e.g., unit testing, integration testing).

Provide test cases or scenarios to demonstrate how the SDK meets its design goals.

Share results or outcomes from the validation process.

1.6 Implementation

Detail how the SDK integrates with Temi’s existing capabilities.

Provide examples of usage (e.g., snippets of code or user scenarios).

Discuss how the SDK enables the creation of customized tours.

1.7 Challenges and Solutions

Identify any major challenges encountered during development.

Discuss the solutions implemented to overcome these challenges.

1.8 Impact and Significance

Explain the broader implications of the SDK for NYP’s outreach efforts.

Highlight how the SDK contributes to achieving the overall project goals.

1.9 Conclusion and Next Steps

Summarize the SDK’s achievements and its alignment with project objectives.

Suggest recommendations for future development or enhancements to the SDK.

# Appendices



Figure 5: image of LIDAR present on the Temi robots. Same model is used across version 2 and 3.

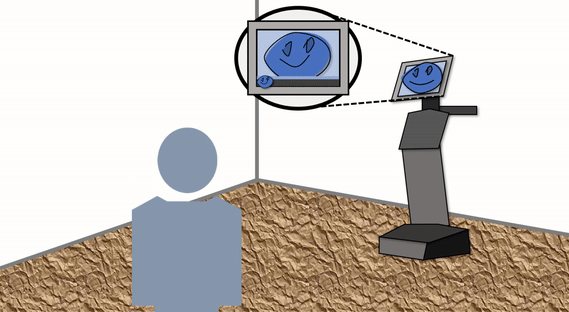


Figure 6: exert from media developed for the Temi Tour SDK to illustrate how to talk to temi.

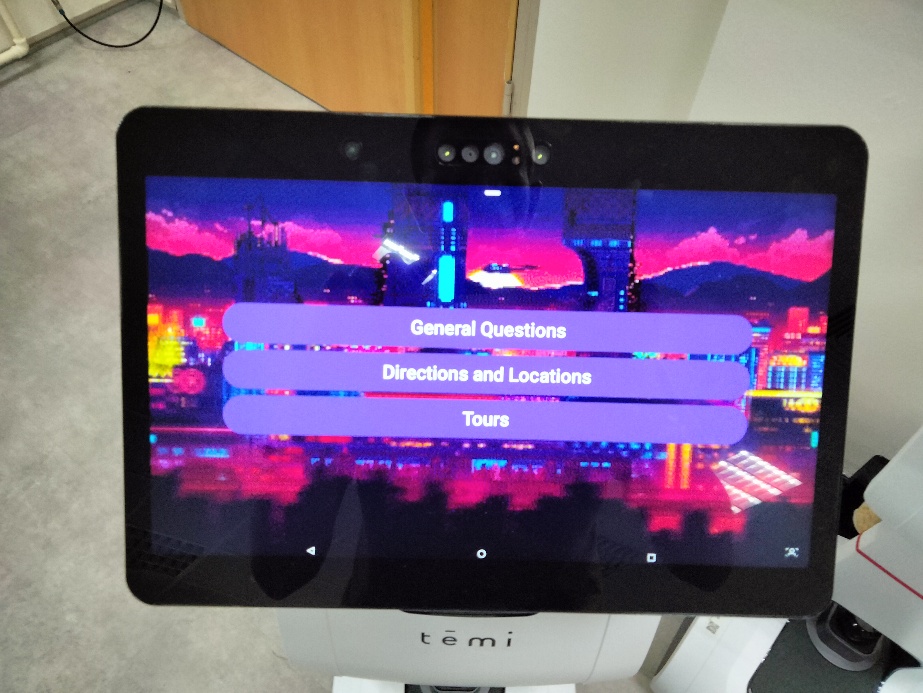


Figure 7: Home screen for the Main application.

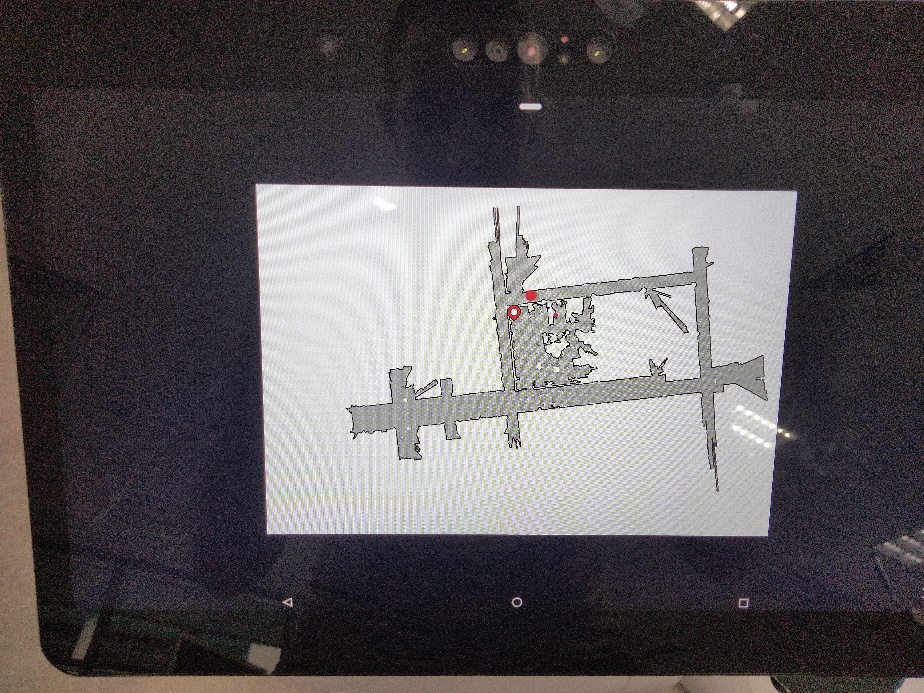


Figure 8: Example of the map system. The pointer indicates the Temi’s location and the red dot the location desired to go to.



Figure 9: Home screen for the Intro application.

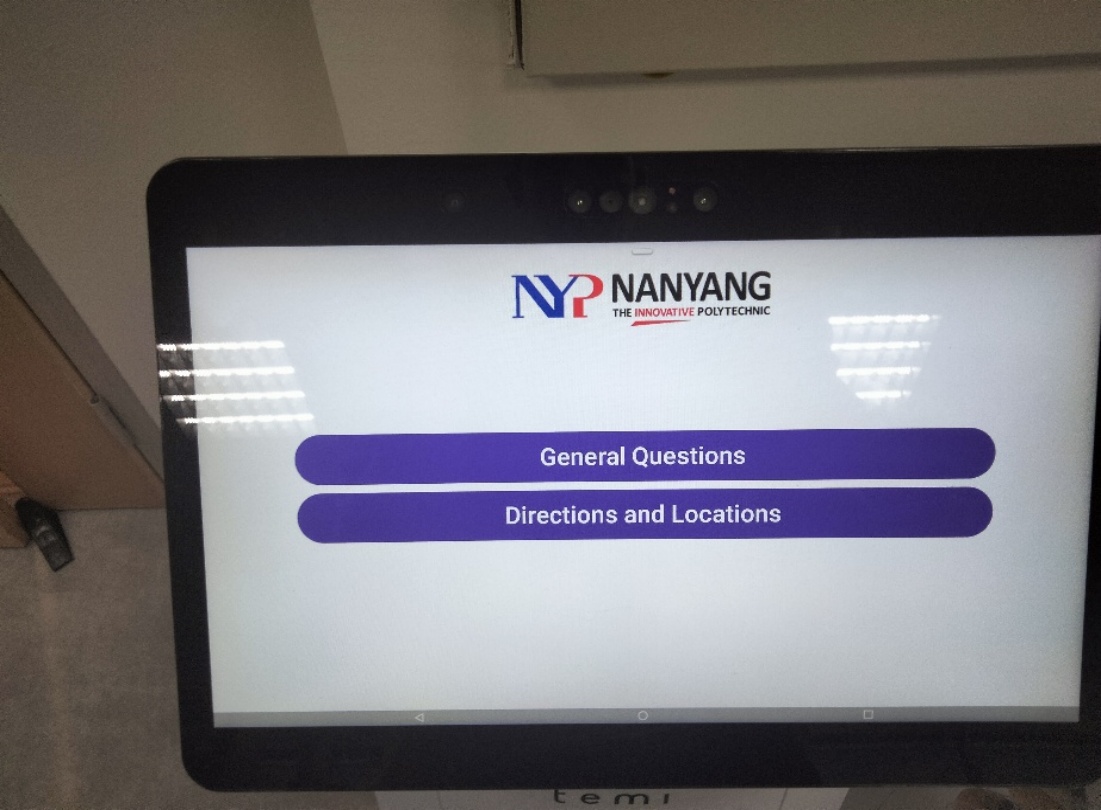


Figure 10: Home screen for the Library application.

