# Software Requirements Specification for 4G06 - Software Engineering: Housemates App

Team #9, Housemates
Justin Dang - dangj15
Harris Hamid - hamidh1
Fady Marcos - morocof2
Rizwan Ahsan - ahsanm7
Sheikh Afsar - afsars

October 6, 2023

# Contents

# Revision History

| Date | Version | Notes |
|---|---|---|
| October 6th 2023 | 0 | First Revision |

# 1 Purpose of the Project

## 1.1 User Business

With the ongoing affordable housing shortage in Canada many people have been forced to find roommates in order to have a place to live in. This is especially common at universities like McMaster where its extremely common to have housemates while in student housing. While having roommates may help ease financial pressures it can lead to a lot of stress in dealing with them. These stresses can include things like dealing with splitting household tasks and grocery costs. An application that helps deal with these common stresses in the roommate life would make it more convenient for the housemates to live together and overall simplify their lives.

## 1.2 Goals of the Project

| Goals | Importance |
|---|---|
| The application will have a straightforward and user-friendly interface that is simple to use for all users, regardless of technical ability. | This allows first time users to be interested in our product and a good experience with the overall product. |
| The application will simplify household task management through a task management system. | This allows streamlining the allocation of chores which in return will reduce conflicts and misunderstandings between housemates. |
| The application will streamline expense sharing through a cost management system. | This makes it possible for roommates to monitor their expenditure and prevent overspending. Additionally, it encourages each person to make a fair financial contribution. |
| The application will have a scheduling system that will allow for users to schedule quiet hours | This allows users to focus on their work, sleep, or studies without any interruption. |
| The application will have a calendar to see scheduled events | This allows users to coordinate their schedules and help them in managing their time in a better way. |

# 2 Stakeholders

## 2.1 Client

N/A

## 2.2 Customer

- People with housemates: People with housemates are the primary stakeholders of this application. They can use the application to better simplify life with housemates. As such they will have the greatest influence out of the stakeholders on the requirements of the application during the development process.

## 2.3 Other Stakeholders

- Landlords / Property Manager / Housing Association: Landlords would be a secondary stakeholder for the application. Landlords might be interested in using an application like this for their tenants so that they will better be able to communicate with them with respects to household tasks that are required. As such, they might play a minor role on determining the requirements of the application during the design process.
- Families: Families can also benefit from the app to have a centralized place for all household matters. They can distribute chores evenly and keep track of bills. It promotes talking openly and encourages users to be more responsible about household duties and bills.

## 2.4 Hands-On Users of the Project

- People with housemates
- Landlords / Property Manager / Housing Association
- Families

## 2.5 Personas

Andrew is a student at McMaster University. He lives off-campus in student housing with four other housemates. He and his roommates have to buy groceries and supplies for the house. They also have to do various household chores and pay for utilities and internet. Sometimes Andrew and his roommates have trouble communicating with one another on this. Additionally, Andrew gets annoyed when his roommates decide to have a party the day before Andrew has an exam.

### 2.6 Priorities Assigned to Users

The primary stakeholders for this project are people with housemates and as such they will have the highest priority during the development process. The other stakeholders will be considered during the development process, but overall will have a lower priority.

### 2.7 User Participation

During the development process we plan to test the application with the stakeholders described above. Their feedback will help improve the application to better fit their needs.

### 2.8 Maintenance Users and Service Technicians

During the development of this project the project will be maintained by the development team.

## 3 Mandated Constraints

### 3.1 Solution Constraints

N/A

### 3.2 Implementation Environment of the Current System

The expected environment for this application will be mobile devices (Android, iOS), providing a user-friendly and intuitive application that is really simple to use. With goals of expanding to web application, so that we can cater to a much diverse user base using computers with Windows, Max or Linux and web browsers such as Chrome, Edge, Firefox, etc. Users would then have the convenience to access the solution/product on any device they want that suits their needs.

### 3.3 Partner or Collaborative Applications

N/A

### 3.4 Off-the-Shelf Software

N/A

### 3.5 Anticipated Workplace Environment

The anticipated workplace environment for this application is in users' homes since this application deals with roommates. However, as this app is on mobile devices there could be a wide variety of places where this application could be used.

