**Project Description:**

Farely is a mobile fare-tracking application developed to help students and workers manage their daily transportation expenses. Created by Rowlet, the app responds to the common challenge of tracking fares manually or relying on memory, which often leads to poor budgeting. Farely simplifies fare monitoring by offering users tools to record commutes, set weekly or monthly budgets, and review summarized expenses. Farely encourages more mindful and financially responsible commuting habits.

**Requirements Summary:**

|  |  |  |
| --- | --- | --- |
| **MINIMUM REQUIREMENTS** |  | ANDROID |
| Processor Cores | Dual Core |
| OS | Android 8.0 (Oreo) or higher |
| RAM | 2 GB |
|  | IOS |
| Device | iPhone 6s or newer |
| OS | iOS 12 |
| RAM | 2 GB |
| **RECOMMENDED REQUIREMENTS** |  | ANDROID |
| Processor Cores | Octa Core |
|  | OS | Android 11 (R) or higher |
| RAM | 4 GB or more |
|  | IOS |
| Device | iPhone 8 or newer |
| OS | iOS 14 or higher |
| RAM | 3 GB or higher |
| **OTHER REQUIREMENTS** | Internet | Wi-Fi or Mobile Data |

**Table 1. System Requirements**

Farely is designed to be lightweight and accessible across a wide range of devices, including both Android and iOS platforms. To maintain performance, stability, and security, the app follows updated minimum specifications while remaining compatible with most mid- to low-range smartphones.

Farely works on both Android and iOS platforms, supporting devices running Android 8.0 (Oreo) and above, or iOS 13 and above. These baselines ensure compatibility with current APIs, user interface components, and data security protocols.

**Prototype Description:**

The prototype for Farely was built using Figma, a cloud-based design and prototyping platform. Figma allows interactive previews and real-time collaboration.

**FARELY Figma Link**:

https://www.figma.com/proto/kK4TJnIJwMULzID1ParEtC/Untitled?node-id=19-741&p=f&t=w3QmpfxGiSt1fkKi-1&scaling=scale-down&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=19%3A741

**User Scenario:**

Alyssa, a 20-year-old college student, commutes daily using a mix of jeepneys and buses. With a tight weekly allowance, she often finds herself short on fare by the end of the week, unsure where her money went. After discovering Farely, she starts logging her daily fares and sets a weekly budget. By using the summary view, she notices she's spending more on tricycle rides than she thought. Adjusting her routine, Alyssa switches to more cost-efficient transport and stays within budget.

**Farely Mock-up/Prototype**:

Screens screenshot of a login screen

Description automatically generated

SUMMARY, SET BUDGET AND INPUT FARE PAGE

**Summary -** Users can check their overall fare summary and total amount spent for a specific timeline. Records can be sorted via date, name (a-z), transportation type. Aside from the sort dropdown option, users can also manually input the date for a faster and specific search option. Users can also edit existing records via the edit button, and they will be routed back to the input fare page where all existing records will be automated in the blanks.

**Set budget -** Users can set their fare budget weekly or monthly. They can also view their budget history records.

**Input Fare –** Users can input their new fare entry and categorize it via transportation type, fare cost, payment method used, date and time. Specific inputs (ex. Integers for fare cost) in some fields will be applied to ensure proper handling.

**Budget History** – users can check their budget history, and edit their current budget set. It will state if their current budget is set to weekly or monthly. They will be routed to the set budget page if they click the edit button next to current budget.

**Edit Fare** – users can edit existing fare records from the summary page to fix inaccuracies.

A screenshot of a phone

Description automatically generatedScreens screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Screens screenshot of a phone

Description automatically generated

**Rationale:**

The team has chosen Figma as the primary tool for creating Farely’s prototype due to its accessibility, real-time collaboration features, and ease of use. Regardless of individual schedules and responsibilities, such as part-time jobs or other academic workloads, the team was able to collaborate on the project.

While Figma is convenient, it does have limitations. It requires an internet connection to save progress, which may disrupt workflow during connectivity issues.