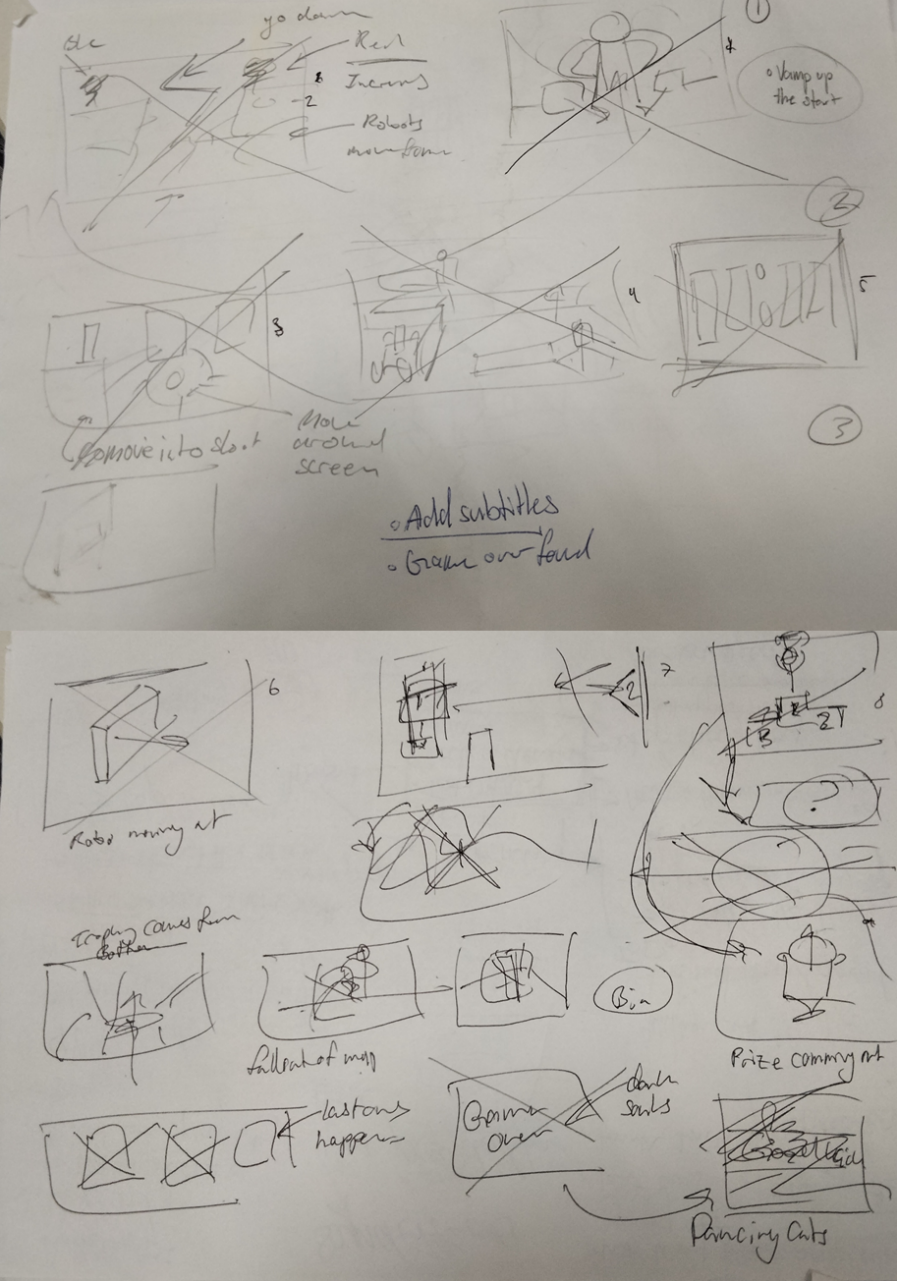


Figure 11: Story board for the Intro Application. The story boards are read from left to right and top to bottom.

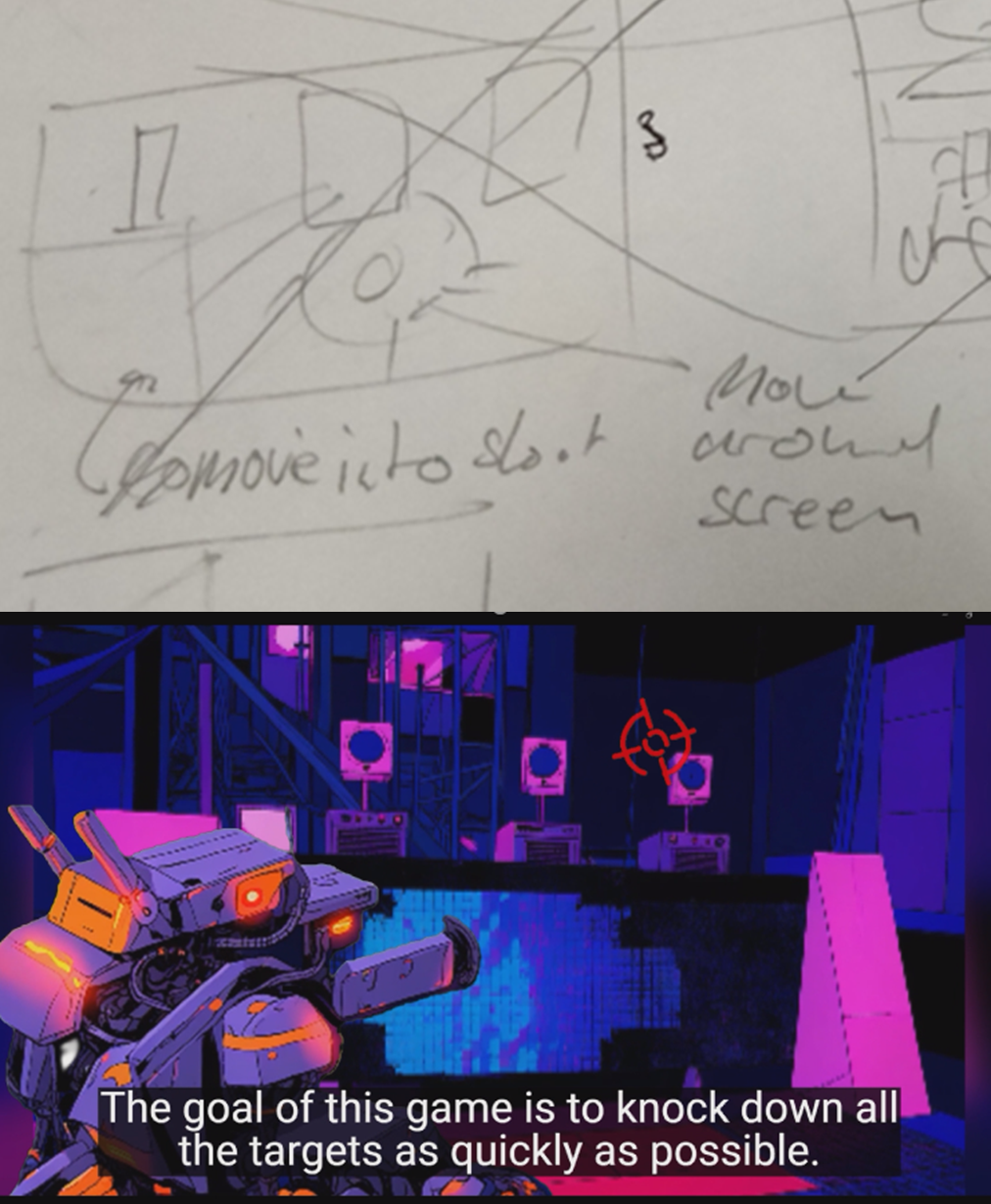


Figure 12: Comparison of slide 3 in the story board from Figure 11 to the developed scene for the video in Intro Application.

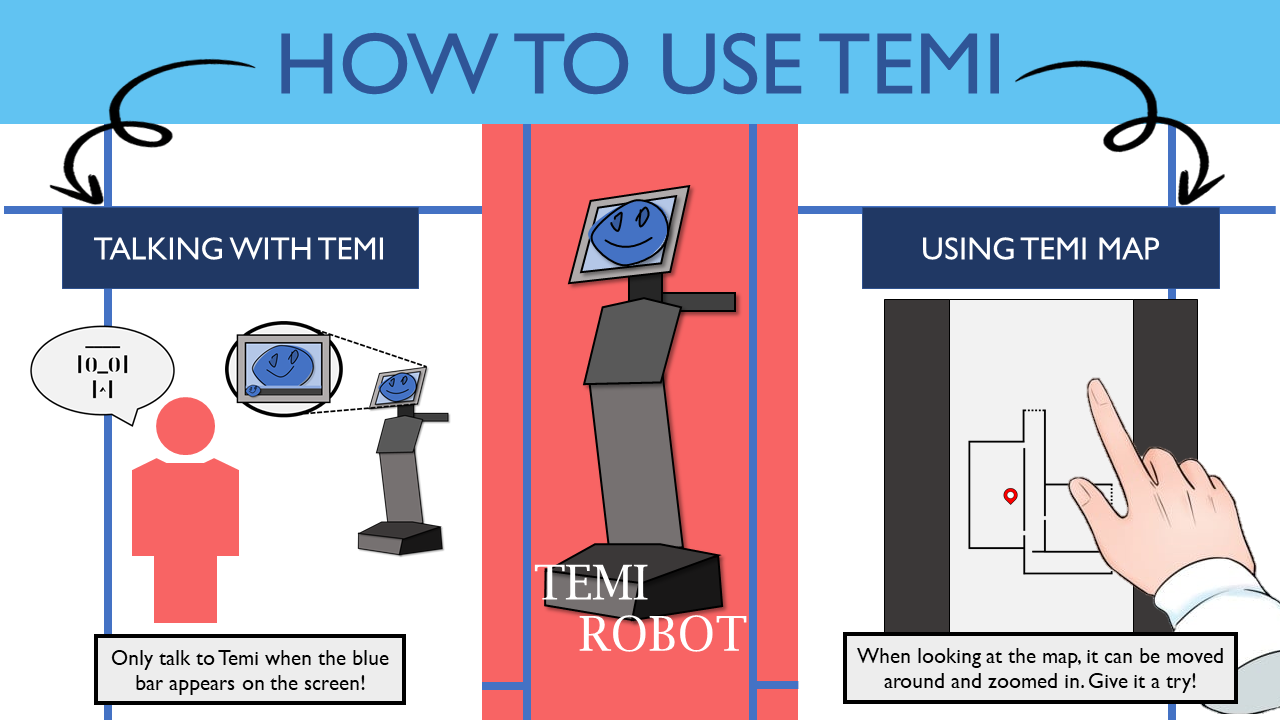


Figure 13: Poster developed for the Library application

Table 9: Metrics used in the Transferable skills survey.