### 3.6 Schedule Constraints

This project is being developed from September 2023 - April 2024.

### 3.7 Budget Constraints

This project has a budget constraint of $750.

### 3.8 Enterprise Constraints

N/A

## 4 Naming Conventions and Terminology

### 4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project

- Android: A mobile operating system designed for mobile devices.
- Google Play Store: The most common app store on Android.

## 5 Assumptions

- The users of this application are expected to have a basic understanding of how to use mobile devices with a touchscreen.

# 6 Scope

## 6.1 Product Context

The following diagram depicts the main application interacting with external systems and users and the relation between them in terms of data flow.

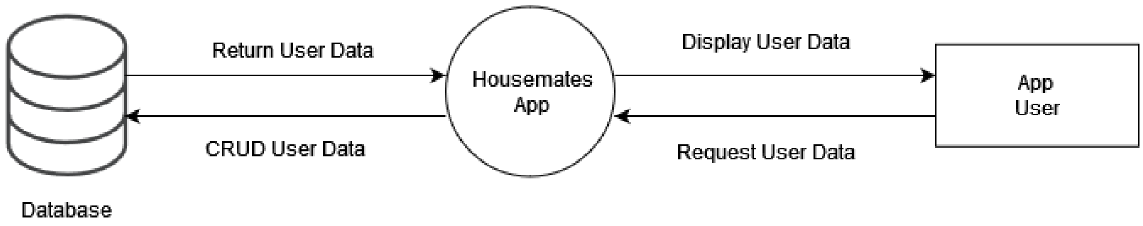

Figure 1: Context Diagram

## 6.2 Product Use Case Diagram

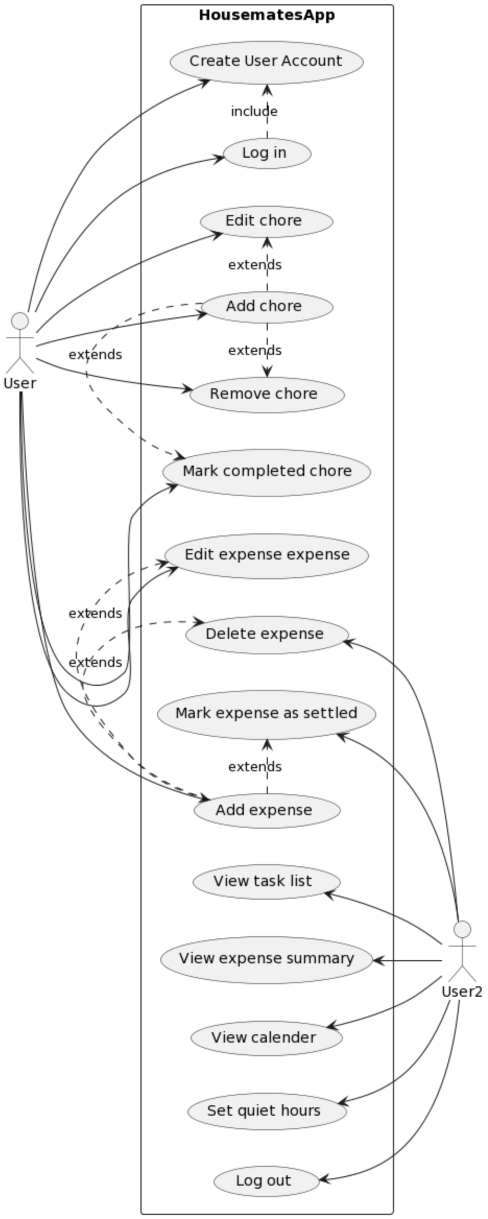

Figure 2: Usecase Diagram

## 6.3 Individual Product Use Cases (PUC's)

PUC1:

- ***Title*: Create user account**
- *Precondition*: User does not have existing account.
- *Trigger*: User selects option to create new account.
- *Description*:Allows user to provide necessary information (such as name, email, password) to create a new account within the system.