**Changes to the Requirements:**

No changes have been made to the core system requirements. However, the usability criteria were revised to focus more on user ease and interaction, especially considering the mobile context of Farely. The team aligned the evaluation with Jakob Nielsen’s 10 Usability Heuristics, focusing on key principles such as:

* Minimalist and clean design
* Recognition over recall
* Consistency and standards

The emphasis was placed on creating a simple, intuitive offline interface to ensure that first-time users won’t feel overwhelmed.

**Initial Evaluation Plan:**

With some members balancing work and multiple academic responsibilities, in-person testing wasn’t feasible. As an alternative, the evaluation will be conducted remotely using platforms such as Discord, allowing the developers to observe participants in real time as they interact with the prototype.

The evaluation will consist of two components:

1. **Usability Specifications**
2. **Heuristic Evaluation**

**Usability Specifications**

The prototype aims to meet the following usability metrics:

* Effectiveness: Measures how well users can complete tasks, such as entering a fare or setting a budget.
* Efficiency: Evaluates how easily and quickly users can interact with the app.
* Utility: Assesses whether the prototype provides necessary and useful features.
* Learnability: Determines how fast new users can learn the app’s features.
* Memorability: Measures how well users remember how to use the app after a period of non-use.

User Testing Population

Around 5 students will participate in user testing. They will be asked to perform essential tasks that reflect the core functionalities of Farely. For example, they may be tasked with entering a new fare entry, setting a weekly budget, or editing a previously logged expense.

**Prototype Tasks**

Tasks are divided into logical sections based on Farely’s main features:

Main Menu Tasks

* Launch and exit the prototype
* Navigate between pages (e.g., Input Fare, Set Budget, Summary)

Input Fare Tasks

* Enter new fare entries
* Edit or delete a specific entry

Budget & Summary Tasks

* Set or adjust weekly/monthly budgets
* View total spending and remaining budget

These tasks were chosen to test whether Farely supports easy navigation, intuitive use, and all basic CRUD (Create, Read, Update, Delete) operations essential to fare tracking.

Roles

|  |  |
| --- | --- |
| **Developer / UI Designer Member** | **Task(s)** |
| Frances Jennifer A. Piccio | Will be recording time users interact with a task section, taking notes of the user’s experience, and relay the task that the participant will do. |
| Lythe Marvin Lacre | Will be recording time users interact with a task section, taking notes of the user’s experience, and relay the task that the participant will do. |
| Dwayne Basadre | Will be recording time users interact with a task section, taking notes of the user’s experience, and relay the task that the participant will do. |

Table 2. Team Member Tasks

|  |  |  |  |
| --- | --- | --- | --- |
| Main Menu | Within 1 minute or Below | Highly Acceptable | Successful |
| Above 1 minute | Not Acceptable | Unsuccessful |
| Input Fare | Within 5 minutes or Below | Highly Acceptable | Successful |
| Above 5 minutes | Not Acceptable | Unsuccessful |
| Budget &  Summary | Within 5 minutes or Below | Highly Acceptable | Successful |
| Above 5 minutes | Not Acceptable | Unsuccessful |

Table 3. Time Interpretation

Table 3 represents the interpretation above represents how the team will be interpreting the time spent with each participant in their tasks. The table will be used as a guideline to interpret if the design of given task is successful or not at a given task.

**Heuristic Evaluation**

Evaluation of Farely will also use the 10 Usability Heuristic method of Evaluation.

*Visibility of System Status*

The Prototype will keep the participants informed on what is happening in the Prototype.

# Match Between System and Real World

The prototype speaks the user’s language, with familiar words, phrases, and concepts rather than system-oriented terms.

# User control and Freedom

The Prototype offers to deal possible mistakes through confirmation modals for every action.

# Consistency and Standards

Users will not have to worry whether different words, situations, or actions mean the same thing.

# Error Prevention

Error Messages are Carefully designed which prevents a problem from occurring in the first place.

# Recognition rather than recall

Make objects, actions, and options visible. The user does not have to remember information from one part of the dialogue to another.

# Flexibility and Efficiency of Use

The prototype caters to both experienced and inexperienced users. Users readily tailor frequent actions

# Aesthetic and Minimalist Design

The prototype does not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

# Help Users Recognize, Diagnose, and Recover from Errors

Error messages are explained in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

# Help and Documentation

User can easily find help and documentation when need to interact with the prototype. This information is easy to search for.