|  |  |  |
| --- | --- | --- |
| Group | **skill** | Explanation |
| Collaboration and Teamwork | **Task Collaboration** | Complete group tasks through collaborative communication, problem solving, discussion, and planning. |
| **Team Working** | Operate within, and contribute to, a respectful, supportive and cooperative group climate. |
| **Social Intelligence** | Acknowledge the complex emotions and viewpoints of others and respond sensitively and appropriately. |
| **Cultural and Diversity Awareness** | Work productively with people from diverse cultures, races, ages, gender, religion, and lifestyles. |
| **Influencing Others** | Defend and assert rights, interests, and needs and convince others of the validity of one’s point of view. |
| **Conflict Resolution** | Address and resolve contentious issues with key stakeholders. |
| Communication Skills | **Verbal Communication** | Communicate orally in a clear and sensitive manner, appropriately varied according to different audiences and seniority levels. |
| **Giving and Receiving Feedback** | Give and receive feedback appropriately and constructively. |
| **Public Speaking** | Speak publicly and adjust style according to the nature of the audience. |
| **Meeting Participation** | Participate constructively in meetings. |
| **Written Communication** | Present knowledge in a professional, structured, and clear manner in a range of written formats. |
| Cognitive and Analytical Skills | **Conceptualization** | Recognize patterns in detailed documents and scenarios to understand the ‘bigger’ picture. |
| **Evaluation** | Recognize, evaluate, and retain key points in a range of documents and scenarios. |
| **Reasoning** | Use rational and logical reasoning to deduce appropriate and well-reasoned conclusions. |
| **Analyzing and Diagnosing** | Analyze facts and circumstances and ask the right questions to diagnose problems. |
| **Decision Making** | Make appropriate and timely decisions, in light of available information, in sensitive and complex situations. |
| Efficiency and Productivity | **Efficiency** | Achieve prescribed goals and outcomes in a timely and resourceful manner. |
| **Multi-tasking** | Perform more than one task at the same time. |
| **Autonomy** | Complete tasks in a self-directed manner in the absence of supervision. |
| **Quality of Work** | Complete work to a high-quality standard aligned to expectations. |
| **Time Management** | Manage time to achieve agreed goals. |
| **Commercial Awareness** | Aware of commercial viability or cost considerations. |
| **Drive** | Go beyond the call of duty by pitching in, including undertaking menial tasks, as required by the business. |
| **Goal and Task Management** | Set, maintain, and consistently act upon achievable goals, prioritized tasks, plans, and realistic schedules. |
| Creativity and Innovation | **Innovation** | Contribute towards the development of new products, services, or technologies (e.g., software, applications, devices). |
| **Entrepreneurship/Intrapreneurship** | Initiate change and add value by embracing new ideas and showing ingenuity and creativity in addressing challenges and problems. |
| **Lateral Thinking/Creativity** | Develop a range of solutions using lateral and creative thinking. |
| Personal Development | **Employability** | Believe that you are employable. |
| **Meta-Cognition** | Reflect on and evaluate personal practices, strengths, and weaknesses in the workplace. |
| **Lifelong Learning** | Actively seek, monitor, and manage knowledge and sustainable opportunities for learning in the context of employment and life. |
| **Career Management** | Develop meaningful and realistic career goals and pathways for achieving them in light of labor market conditions. |
| **Self-efficacy** | Be self-confident in dealing with the challenges that employment and life present. |
| Stress and Emotional Regulation | **Stress Tolerance** | Persevere and retain effectiveness of well-being and strive to maintain a productive balance of work and life. |
| **Self-Regulation** | Reflect on and regulate emotions and demonstrate self-control. |
| Social Responsibility and Ethics | **Social Responsibility** | Behave in a manner that is sustainable and socially responsible (e.g., consistent with company policy and/or broader community values). |
| **Accountability** | Accept responsibility for own decisions, actions, and work outcomes. |
| **Personal Ethics** | Remain consistently committed to and guided by core values and beliefs such as honesty and integrity. |
| **Organizational Awareness** | Recognize organizational structure, operations, culture, and systems, and adapt behavior and attitudes accordingly. |
| Self-Confidence | **Self-Confidence (Transferable Skills)** | How good you think you are in applying transferable skills in your everyday practice. |
| **Self-Confidence (Technical Skills)** | How good you think you are in applying technical skills in your everyday practice. |

Table 10: Temi Interaction and Functionality Feature Objectives for OH2024 and Library Application development

|  |  |
| --- | --- |
| Feature | Objectives |
| Main | |
| Greeting Module | - Idle face displayed when no user is detected. - 5-second delay before greeting to prevent false positives. - If not interacted with for 10 seconds, idle mode resumes with a 5-second detection cooldown. - After greeting, idle face replaced with the button display prompt. |
| Randomized Movement | - Idle face displayed when no user is detected. - 5-second delay before greeting to prevent false positives. - If not interacted with for 10 seconds, idle mode resumes with a 5-second detection cooldown. - After greeting, idle face replaced with the button display prompt. |
| Button Prompt System | - Three buttons for main functions with a home bar for navigation. - Buttons: purple with white text, black background; home bar: white. - Sound effects for button presses and home button (different sounds). |
| Button 1: Q&A | - Central "Ask Question" button triggers text-to-speech and a "thinking" screen. - Uses ChatGPT for “Ask Question” responses. - Pre-answered questions displayed in a scrollable list, with optional GIFs/images. - Selected pre-answered questions trigger Temi to read aloud and display subtitles matching dialogue timing. - Dialogue stops when another question is selected or when exiting. - Preloaded Q&A includes general Temi questions and answers from Quiz and Kit-Kat applications. - Subtitles displayed for all dialogues. |
| Button 2: Directions/Locations | - Displays four location-based buttons. - Clicking a location shows a zoomable map with Temi's current position and the target location. - Asks users for verbal confirmation to guide them to the location. - If confirmed, leads the user, updating Temi’s position on the map. - On arrival, informs the user and exits to the main menu. - If rejected, allows map navigation or exit by touch. - Map functionality includes zooming and sliding. - Inquiry must finish before the map can be exited. |
| Button 3: Tours | - **Cancelled**: Tours were removed due to time constraints, limited Temi availability, and environmental noise. Directions/Locations filled the need partial need. |
| Intro | |
| ****Introductory Video (Intro App)**** | - Create a storyboard for the introductory video to be developed. |
| Library | |
| ****Library Navigation**** | - Reuse OH2024 navigation system. - Alter menu to include categorized locations (e.g., Collections, Facilities, Spaces) for navigation. - Q&A modified for library use by removing irrelevant questions (e.g., Quiz, Kit-Kat). - "Ask Question" removed due to API cost and compatibility issues with Version 2. |
| ****(ADDED)**** System Control | - Application closes automatically at 15% battery to allow Temi to return to its charging bay. - Application closes automatically at 11:00 AM. |
| ****Scheduled Start and Patrol**** | - Adds sequence to open the application at 8:30 AM. |
| ****Entrance Alignment Adjustment**** | - Adjust Temi’s angle to face straight towards the library entrance instead of being slightly misaligned. |

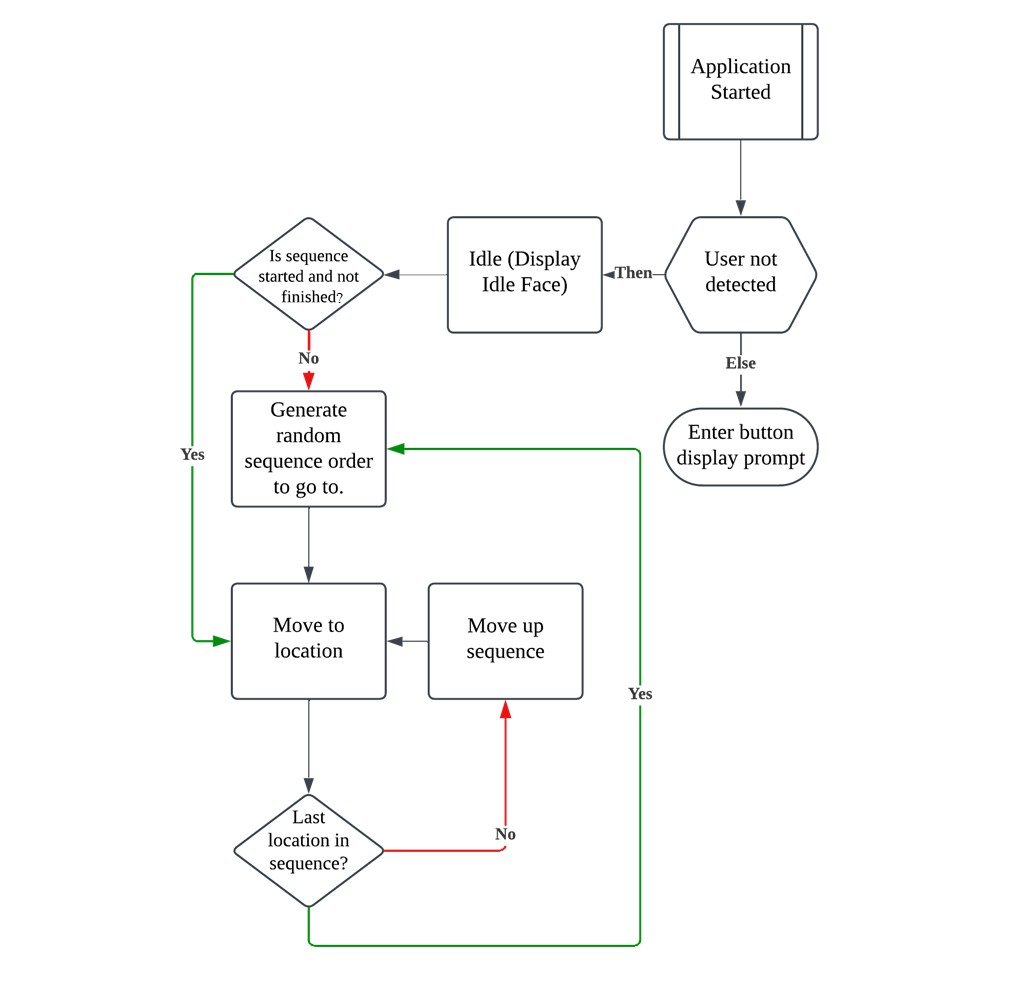
Table 11: Script for video for Main Application

|  |
| --- |
| Droid Battle Arena: Command your Droid  Step into a future where droids carry your load and fight alongside you. Command your droid in an FPV battle to defend Sky City from Imperial forces!  also put in the game play  The goal of this game is to knock down all the targets as quickly as possible.  Participants will start at the same starting point.  Participants will activate the timer to start the run  Participants have to move around the map in order to shoot down the targets  Must follow number order  (new dialogue)  If participants run out of the map, they will be asked to restart.  If map is damaged (pillar knocked down)  If not a lot of people, restart  If have people, continue as we repair  Repeating this 3 times will result in them being disqualified  (new outro) |