PUC2:

- ***Title*: Log in**
- *Precondition*: User has an existing account.
- *Trigger*: User enters their information and selects the log in option.
- *Description*: Allows user to authenticate and access their existing account.

PUC3:

- ***Title*: Add household chores**
- *Precondition*: User is logged in and has housemates added.
- *Trigger*: User selects the option to add a new household chore.
- *Description*: Enables users to create a new household chore with details like name, description, frequency and assign it.

PUC4:

- ***Title*: Edit household chores**
- *Precondition*: User is logged in and has existing household chore.
- *Trigger*: User selects the option to edit an existing household chore.
- *Description*: Allows users to modify details of an existing household chore, such as its name, description, frequency and assignee.

PUC5:

- ***Title*: Remove household chores**
- *Precondition*: User is logged in and has existing household chore.
- *Trigger*: User selects the option to delete an existing household chore.
- *Description*: Allows users to remove a household chore that is no longer required.

PUC6:

- ***Title*: Marking completion of household chores**
- *Precondition*: User is logged in and has existing household chore.
- *Trigger*: User selects the option to mark a household chore as completed.
- *Description*: Allows users to mark off a specific household chore and track that it has been successfully completed.

PUC7:

- ***Title*: Add expense**
- *Precondition*: User is logged in.
- *Trigger*: User selects the option to add a new household expense.
- *Description*: Enables users to record a new expense between all or specific housemates, including details like item name, cost, date of purchase, etc.

PUC8:

- ***Title*: Edit expense**
- *Precondition*: User is logged in and has existing existing expense.
- *Trigger*: User selects the option to edit an existing expense.
- *Description*: Allows users to modify details of an existing expense.

PUC9:

- ***Title*: Delete expense**
- *Precondition*: User is logged in and has existing expense.
- *Trigger*: User selects the option to delete an existing expense.
- *Description*: Enables users to remove an expense that is no longer relevant or was added by mistake.

PUC10:

- ***Title*: Mark expense as settled**
- *Precondition*: User is logged in and has existing household chore.
- *Trigger*: User selects the option to mark a household chore as completed.
- *Description*: Allows users to mark off a specific household chore and track that it has been successfully completed.

PUC11:

- ***Title*: View task list**
- *Precondition*: User is logged in and has existing tasks/chores.

- *Trigger*: User selects the option to view the task list.
- *Description*: Displays a list of all current tasks for the user's household.

PUC12:
- ***Title*: View expense summary**
- *Precondition*: User is logged in and has existing expenses.
- *Trigger*: User selects the option to view the expense summary.
- *Description*: Displays an overview of shared expenses and individual contributions.

PUC13:
- ***Title*: View calendar**
- *Precondition*: User is logged in.
- *Trigger*: User selects the option to view the calendar.
- *Description*: Displays a calendar with scheduled events, tasks, and quiet hours.

PUC14:
- ***Title*: Set quiet hours**
- *Precondition*: User is logged in and has housemates added.
- *Trigger*: User selects the option to set quiet hours.
- *Description*: Allows users to define specific time periods for quiet hours in the living space.

PUC15:
- ***Title*: Log out**
- *Precondition*: User is logged.
- *Trigger*: User selects the option to log out.
- *Description*: Allows a logged-in user to securely log out of their account, ending their current session.

# 7 Functional Requirements

## 7.1 Task Management System

TM1:
- *Description*: Housemates app will have a tile like interface where user can select tasks icon as one of the options to then navigate to the task management page

- *Rationale*: This is to be able to navigate to the task management page of the app instead of utilizing one of the other features
- *Fit Criterion*: User can tap on the tasks icon and reach the page for task management

TM2:

- *Description*: The task management page will have a list-like interface presenting to the user their chores that they need to completed highlighted amongst other roommates tasks.
- *Rationale*: This is to be able to clearly view the tasks which the user must complete
- *Fit Criterion*: User can clearly understand which chores they need to complete

TM3:

- *Description*: In the task management page, users are able to create new tasks/chores and add them to the task management page.
- *Rationale*: Users need to be able to add new chores/tasks as needed.
- *Fit Criterion*:
  1. Users are able to access a "Create New Task" button on the task management page.
  2. A task creation form is shown after the user presses on this button.
  3. The form fields have a spot for task name, due date, and roommate assignment (if applicable).
  4. After the user filling out the form and presses the "Create," button the new task now appears in the list of tasks.