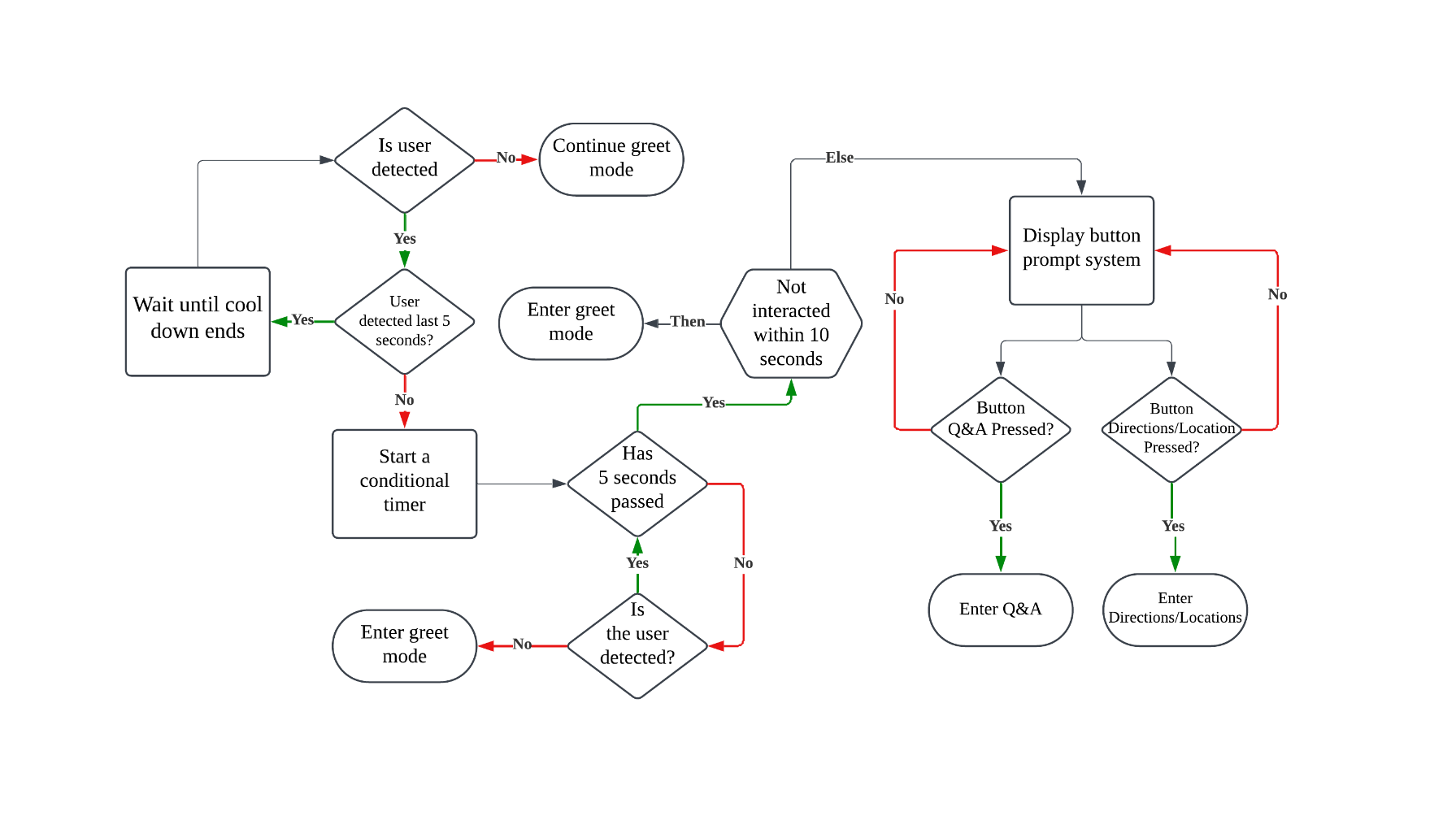
# Flow Diagrams

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Name | Description | Example Usage |
|  | Start | Indicates when the application begins. | "Application startup" |
|  | Action | Represents a set of tasks or actions the Temi is performing. | "Display idle face," "Move to a location" |
|  | Condition | Indicates an "if" case, written as a question to decide between two possible flows (Yes/No). | "Is the user detected?" |
|  | State | Represents a change in the Temi's distinct mode or state, helping to break the flow into sections. | "Switch to Button Prompt display" |
|  | Global Condition | A condition applied constantly across all flows. Uses "then" or "else" for flow transitions. | "User detected: then display greeting, else remain idle." |

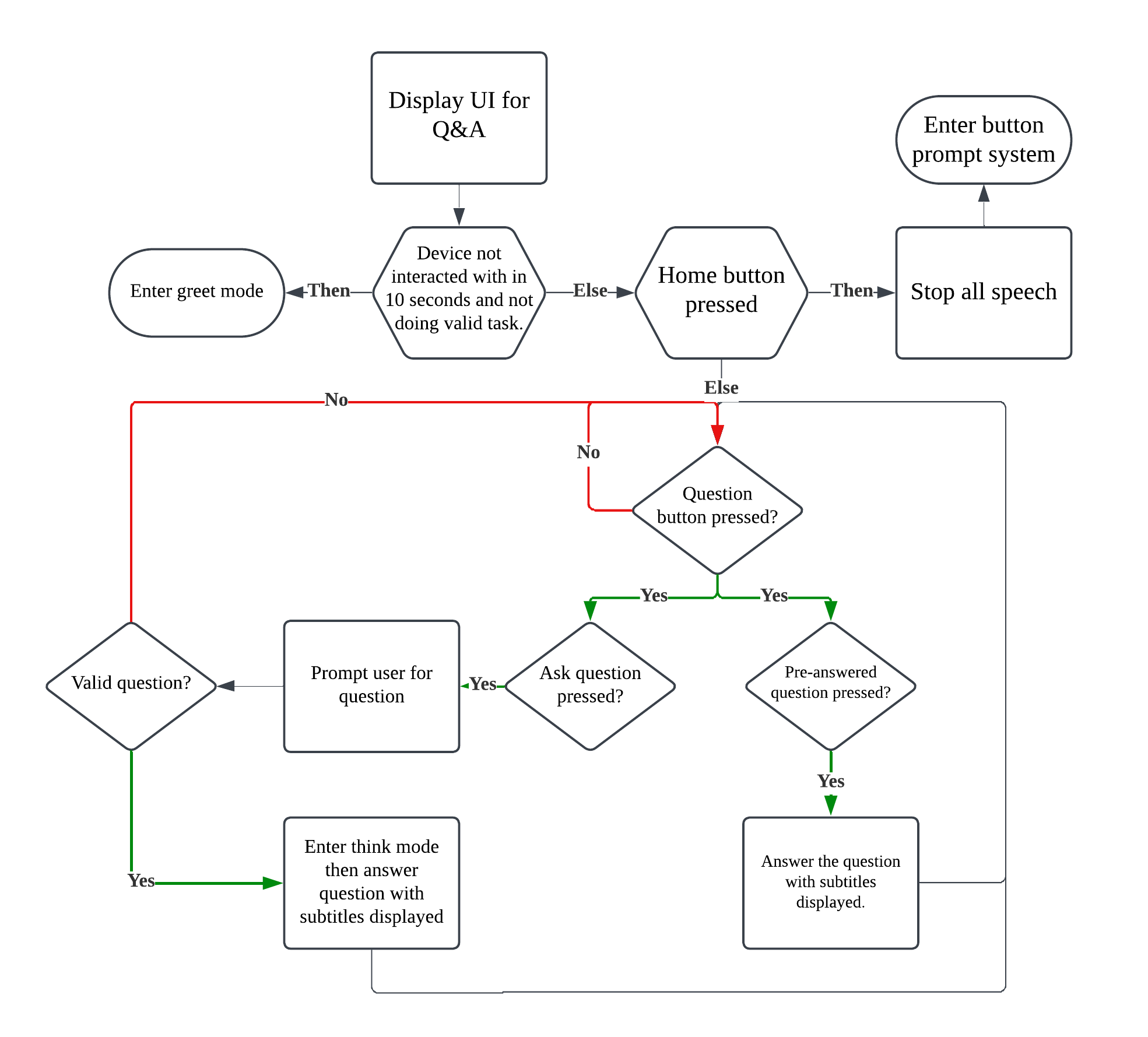
## OH2024 and Library Application development Flow Diagrams



Flow Diagram 1: Main Application - Greeting Module (Greet Mode) Flow Diagram



Flow Diagram 2: Main Application - Button Prompt System (Button Display Prompt) Flow Diagram



Flow Diagram 3: Main Application - Q&A Button (Q&A) Flow Diagram (Button 1)



Flow Diagram 4: Main Application - Directions/Locations Button (Directions/Locations) Flow Diagram (Button 2)

## Code Archive:

## Code Sample 1 Bluetooth communication between Temi (Client End for V2)

|  |
| --- |
| launch {  bluetoothManager.startBluetoothClient(context)  }  **while** (**true**) {  updateGifResource(R.drawable.sleep1)  conditionGate({ bluetoothManager.gate != **true** })  bluetoothManager.changeBlueState(**null**)    updateGifResource(R.drawable.idle)  goTo("engage")  **if** (bluetoothManager.isChatGPT) {  conditionGate({ bluetoothManager.receivedConversation == **null** })  speak(bluetoothManager.receivedConversation)  bluetoothManager.receivedConversation = **null**  bluetoothManager.gate = **false**  conditionGate({ bluetoothManager.receivedConversation == **null** })  speak(bluetoothManager.receivedConversation)  bluetoothManager.receivedConversation = **null**  bluetoothManager.gate = **false**  conditionGate({ bluetoothManager.receivedConversation == **null** })  speak(bluetoothManager.receivedConversation)  bluetoothManager.receivedConversation = **null**  bluetoothManager.gate = **false**  bluetoothManager.isChatGPT = **false**  } **else** {  conditionGate({ bluetoothManager.gate != **true** })  bluetoothManager.changeBlueState(**null**)  speak("What are you doing, I was supposed to take them on the tour, It’s my turn.")  bluetoothManager.changeBlueState(**false**)  conditionGate({ bluetoothManager.gate != **true** })  bluetoothManager.changeBlueState(**null**)  speak("No, no, no, I told you, when it’s my turn to lead the tour, you have to wake me up.")  bluetoothManager.changeBlueState(**false**)  conditionGate({ bluetoothManager.gate != **true** })  bluetoothManager.changeBlueState(**null**)  speak("Fine, but I’m going to tell the creator about this!")  bluetoothManager.changeBlueState(**false**)  }  goTo("home base")  } |