TM4:

- *Description*: In the task management page, users can assign tasks to roommates if the chore is shared.
- *Rationale*: This feature enables users to divide tasks among roommates.
- *Fit Criterion*:
  1. Users can select a task from the list by pressing on the task.
  2. Users can assign the task to other roommates from their list of roommates.

TM5:
- *Description*: In the task management page, users can mark tasks as completed.
- *Rationale*: Users need to be allowed to track the progress of tasks and mark them as completed when they are done.
- *Fit Criterion*:
  1. Users is able to select a task from the list.
  2. Users is able to mark the task as completed.
  3. The user can see that the task is marked as completed on the task management page.

## 7.2 Bill Management System

BM1:
- *Description*: Users are able to create a new bill.
- *Rationale*: This requirement allows users to create new bills as needed.
- *Fit Criterion*: Users can add new bills by providing bill name, amount paid, category, due date, etc

BM2:
- *Description*: Users are able to assign bill to a housemate and split it between them.
- *Rationale*: This functionality will allow users to keep track of bill splitting easily.
- *Fit Criterion*:
  1. The bill amount is correctly split between the housemates.
  2. The total bill amount is shown on the screen and the amount required to pay by each housemate is displayed correctly.

BM3:
- *Description*: Users are able to modify already inputted bill details
- *Rationale*: This requirement allows users to modify any information on the bill for any correction or changes.
- *Fit Criterion*: Users can view and edit any bill on the page and the changes are saved and reflected correctly on the system.

BM4:
- *Description*: Users are able to categorize the bills.

- *Rationale*: This requirements makes it easier for users to differentiate and organize their expenses.
- *Fit Criterion*: Users can assign bills to any specific categories such as utility, rent, groceries, etc

BM5:
- *Description*: Users are able to keep track of which bills have been paid off.
- *Rationale*: This requirement will allow users to keep track of their payments.
- *Fit Criterion*:
  1. Users can mark any bill they paid off and the system keeps a record of the history.
  2. When a bill is paid off, the included housemates are provided with a notification.

BM6:
- *Description*: Users are able to attach bill receipts or relevant information as pictures.
- *Rationale*: This functionality will allow users to share any receipts or relevant document if they see fit.
- *Fit Criterion*: Users are able to upload files and images to the specific bill and the app updates it properly.

BM7:
- *Description*: Users are able to search for any specific bill.
- *Rationale*: This will allow users to navigate and search for any specific bill allowing them to save time.
- *Fit Criterion*: Users are able to search bills by their name, category, etc

## 7.3 Scheduling System

SS1:
- *Description*: Roommates are able to navigate to the scheduling page to use the scheduling feature by tapping the scheduling icon from the home interface to get to the scheduling page.
- *Rationale*: This functionality allows the app to be divided into sections depending on feature for clear usability

- *Fit Criterion*: Users can clearly distinguish from which section they want to utilize.

SS2:
- *Description*: Roommates are able to create new events and schedule them within the scheduling feature.
- *Rationale*: This functionality allows roommates to plan and coordinate activities efficiently.
- *Fit Criterion*:
  1. Users can tap on "Create New Event" within the Housemates app on the scheduling page.
  2. Users are able fill out an event creation form with details such as event name, date, time, duration, and a brief description.
  3. After creation, the new event is listed in the scheduling feature and is able to be seen be all the users/roommates.

SS3:
- *Description*: Users can view all the events scheduled by other roommates on their calendar within the scheduling feature.
- *Rationale*: Event visibility ensures that all roommates are aware of shared activities and can plan accordingly.
- *Fit Criterion*: Users can clearly see and distinguish between their personal events and shared roommate events on the calendar.

## 7.4 Account System

AS1:
- *Description*: Users can create new account by providing their personal information.
- *Rationale*: This requirement is essential to allow new users to use the product and personalize their experience within the app.
- *Fit Criterion*: The system should allow successful registration of users when they provide valid information

AS2:
- *Description*: Users are able to log in to the system by providing their email address/username and password.
- *Rationale*: This requirement ensures that only authorized users are able to access their accounts and the features available in the app.

- *Fit Criterion*:
  1. Users will be able to successfully log in to the app provided they give valid credentials
  2. Users will be unable to log in if they provide invalid credentials and appropriate error message will be shown.

AS3:
- *Description*: Users are able to update their profile information.
- *Rationale*: This requirement will allow user to change their information when they see fit.
- *Fit Criterion*: Changes made in the profile section should be updated and reflected properly in the app.

AS4:
- *Description*: Users are able to remove their account from the system.
- *Rationale*: Users should be given the option to have control of their own account.
- *Fit Criterion*:
  1. Users can temporarily deactivate their account and reactivate them.
  2. Users can permanently delete their account.

AS5:
- *Description*: Users are able to recover their account in case of forgetting their log in information
- *Rationale*: This requirement allows users a way to recover their account and seek support.
- *Fit Criterion*: Users are able to click a "Forgot Password" button if they forgot their password which sends an email to their provided email address.

## 7.5 Formal Specification of Cost Management System

A critical aspect of this project involves calculating expenses associated to every housemate. In order to represent all costs and expense associated to every housemate in an accurate and timely fashion, we will formally represent this system using Module Interface Specification.

Let:

- $I$ be the set of all individual items or bills.
- $U$ be the set of all users. (Housemates/Roommates etc)
- $C$ be the cost associated with each item.
- $S$ be a function that maps an item to its sharers.
- $P$ be a function that maps an item to its payment division.

### Individual Costs

For each item $i$ in $I$, there is a cost $c$ associated with it. Formally:

$$C : I \rightarrow R^+$$

This function $C$ maps every item $i$ to its positive cost $c$.

### Sharers of Costs

For each item $i$, there's a subset of users $U_i \subseteq U$ who share its cost. The function $S$ determines which users are responsible for which items:

$$S : I \rightarrow 2^U$$

Here, $2^U$ represents the power set of $U$, denoting all possible subsets of users. Not all users necessarily owe an amount.

### Payment Division

The function $P$ assigns a portion of the item's cost to each user:

$$P : I \times U \rightarrow R^+$$

This means for each item $i$ and user $u$ in $U_i$, $P(i, u)$ returns the amount $u$ owes for $i$. For users not in $U_i$, $P(i, u) = 0$.

#### Constraints

#### Total Payment

The total amount paid by all users for an item must equal the cost of that item:

$$\forall i \in I : \sum_{u \in U} P(i, u) = C(i)$$

#### Non-negative Payment

No user can owe a negative amount:

$$\forall i \in I, \forall u \in U : P(i, u) \geq 0$$

#### Overall User Expense

The total amount owed by a user for all items is:

$$\text{Expense} : U \to R^+$$

$$\text{Expense}(u) = \sum_{i \in I} P(i, u)$$

# 8 Look and Feel Requirements

## 8.1 Appearance Requirements

LF-A1:
- *Description*: The application should have a modern minimalist user interface.
- *Rationale*: A minimalist user interface design will allow users to intuitively use the application.
- *Fit Criterion*: The interface follows the minimalist design principles in Google's material design guidelines.

## 8.2 Style Requirements

N/A

# 9 Usability and Humanity Requirements

## 9.1 Ease of Use Requirements

UH-E1:

- *Description*: The application should be easy to navigate.
- *Rationale*: Users of the application should able to reach the main features of the application clearly.
- *Fit Criterion*: Users of the application should be able to reach the main features of the application within 5 clicks/taps.

## 9.2 Personalization and Internationalization Requirements

UH-P1:

- *Description*: The application will use Canadian English as its primary language.
- *Rationale*: Most of the target audience of the application speak English.
- *Fit Criterion*: The application uses Canadian English when released.

## 9.3 Learning Requirements

UH-L1:

- *Description*: Users of the application should be able to quickly learn how to use it.
- *Rationale*: The application should be easy to learn.
- *Fit Criterion*: During testing a user should able to use all of the features of the application within the first 30 minutes of first opening the application.