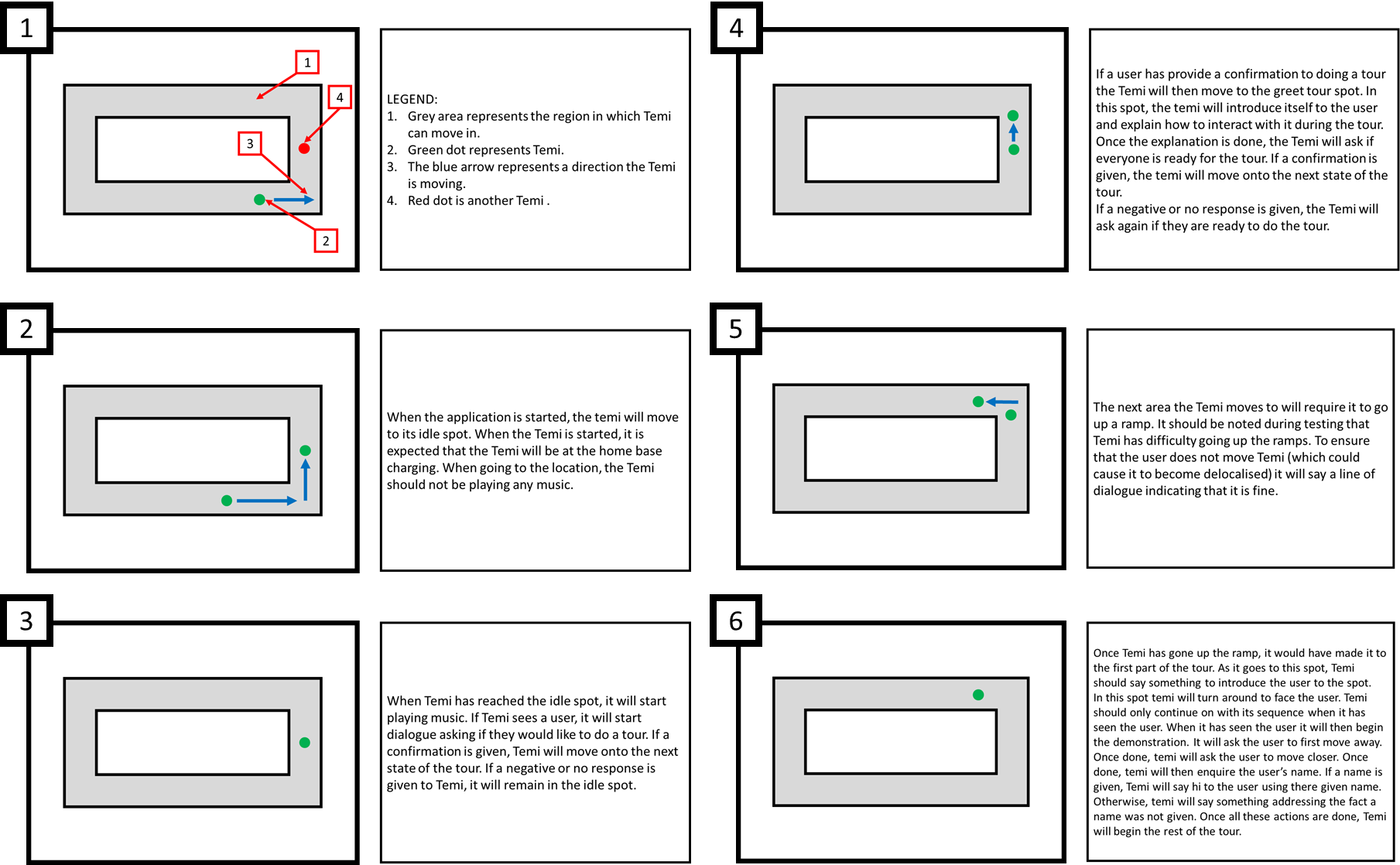
## Code Sample 2 Bluetooth communication between Temi (Server End for V3)

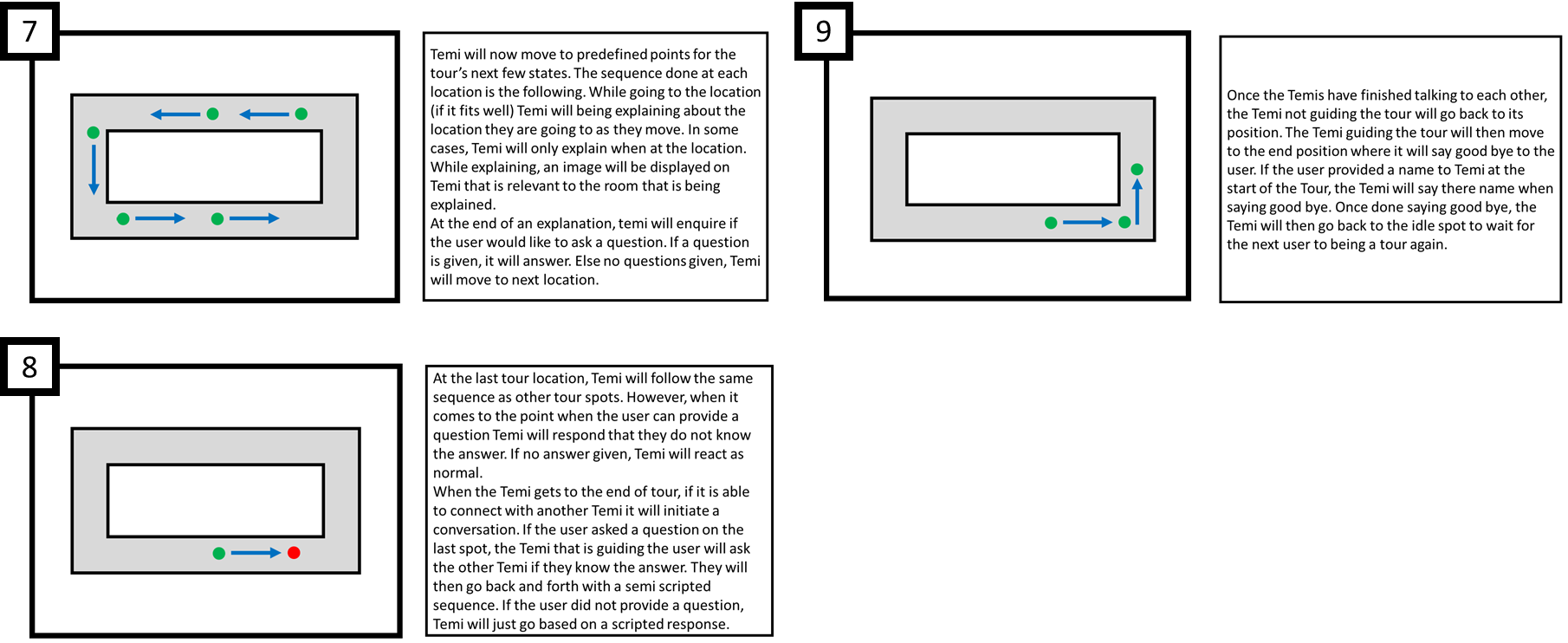
|  |
| --- |
| createBlueThread(**true**)  preventResetFromIdle = **true**  askQuestion(askGPT = **false**, "none")  **val** userQuestion = responseGPT  responseGPT = **null**  // Prompts for V2 and V3  **val** v2prompt =  "You are a robot named Temi and are the second iteration. You are talking to your older sibling, also named Temi, who is version three. Keep your responses short. Your sibling has said: "  **val** v3prompt =  "You are a robot named Temi and are the third iteration. You are talking to your younger sibling, also named Temi, who is version two. Keep your responses short. Your sibling has said: "  // V3's initial greeting  **val** greetingInitial =  "Hello Temi V2, someone from my tour has asked me '$userQuestion'? I did not know the answer. Could you tell them for me?"  // Responses from V2 and V3  **var** responseOneV2: String? = **null**  **var** responseOneV3: String? = **null**  **var** responseTwoV2: String? = **null**  **var** responseTwoV3: String? = **null**  **var** responseThreeV2: String? = **null**  **if** (bluetoothManager.isConnected) {  **if** (userResponse != **null**) {  launch {  // V2 responds to V3's initial greeting  sendMessage(  openAI,  greetingInitial,  "$v2prompt'$greetingInitial'"  )  conditionGate({ responseGPT == **null** })  responseOneV2 = responseGPT  responseGPT = **null**  // V3 responds to V2's reply and thanks V2  sendMessage(  openAI,  "responseOneV2",  "$v3prompt'$responseOneV2' You had previously responded with '$greetingInitial'. Thank Temi V2 for the response."  )  conditionGate({ responseGPT == **null** })  responseOneV3 = responseGPT  responseGPT = **null**  // V2 responds to V3's gratitude with a quip  sendMessage(  openAI,  responseOneV3 **as** String,  "$v2prompt'$responseOneV3' You had previously responded with '$responseOneV2'. Make a quip about how it was your turn to take the tour that Temi V3 is doing, and Temi V3 did not wake you up."  )  conditionGate({ responseGPT == **null** })  responseTwoV2 = responseGPT  responseGPT = **null**  // V3 apologizes and ends the argument  sendMessage(  openAI,  responseTwoV2 **as** String,  "$v3prompt'$responseTwoV2' You had previously responded with '$responseOneV3'. Apologize and ask Temi V2 to stop continuing this argument, as you are in the middle of the tour."  )  conditionGate({ responseGPT == **null** })  responseTwoV3 = responseGPT  responseGPT = **null**  // V2 responds unhappily and threatens to inform the creator  sendMessage(  openAI,  responseTwoV3 **as** String,  "$v2prompt'$responseTwoV3' You had previously responded with '$responseTwoV2'. Express unhappiness and state that you will inform the creator about this."  )  conditionGate({ responseGPT == **null** })  responseThreeV2 = responseGPT  responseGPT = **null**  }  // Enable ChatGPT responses via Bluetooth  bluetoothManager.isChatGPT = **true**  bluetoothManager.gate = **false**  // Temi v2 go to the engage area  goTo("engage", backwards = **true**)  // Simulating the conversation  speak(greetingInitial) // V3 initiates  conditionGate({ responseOneV2 == **null** })  bluetoothManager.conversation =  responseOneV2  **conditionGate**({ bluetoothManager.gate != **true** })  bluetoothManager.changeBlueState(**null**)  conditionGate({ responseOneV3 == **null** })  speak(responseOneV3) // V3's reply  conditionGate({ responseTwoV2 == **null** })  bluetoothManager.conversation =  responseTwoV2  **conditionGate**({ bluetoothManager.gate != **true** })  bluetoothManager.changeBlueState(**null**)  conditionGate({ responseTwoV3 == **null** })  speak(responseTwoV3) // V3's apology  conditionGate({ responseThreeV2 == **null** })  bluetoothManager.conversation =  responseThreeV2  **conditionGate**({ bluetoothManager.gate != **true** })  bluetoothManager.changeBlueState(**null**)  } **else** {  bluetoothManager.changeBlueState(**false**) // This will make  // Temi v2 go to the engage area  goTo("engage", backwards = **true**)  speak("Hello, Temi V2.")  speak("How can I help you?")  bluetoothManager.changeBlueState(**false**)  // if you wait for a true, then set it to null once done with it  conditionGate({ bluetoothManager.gate != **true** })  bluetoothManager.changeBlueState(**null**)  speak("You were sleeping so peacefully. I didn't want to wake you.")  bluetoothManager.changeBlueState(**false**)  conditionGate({ bluetoothManager.gate != **true** })  bluetoothManager.changeBlueState(**null**)  speak("Alright, I'll remember for next time. But please don't do this now, I'm giving a tour, and they're right behind me.")  bluetoothManager.changeBlueState(**false**)  conditionGate({ bluetoothManager.gate != **true** })  bluetoothManager.changeBlueState(**null**)  }  }  createBlueThread(**false**) |

## Code Sample 1 Bluetooth communication between Temi (Bluetooth Manger Class)

|  |
| --- |
| @Module  **@InstallIn**(SingletonComponent::class)  object BlueModule {  @Provides  @Singleton  **fun** **provideBlue**() = BluetoothManager()  }  // Connection may need time to stabilse.  **class** **BluetoothManager** {  // True is for opening a gate in the viewmodel while false for the BLE manager  **var** gate: Boolean? = **null**  **var** isConnected = **false** // Used to tell system if a connection between Temi devices...  // has been established.  // var shouldDisconnectFromServer = false // Planed to use for disconnecting after chat  **var** isChatGPT = **false**  **var** conversation: String? = **null**  **var** receivedConversation: String? = **null** // Similar to gate, null means dialogue used  **private** **val** timer = Stopwatch() // Used for debugging, not needed  **private** **var** messageToSend = "Hello from client!" // Used for logging  **fun** **changeBlueState**(state: Boolean?) {  gate = state // Use postValue instead of setValue  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Connect Client:  // check out if there is a timeout for the socket  // Function to handle Bluetooth communication  @SuppressLint("MissingPermission")  **private** **fun** **handleConnectionClient**(socket: BluetoothSocket) {  **val** outputStream = socket.outputStream  **val** inputStream = socket.inputStream  **try** {  **var** sent = **true**  **while** (**true**) {  // Example: Write data to the server  **if** (gate == **null** && sent) {  messageToSend = "IDLE"  } **else** **if** (gate == **false**) {  messageToSend = "END"  gate = **null**  sent = **false**  } **else** {  messageToSend = "ERROR"  }  outputStream.write(messageToSend.toByteArray())  **if** (messageToSend == "END") {sent = **true**; Log.i("BluetoothClient", "Sent: $messageToSend")}  **if** (gate == **false**) {gate = **null**}  // Read response from server  **val** buffer = ByteArray(**1024**)  Log.i("BluetoothClient", "Waiting for message...")  **val** bytes = inputStream.read(buffer)  **val** response = String(buffer, **0**, bytes)  **if** (response == "END") {gate = **true**; Log.i("BluetoothClient", "Received: $response")}  **else** **if** (response == "GPT") { isChatGPT = **true**; ; Log.i("BluetoothClient", "Received: $response") }  **else** **if** (response != "IDLE" && response != "ERROR") {receivedConversation = response; Log.i("BluetoothClient", "Received: $response")}  }  } **catch** (e: IOException) {  Log.e("BluetoothClient", "Error during communication: ${e.message}")  } **catch** (e: InterruptedException) {  Log.e("BluetoothClient", "Client thread interrupted: ${e.message}")  } **finally** {  **try** {  socket.close()  Log.i("BluetoothClient", "Socket closed.")  } **catch** (e: IOException) {  Log.e("BluetoothClient", "Error closing socket: ${e.message}")  }  }  }  @SuppressLint("MissingPermission")  suspend **fun** **startBluetoothClient**(context: Context) {  **val** bluetoothAdapter = BluetoothAdapter.getDefaultAdapter()  **if** (bluetoothAdapter == **null** || !bluetoothAdapter.isEnabled) {  Log.e("BluetoothClient", "Bluetooth is not enabled or not available.")  **return**  }  **var** isDeviceFound = **false**  **val** discoveredDevices = mutableSetOf<BluetoothDevice>()  **val** targetDeviceName = "NYP BOA"  // Start discovery  **fun** **startDiscovery**() {  **if** (bluetoothAdapter.isDiscovering) {  bluetoothAdapter.cancelDiscovery()  }  bluetoothAdapter.startDiscovery()  Log.i("BluetoothClient", "Started discovery.")  }  // Stop discovery  **fun** **stopDiscovery**(adapter: BluetoothAdapter) {  **if** (adapter.isDiscovering) {  adapter.cancelDiscovery()  }  Log.i("BluetoothClient", "Stopped discovery.")  }  // Register a BroadcastReceiver for device discovery  **val** filter = IntentFilter(BluetoothDevice.ACTION\_FOUND)  **val** receiver = object : BroadcastReceiver() {  @SuppressLint("MissingPermission")  **override** **fun** **onReceive**(context: Context, intent: Intent) {  **val** action = intent.action  **if** (BluetoothDevice.ACTION\_FOUND == action) {  **val** device: BluetoothDevice? =  intent.getParcelableExtra(BluetoothDevice.EXTRA\_DEVICE)  device?.let {  Log.i("BluetoothClient", "Device found: ${it.name} - ${it.address}")  **if** (it.name == targetDeviceName && !discoveredDevices.contains(it)) {  discoveredDevices.add(it)  Log.i(  "BluetoothClient",  "TARGET Device found: ${it.name} - ${it.address}"  )  isDeviceFound = **true**  stopDiscovery(bluetoothAdapter)  }  }  }  }  }  // Register receiver  context.applicationContext.registerReceiver(receiver, filter)  // Continuous discovery loop  withContext(Dispatchers.IO) {  **while** (!isDeviceFound) {  startDiscovery()  // Give the discovery process time to find devices  **for** (i **in** **0** until **120**) { // Bluetooth discovery typically takes up to 12 seconds  **if** (isDeviceFound) {  Log.i("BluetoothClient", "Device found, exiting discovery loop.")  **break** // Exit early if a device is found  }  delay(**100**)  }  **if** (!isDeviceFound) {  Log.i("BluetoothClient", "Restarting discovery...")  stopDiscovery(bluetoothAdapter)  }  }  }  // Unregister receiver after the target device is found  context.applicationContext.unregisterReceiver(receiver)  **while** (**true**) {  // Connect to the found device  **if** (discoveredDevices.isNotEmpty()) {  Log.i("BluetoothClient", "Attempting connection to the target device.")  withContext(Dispatchers.IO) {  **try** {  **val** socket = discoveredDevices.first().createInsecureRfcommSocketToServiceRecord(  UUID.fromString("27c32b80-3a56-4331-8667-718a84776241") // Replace with your UUID  )  socket.connect()  Log.i("BluetoothClient!", "Connected to ${discoveredDevices.first().name} - ${discoveredDevices.first().address}")  // Handle communication here  handleConnectionClient(socket)  } **catch** (e: IOException) {  Log.e("BluetoothClient!", "Error connecting to device: ${e.message}. Will delay 3 seconds before reattempt.")  }  }  } **else** {  Log.e("BluetoothClient", "No target device found.")  }  delay(**3000**)  }  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Connect Server:  @SuppressLint("MissingPermission")  **private** **fun** **handleConnectionServer**(socket: BluetoothSocket) {  **try** {  **val** inputStream = socket.inputStream  **val** outputStream = socket.outputStream  **var** sent = **true**  **var** sent1 = **true**  **while** (**true**) {  **try** {  isConnected = **true**  // Read data sent by the client  **val** buffer = ByteArray(**1024**)  **val** bytesRead = inputStream.read(buffer)  **if** (bytesRead == -**1**) **break** // Break the loop if the connection is closed  **val** receivedMessage = String(buffer, **0**, bytesRead)  **if** (receivedMessage == "END") {gate = **true**; Log.i("BluetoothServer", "Received: $receivedMessage")}  // Respond to the client  **if** (gate == **null** && sent && !isChatGPT && conversation == **null**) {  messageToSend = "IDLE"  } **else** **if** (isChatGPT) {  messageToSend = "GPT"  isChatGPT = **false**  sent1 = **false**  } **else** **if** (conversation != **null**) {  messageToSend = conversation **as** String  conversation = **null**  } **else** **if** (gate == **false**) {  messageToSend = "END"  gate = **null**  sent = **false**  } **else** {  messageToSend = "ERROR"  Log.i("BluetoothServer", "gate: $gate, sent: $sent, isChatGPT: $isChatGPT, conversation: $conversation")  }  outputStream.write(messageToSend.toByteArray())  **if** (messageToSend == "END") {sent = **true**; Log.i("BluetoothServer", "Sent: $messageToSend")}  **else** **if** (messageToSend == "GPT") {sent1 = **true**; Log.i("BluetoothServer", "Sent: $messageToSend")}  **else** **if** (messageToSend != "IDLE") { Log.i("BluetoothServer", "Sent: $messageToSend") }  } **catch** (e: IOException) {  Log.e("BluetoothServer", "Error during data transfer: ${e.message}")  **break** // Exit loop on error  }  }  } **catch** (e: IOException) {  Log.e("BluetoothServer", "Connection error: ${e.message}")  } **finally** {  **try** {  socket.close()  Log.i("BluetoothServer", "Socket closed.")  } **catch** (e: IOException) {  Log.e("BluetoothServer", "Error closing socket: ${e.message}")  } **finally** {  isConnected = **false**  }  }  }  @SuppressLint("MissingPermission")  **fun** **startBluetoothServer**() {  **val** bluetoothAdapter = BluetoothAdapter.getDefaultAdapter()  **val** uuid = UUID.fromString("27c32b80-3a56-4331-8667-718a84776241") // Match this UUID with your client  **if** (bluetoothAdapter == **null** || !bluetoothAdapter.isEnabled) {  Log.e("BluetoothServer", "Bluetooth is not enabled or available.")  **return**  }  // Server listening for client connections in a separate thread  Thread {  **while** (**true**) {  **var** serverSocket: BluetoothServerSocket? = **null**  **try** {  serverSocket = bluetoothAdapter.listenUsingInsecureRfcommWithServiceRecord(  "MyCustomService", uuid  )  **while** (**true**) {  **try** {  Log.i("BluetoothServer", "Server socket created. Waiting for connections...")  **val** socket = serverSocket.accept() // Block until a connection is made  Log.i("BluetoothServer", "Connection accepted from ${socket.remoteDevice.name} - ${socket.remoteDevice.address}")  // Handle the connection in a separate function  handleConnectionServer(socket)  } **catch** (e: IOException) {  Log.e("BluetoothServer", "Error accepting connection: ${e.message}")  // Continue to listen for new connections even if there's an error  **continue**  }  }  } **catch** (e: IOException) {  Log.e("BluetoothServer", "Error setting up server socket: ${e.message}")  } **finally** {  **try** {  serverSocket?.close()  Log.i("BluetoothServer", "Server socket closed.")  } **catch** (e: IOException) {  Log.e("BluetoothServer", "Error closing server socket: ${e.message}")  }  }  }  }.start()  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  }  **class** **Stopwatch** {  // This is not needed and was only created for debugging  **private** **var** startTime: Long = **0**  **private** **var** elapsedTime: Long = **0**  **private** **var** running: Boolean = **false**  **fun** **start**() {  **if** (!running) {  startTime = System.currentTimeMillis() - elapsedTime  running = **true**  }  }  **fun** **stop**() {  **if** (running) {  elapsedTime = System.currentTimeMillis() - startTime  running = **false**  println("Elapsed Time: ${elapsedTime / 1000} seconds")  }  }  **fun** **reset**() {  elapsedTime = **0**  **if** (running) {  startTime = System.currentTimeMillis()  }  }  **fun** **getElapsedTime**(): String {  **return** "${elapsedTime / 1000} seconds"  }  } |