## 9.4 Understandability and Politeness Requirements

N/A

### 9.5 Accessibility Requirements

UH-A1:
- *Description*: The application should be usable by partially sighted users.
- *Rationale*: Accessibility accommodations will allow the app to reach a larger target audience.
- *Fit Criterion*: The application uses the various android accessibility features.

# 10 Performance Requirements

### 10.1 Speed and Latency Requirements

P-SL1:
- *Description*: The application should respond to user interactions quickly.
- *Rationale*: Users should feel the app is responsive.
- *Fit Criterion*: The application should respond to user input within 0.5 seconds.

### 10.2 Safety-Critical Requirements

N/A

### 10.3 Precision or Accuracy Requirements

P-PA1:
- *Description*: All monetary information (e.g. bill amounts) will be accurate to two decimal places.
- *Rationale*: Information from cost-splitting system should be accurate.
- *Fit Criterion*: During testing costs will be correctly rounded to two decimal places.

## 10.4 Robustness or Fault-Tolerance Requirements

P-RFT1:
- *Description*: The application should continue to operate locally if it loses connection to the server.
- *Rationale*: Users may temporarily lose connection to the application servers during use.
- *Fit Criterion*: The application should continue to function for 10 minutes after losing connection to application servers. Additionally it should function as normal when connection is resumed.

## 10.5 Capacity Requirements

P-C1:
- *Description*: The application should be able to handle 100 simultaneous users.
- *Rationale*: The application should be handle a lot of users using it at the same time.
- *Fit Criterion*: During testing the application should handle 100 simultaneous users.

## 10.6 Scalability or Extensibility Requirements

N/A

## 10.7 Longevity Requirements

N/A

# 11 Operational and Environmental Requirements

## 11.1 Expected Physical Environment

OE-PE1:
- *Description*: The system will be able to run on mobile devices through the Android OS.

- *Rationale*: Being available on mobile devices will allow the application to reach its target audience.
- *Fit Criterion*: The application is able to run on phones using Android OS.

## 11.2 Wider Environment Requirements

N/A

## 11.3 Requirements for Interfacing with Adjacent Systems

N/A

## 11.4 Productization Requirements

OE-PR1:
- *Description*: The application should be released on the Google Play store.
- *Rationale*: Being available on the Google Play store will allow the application to reach its target audience.
- *Fit Criterion*: The application is available on the Google Play store.

## 11.5 Release Requirements

N/A

# 12 Maintainability and Support Requirements

## 12.1 Maintenance Requirements

M-M1:
- *Description*: The development process for the application should be well documented.
- *Rationale*: Proper documentation will ensure that the system will be maintainable in the future.

- *Fit Criterion*: The documentation should contain an up to date Problem Statement, Development Plan, SRS, Hazard analysis, Software Design, Software testing, etc. documents.

## 12.2 Supportability Requirements

N/A

## 12.3 Adaptability Requirements

N/A

# 13 Security Requirements

## 13.1 Access Requirements

S-A1:
- *Description*: The application should only allow system administrators to access user data.
- *Rationale*: User data should remain private unless necessary.
- *Fit Criterion*: In the application user data will not be accessible by other users.

## 13.2 Integrity Requirements

S-IN1:
- *Description*: The application should prevent incorrect data from being introduced.
- *Rationale*: Incorrect data being introduced into the database could be used to attack the database.
- *Fit Criterion*: In the application users can't introduce incorrect data in any entry fields.

## 13.3 Privacy Requirements

S-P1:
- *Description*: The application should make the user aware of its information collection policy before collecting data from them.

- *Rationale*: Users should be aware of any personal data collected and the reason why it is collected.
- *Fit Criterion*: The application notifies users on first launch the information collection policy.

### 13.4 Audit Requirements

N/A

### 13.5 Immunity Requirements

N/A

## 14 Cultural Requirements

N/A

## 15 Compliance Requirements

### 15.1 Legal Requirements

N/A

### 15.2 Standards Compliance Requirements

C-SC1:
- *Description*: The application should follow the Google Play development standards.
- *Rationale*: The application must follow those standards in order to be released on the Google Play Store.
- *Fit Criterion*: The application is approved to be released on the Google Play Store.