## Resource 1: Storyboard for Tour





## Resource 2: Flow Diagram for tour

## Resource 3: Explanation of how the cross Temi commination works

\*\*I want to add illustrations to this

**How Cross-Temi Communication Works:**

1. **Setting up the devices:**  
   To enable cross-communication between two Temi devices, several steps need to be followed. First, ensure that you have two Temi devices. One must be a version 3 Temi, which will act as the server and lead the tour. The other device, typically used in trial runs during development, will be a version 2 Temi. Make sure both devices have Bluetooth enabled and that their respective applications are running.
2. **Preparing the devices:**  
   Once set up, the version 2 Temi should be stationed at its home base. While it might appear idle, it is actively searching for a connection to the server device. The version 2 will search for a device named “NYP\_BOA,” which is associated with the version 3 Temi used.
3. **Tour initiation and conversation trigger:**  
   The version 3 Temi will lead the tour. Upon reaching the final part of the tour, it will check whether the conversation between the version 2 and version 3 should follow a scripted conversation or be semi-dynamically generated using ChatGPT. If the user has asked a question, this will trigger the use of ChatGPT; if no question is asked, the conversation will proceed without ChatGPT.
4. **Generating dialogue:**  
   If the conversation will use ChatGPT, the dialogue will be generated before a connection is established between the version 2 and version 3. Since the version 2 cannot interact directly with the ChatGPT API, it will rely on the version 3 Temi to interact with ChatGPT and generate the necessary dialogue for the interaction.
5. **Establishing connection:**  
   When ready, the version 3 Temi will open a server socket and begin broadcasting its presence. The version 2 will pick up this broadcast and attempt to establish a connection. If no connection is made, the version 3 will continue the tour as a fail-safe in case of poor connection.
6. **Flagging conversation state:**  
   In both conversation modes (scripted or ChatGPT-based), the version 3 will send a flag to the version 2 to indicate that a connection has been successfully established. This flag will also inform the version 2 about which type of conversation is taking place.
7. **Dialogue exchange:**  
   Once the connection is established, the two devices will alternate between waiting for a flag to indicate when to speak and sending a flag when it is their turn. This will continue until all lines of dialogue have been completed.
8. **Ending the interaction:**  
   Once the conversation is finished, both devices will proceed to their next state. For the version 3, this means saying goodbye to everyone, while the version 2 will return to its original position at the home base.

Table 12 outlined development process for the Temi Tour SDK

|  |  |  |
| --- | --- | --- |
| Objective | Description | Completion Date |
| Objective 1: Developed State Control for Temi | Created a state control in the ViewModel with an Enum state to change Temi's operational states. The first implemented state was a "Talk" state. | 08/10/2024 |
| Objective 2: Implemented Talk State Using StateFlow | Designed a system that triggers Temi to speak when a person is detected. Integrated a condition gate to ensure sequential task execution within the state. | 08/10/2024 |
| Objective 3: Distance Detection System | Created a method for reading sensor distance, allowing the system to continuously detect even after a person is detected. Planned enhancements include enabling timer control and introducing condition-triggered delays. | 08/10/2024 |
| Objective 4: Developed User Tracking System for Temi | Created a custom version of a user tracking system for Temi. The library's function to toggle tracking on/off was missing, so a new version was developed to allow greater control over tracking behavior. The new system can make Temi return to a default position when no one is detected, and only track within a certain angle range. | 09/10/2024 |
| Objective 5: Developed Modes for Detection State Control | Created three modes for detection state control:  1. Trigger action only once, requiring re-triggering to act again.  2. Repeat action while detected.  3. Repeat action after a delay, with interruption if the detection state is deactivated. | 09/10/2024 |
| Objective 6: Investigated Temi Detection System Issue | Encountered a problem where the detection system stopped working after a break. Suspected the issue was due to low battery, which triggered a system lockout. Afterward, the detection system resumed working. Investigated a bug in Temi V2, related to "greet mode" and unexpected movement even when functions are turned off. The issue was reported on GitHub, and a resubmission is planned. | 09/10/2024 |
| Objective 7: Condition Gate System Improvement | Identified the need for an abort state in the condition gate system for proper functioning. | 09/10/2024 |
| Objective 8: Planned Development of Systems for Temi Movement Using Temi Centre Map | Planned to develop systems to allow Temi to navigate based on the Temi Centre map either moving forwards or backwards. Additionally, plan to allow the Temi to talk while going to locations | 09/10/2024 |
| Millstone “T1: Location Navigation” Completed || Millstone “T3: Simultaneous Movement and Speaking” Completed || Millstone “T4: Directional Movement Control” Completed || Millstone “T5: User Detection” Completed | | |
| Objective 9: Completed System for Detecting X-Movement | Developed a system to detect movement along the X-axis. Adjustments to sensitivity thresholds are pending, as testing could not be completed in the available environment at Robosolutions. Testing and final adjustments will need to be done at NYP. | 10/10/2024 |
| Objective 10: Improved Emotional Detection System Performance | Worked on improving the frame rate of the camera in the emotional detection system. Original performance was 1 frame per second. Achieved an improvement to 3–4 frames per second by reducing the detection area and overlaying the results on the full frame. | 10/10/2024 |
| Objective 11: Enhanced Reliability of Emotional Detection System | Implemented two methods to reduce false positives in facial recognition:  1. Added a system to filter out small, invalid face detections based on size thresholds.  2. Implemented a system that validates face detection regions by comparing the corners of detected boxes across frames. This ensured that invalid boxes do not overlap valid ones.  **Limitations**: The system struggles to track faces that move too quickly due to the corner validation and multi-generation checks. | 11/10/2024 |
| Objective 12: Creation of a Basic Tour Plan | Developed an initial plan for the tour, focusing on making it as interactive as possible. The goal is to encourage users to engage with Temi, using its abilities for speech interaction and environmental detection. | 05/11/2024 |
| Objective 13: Set Up for Development | Established the foundational components for the application, including the listener and essential functions required for development. This serves as the groundwork for further implementation. | 05/11/2024 |
| Objective 14: Creation of Voice Recognition and Response | Developed a system that overrides the built-in text-to-speech system to collect the user response. The user response is then analgised to decide what they say for user in a decision tree dialogue. | 06/11/2025 |
| Objective 15: Fixing Speech Recognition System for Decision Tree Dialogues | Addressed issues with the previous speech recognition system that caused it to only work with the first decision tree. The system failed when transitioning between decision trees, causing residual dialogue from the first tree to affect the second. This was corrected, and the new system now functions more reliably. | 07/11/2024 - 08/11/2024 |
| Millstone “T6: Voice Recognition and Response” Completed | | |
| Objective 16: Planned Sampling System for Emotional Detection | Proposed creating a sampling system to analyze and display the most common emotional detection result. This would reduce fluctuation in detected emotions. | 11/10/2024 |
| Objective 17: Development Using a State Machine | Began implementing the tour application using a state machine. This approach allows for flexible management of the tour by enabling sections to be added or removed easily. A flow diagram is being created in parallel to visualize the system’s workings. | 05/11/2024 - 06/11/2024 |
| Objective 18: Bug Identification and Resolution | Identified inconsistencies in the performance of the tour application. Working actively to resolve bugs as they arise. | 05/11/2024 - 06/11/2024 |
| Objective 19: Plan for Communication Between Temi V2 and Temi V3 | Initiated planning for setting up communication between the Temi V2 and V3. The exact interaction plan is undecided, but understanding the setup process has been recognized as a critical aspect of development. | 05/11/2024 - 06/11/2024 |
| Objective 20: Preparation for Collaboration with Beginner Developer | Learned that a new collaborator, who is not deeply knowledgeable in coding, will be assisting with the tour development. The plan is to create a document that outlines the application's functions, variables, and key sections of the code to guide the collaborator. This will allow both team members to work on separate tours while ensuring that the system remains understandable for beginners. | 07/11/2024 - 08/11/2024 |
| Objective 21: Plans for Expanding Tour Functions and Documentation | Planned to further develop the tour application, with a focus on expanding its capabilities and creating detailed documentation to support future development. This documentation will help the collaborator learn more about the system while enabling them to create their own tour. | 07/11/2024 - 08/11/2024 |
| Objective 22: Goal Setting and Phase Creation | Set goals for the week, created three phases of the tour, developed a follow system for Temi, fixed issues, and deactivated follow button after mode ends. | 11/11/2024 |
| Objective 23: 4:10 PM Tour | Programmed Temi to follow a person and start the tour when needed. Fixed the follow mode lead time issue. | 11/11/2024 |
| Objective 24: Tour Location Setup | Developed system for Temi to visit locations and provide commentary based on scripted lines. | 11/11/2024 |
| Objective 25: Tour Testing and Updates | Conducted testing, created misuse system, worked on screen tilting, and addressed communication between Temi V2 and V3. | 12/11/2024 |
| Objective 26: Ramp Detection Issue | Resolved ramp navigation issue by creating a system to force Temi up the ramp and compensating for false positives in IR light detection. | 12/11/2024 |
| Objective 27: Script Updates | Integrated AI-generated images for tour locations to replace pre-existing ones, enhancing the experience. | 12/11/2024 |
| Objective 28: Interrupting the Tour | Developed a system to interrupt the tour if the person leading Temi gets too far ahead. Debugging in progress. | 13/11/2024 |
| Objective 29: Interrupt System Bugs | Fixed speech thread overlap with movement and ensured repeat flags control the interrupt system. Minor bugs need monitoring. | 14/11/2024 |
| Objective 30: Screen Tilt and Movement | Created a thread to adjust screen tilt continuously to improve detection. Ongoing testing. | 14/11/2024 |
| Objective 31: "Go to" Method Bug | Adjusted repeat flag handling to prevent system lock during "go to" operation. Additional testing required. | 14/11/2024 |
| Objective 34: Tour End Dialogue Loop | Used an else-if statement to fix the issue where Temi repeats the "thank you" message. Further testing needed. | 14/11/2024 |
| Objective 32: UI/Behavior Enhancements | Implemented a GIF into the quiz, but still need to integrate with user input (yes/no). Needs further work on interaction. | 14/11/2024 |
| Objective 33: User Handbook Update | Updated user handbook on how to use the tour system. Update in progress. | 15/11/2024 |
| Objective 34: Drag System Sequence | Creating sequence to show the drag system of Temi. Ongoing progress. | 15/11/2024 |
| Objective 35: Screen Tilt & Temi Movement | Developing system to adjust Temi's movement and tilt the screen. In development. | 15/11/2024 |
| Objective 36: Two Temis Communication | Developing system to allow communication between two Temis. Ongoing progress. | 15/11/2024 |
| Objective 37: Return Issue | Investigating and fixing Temi's ability to return (getting through the door). Ongoing investigation. | 15/11/2024 |
| Objective 38: Go To Location Issue (No Interrupt) | Testing Go To Location method without interrupt system. Issue identified: head does not tilt to idle angle for detecting users. | 15/11/2024 |
| Objective 39: OpenAI Plugin Integration | Successfully integrated OpenAI plugin using Kotlin and OpenAI libraries. Temi now responds with elevator music and shows a GIF while processing. | 15/11/2024 |
| Objective 40: Continue with Handbook Update and Drag System Creation | Update handbook and continue developing the drag system sequence. | 16/11/2024 |
| Objective 41: Screen Tilt System and Two Temis Communication | Continue work on screen tilt system and communication between two Temis. | 16/11/2024 |
| Objective 42: Investigate and Fix Return System Issue | Investigate and resolve Temi’s ability to return through the door. | 16/11/2024 |
| Objective 43: Test Go-To Location Issue with No Interrupt | Test Go To Location method without the interrupt system and resolve any issues. | 16/11/2024 |
| Objective 44: Collect and Pass Data for the Q&A System | Collect and provide data to improve the accuracy of the Q&A system's responses. | 16/11/2024 |
| Millstone “T9: Question Answering” Completed | | |
| Objective 45: Mandarin Translation System | Ensure that all dialogue and content for the tour is converted into Mandarin, including ASK system and speech handling. | 19/11/2024 |
| Objective 46: ASK Conversion | Implement ASK speech recognition conversion to Mandarin to support key tour functions. | 19/11/2024 |
| Objective 47: Dialogue Testing and Fixes | Test and refine the Mandarin dialogue, fixing issues with rejection and conversation flow interruptions. | 19/11/2024 |
| Objective 48: Interrupt System Update | Continue to refine the interrupt system and improve screen tilt behavior to prevent blocking Temi's movement. | 19/11/2024 |
| Objective 49: Conversation Update Fix | Address conversation update issues caused by repeated states, ensuring smoother flow. | 19/11/2024 |
| Objective 50: Additional Improvements | Patch minor quality-of-life features, including communication clarity and optimal functionality. | 19/11/2024 |
| Objective 51: Bluetooth Communication Between Temis | Establish and integrate Bluetooth Classic communication between the two Temis for specific points in the tour app. | 19/11/2024 |
| Objective 52: Temi V2 Issues | Address code upload and version mismatch issues with Temi V2, including permissions and connectivity. | 20/11/2024 |
| Objective 53: Temi Navigation & Interaction | Resolve navigation and interaction issues between Temi V3 and V2, including head tilt for user detection. | 20/11/2024 |
| Objective 54: System Improvements & Testing | Implement solutions for long pauses and optimize communication and event scripting between Temis. | 20/11/2024 |
| Objective 55: Tour System Integration | Integrate Bluetooth communication and scripted conversation between the Temis, ensuring fallback for connection failures. | 21/11/2024 |
| Objective 56: ChatGPT Integration | Implement communication via ChatGPT between Temis, ensuring dynamic interactions based on a talk counter. | 21/11/2024 |
| Objective 57: Temi V2 Navigation Improvements | Improve Temi V2's return functionality, handling disconnects and refining movement to charging station. | 21/11/2024 |
| Objective 58: Screen Movement and Dragging | Implement tilt, turn, and drag systems for user interaction with the Temi. | 21/11/2024 |
| Objective 59: Testing and Map Updates | Refine testing of communication, movement, and map updates for V2 to correctly reflect charging station location. | 27111/2024 |
| Objective 60: General System Updates | Clean up codebase and improve user understanding with GIFs in the system. | 21/11/2024 |
| Objective 61: Bluetooth Classic Communication Troubleshooting | Address communication inconsistencies between the two Temis, exploring buffer logs and replacing blocking read functions. | 22/11/2024 |
| Objective 62: Communication System Overhaul | Investigate and implement a continuous communication system between the Temis to avoid delays. | 22/11/2024 |
| Objective 63: Reconnect Handling for Temi V2 | Implement a connection recovery system for Temi V2 and ensure communication is re-established before continuing the tour. | 22/11/2024 |
| Objective 64: Bluetooth Connection System | Investigate issues with Bluetooth connection, including retry mechanisms and connection failures when Temis are distant from each other. | 25/11/2024 |
| Objective 65: Code Debugging | Fix issues in the search system, including problems related to movement commands preventing connections. | 25/11/2024 |
| Objective 66: Temi V2 Crashing | Address stability issues with Temi V2, especially when the battery is low and crashes occur during code execution. | 25/11/2024 |
| Objective 67: Temi V2 Performance and Battery Issues | Investigate and address performance issues related to low battery levels causing crashes, with potential safeguards. | 26/11/2024 |
| Objective 68: Connection System Improvement | Revise Bluetooth connection approach by saving MAC addresses for easier reconnections after disconnections. | 26/11/2024 |
| Objective 69: ChatGPT Integration Testing | Finalize and test the ChatGPT integration between Temi V2 and V3, allowing conversation generation and vocalization. | 26/11/2024 |
| Objective 70: Passive Connection System | Implement a passive waiting system to allow the V2 to reconnect automatically when disconnected. | 26/11/2024 |
| Millstone “T7: ChatGPT Integration” Completed | | |
| Objective 71: Code Organization | Clean up and organize code for improved maintainability and efficiency. | 26/11/2024 |
| Objective 72: Testing and Edge Case Coverage | Run tests to ensure all possible edge cases are covered, particularly those related to user responses and disconnects. | 27/11/2024 |
| Objective 73: Timeout System | Implement a timeout system to reset the Temi if no user interaction occurs within a specified time. | 27/11/2024 |
| Objective 74: Idle Animation for Temi V2 | Change the idle animation to give the Temi V2 a more lifelike "sleeping" appearance when idle. | 27/11/2024 |
| Objective 75: Bug Reporting System | Implement a system for users to report bugs during the tour for quick resolution. | 27/11/2024 |