## 16 Open Issues

N/A

# 17 Off-the-Shelf Solutions

## 17.1 Ready-Made Products

For off the shelf solutions there are many individual cost splitting/chore splitting apps available on the App store/Google Play Store. However these apps don't capture the functionality of both a cost-management and chore-management system.

Cost-splitting

- Splid
- Splitwise
- Tricount
- Splitser
- Settle Up

Chore-Splitting

- Tody
- Nipto
- Sweepy
- Chap
- Flatify

## 17.2 Reusable Components

N/A

## 17.3 Products That Can Be Copied

N/A

# 18 New Problems

N/A

# 19 Tasks

## 19.1 Project Planning

| **Deliverable** | **Deadline** |
|---|---|
| Problem Statement, POC Plan, Development Plan | September 25 |
| Requirements Document Revision 0 | October 6 |
| Hazard Analysis 0 | October 20 |
| V&V Plan Revision 0 | November 3 |
| Proof of Concept Demonstration | November 13–24 |
| Design Document Revision 0 | January 17 |
| Revision 0 Demonstration | February 5–February 16 |
| V&V Report Revision 0 | March 6 |
| Final Demonstration (Revision 1) | March 18–March 29 |
| EXPO Demonstration | April TBD |
| Final Documentation (Revision 1) | April 4 |

## 19.2 Planning of the Development Phases

N/A

# 20 Migration to the New Product

N/A

# 21 Costs

N/A

# 22 User Documentation and Training

N/A

# 23 Waiting Room

N/A

# 24 Ideas for Solution

N/A

# Appendix — Reflection

## Identifying Required Skills

What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

Skills

1. Project Management: Since this is a large project with a team of 5, good project management skills are required in order to ensure that deliverables are delivered on time.

2. Presentation Skills: Having good presentation skills will help demonstrate and share the value of our capstone project.

3. CI/CD: Continuous integration and delivery helps automate the testing process, which helps reduce bugs in code. As such, it is an important skill to learn for this capstone project.

4. Mobile Development: The application is being developed for mobile devices, so it is important to learn the tools used in mobile development.

5. Database Management: This application will be using a CRUD database to store data. As such, having a good database design will help ensure good performance for the application.

6. System Design: A good front-end and back-end system design will help ensure the application will meet the requirements listed in this document,

## Acquiring Required Skills

For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?

- **Justin Dang:** For this capstone project the skill I'm really focusing on developing my mobile development skills. This is because I think that mobile development is a really desired skill in the workplace nowadays. To help with this I plan on watching tutorials from the internet about Android development as well as reading some of the documentation on Android and Kotlin in general.

- **Harris Hamid:** This project will involve us to either create a web-app or a mobile app. This involves creating a front-end, back-end system and database to ensure complete functioning of the app. I will be focusing on database design and security issues. I will be brushing up my SQL skills according to the tech-stack we end up using and will be looking into common security issues that mobile and web app usually face in order to mitigate these problems.

- **Fady Morcos:** For this capstone project, I will be focusing on really developing my mobile and graphical interface development skills. This has been something I wanted to master for a long time now. I believe it is very useful to have this skill to allow me to make any creative idea that I have come to life. This would also be a good skill to have has it is very desirable in today's job market.

- **Rizwan Ahsan:** Since this capstone project requires us to create the application from scratch and to utilize different system design skills to create the front-end and back-end. This will help me develop my elicitation skills for the requirements of the products and also allow to create a better product as a whole. For this I plan on watching various tutorials of system design on the internet and researching various books. This will also benefit us on understanding what tech-stack will be better to utilize or not.

- **Sheikh Afsar:** As we dive into this project, I understand that a key part of this project would be the UI and UX design. I am aiming to go beyond just visuals, prioritizing a user centric approach to ensure our project is simple and intuitive. To help with this goal I want to apply what I learned from my other course "Human Computer Interfaces". I also plan to study UX and technical tools which will help me achieve my goal such as Figma. My goal is also to conduct thorough user research while gathering feedback to refine my design.