

Software Requirements Specification for Housemates (Volere)

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

April 4, 2024

Contents

1	Template and Changes	v
2	Naming Conventions and Terminology	v
2.1	Glossary of all terms used by stakeholders	v
3	Purpose of the Project	v
3.1	User Business	v
3.2	Goals of the Project	vi
4	Stakeholders	vi
4.1	Customer	vi
4.2	Other Stakeholders	vii
4.3	Hands-On Users of the Project	vii
4.4	Personas	vii
4.5	Priorities Assigned to Users	vii
4.6	User Participation	viii
4.7	Maintenance Users and Service Technicians	viii
5	Mandated Constraints	viii
5.1	Implementation Environment of the Current System	viii
5.2	Anticipated Workplace Environment	viii
5.3	Schedule Constraints	viii
5.4	Budget Constraints	viii
6	Assumptions	ix
7	Scope	ix
7.1	Product Context	ix
7.2	Individual Product Use Cases (PUC's)	x
7.3	Product Use Case Diagram	xiii
8	Functional Requirements	xiv
8.1	Task Management System	xiv
8.2	Bill Management System	xiv
8.3	Scheduling System	xvi
8.4	Account System	xvi
8.5	Formal Specification of Bill Management System	xvii

9	Look and Feel Requirements	xix
9.1	Appearance Requirements	xix
9.2	Style Requirements	xx
10	Usability and Humanity Requirements	xx
10.1	Ease of Use Requirements	xx
10.2	Personalization and Internationalization Requirements	xx
10.3	Learning Requirements	xx
10.4	Accessibility Requirements	xxi
11	Performance Requirements	xxi
11.1	Speed and Latency Requirements	xxi
11.2	Precision or Accuracy Requirements	xxi
11.3	Robustness or Fault-Tolerance Requirements	xxii
11.4	Capacity Requirements	xxii
11.5	Scalability or Extensibility Requirements	xxii
12	Operational and Environmental Requirements	xxii
12.1	Expected Physical Environment	xxii
12.2	Productization Requirements	xxiii
13	Maintainability and Support Requirements	xxiii
13.1	Maintenance Requirements	xxiii
14	Security Requirements	xxiii
14.1	Access Requirements	xxiii
14.2	Integrity Requirements	xxiv
14.3	Privacy Requirements	xxiv
15	Cultural Requirements	xxiv
15.1	Neutrality Requirements	xxiv
16	Compliance Requirements	xxv
16.1	Standards Compliance Requirements	xxv
17	Off-the-Shelf Solutions	xxv
17.1	Ready-Made Products	xxv

18 Tasks	xxvi
18.1 Project Planning	xxvi
18.2 Planning of the Development Phases	xxvi
19 Traceability Matrices	xxvii

Revision History

Date	Version	Notes
6 October 2023	1.0	Initial Draft
4 April 2024	1.1	Revision 1

1 Template and Changes

This SRS follows the Volere SRS template. A traceability section was added to ensure that the requirements of Housemates are traceable with the other Housemates documentation. Some sections were removed from this document if they didn't apply to the SRS of Housemates (e.g. Open Issues, New Problems, User Documentation and Training, Waiting Room, etc.).

2 Naming Conventions and Terminology

2.1 Glossary of all terms used by stakeholders

- Housemates: The main application being developed.
- Roommates: People living in the same household.
- Group: Users who share common expenses, tasks or schedule for a particular purpose.

3 Purpose of the Project

3.1 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 roommates 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 roommates to live together and overall simplify their lives.

3.2 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 tasks through a task management system.	This allows streamlining the allocation of chores which in return will reduce conflicts and misunderstandings between roommates.
The application will streamline expense sharing through a bill 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.

4 Stakeholders

4.1 Customer

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

4.2 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.

4.3 Hands-On Users of the Project

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

4.4 Personas

Andrew is a student at McMaster University. He lives off-campus in student housing with four other roommates. 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.

4.5 Priorities Assigned to Users

The primary stakeholders for this project are people with roommates 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.

4.6 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.

4.7 Maintenance Users and Service Technicians

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

5 Mandated Constraints

5.1 Implementation Environment of the Current System

The expected environment for this application will be a browser-supported device providing a user-friendly and intuitive application that is really simple to use. With goals of expanding as a mobile app, so that we can cater to a much diverse user base using mobiles with Android or iOS. Users would then have the convenience to access the solution/product on any device they want that suits their needs.

5.2 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 also on mobile devices there could be a wide variety of places where this application could be used.

5.3 Schedule Constraints

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

5.4 Budget Constraints

This project has a budget constraint of \$750.

6 Assumptions

- The users of this application are expected to have a basic understanding of how to access a website through a browser.

7 Scope

7.1 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.

Database:

This acts as the storage for all the information that the application will receive, process and save.

Housemates App:

This is the main software application that the user will interact with to achieve their goals by making requests to different components of the application.

App User:

This is the user interacting with the Housemates app in order to trigger events to request for information to be displayed.

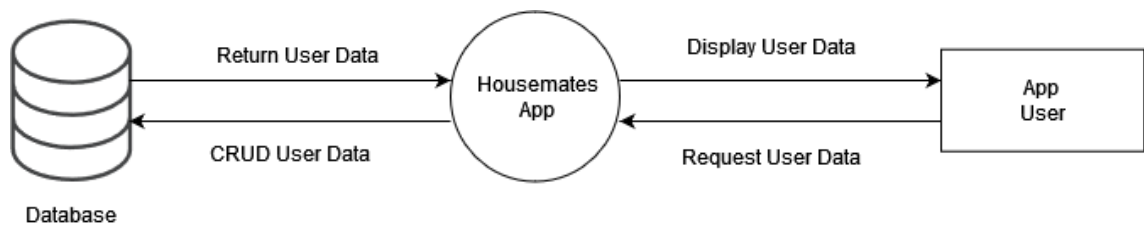


Figure 1: Context Diagram

7.2 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 is present in a group.
 - *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 roommates, 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 roommates 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.

7.3 Product Use Case Diagram

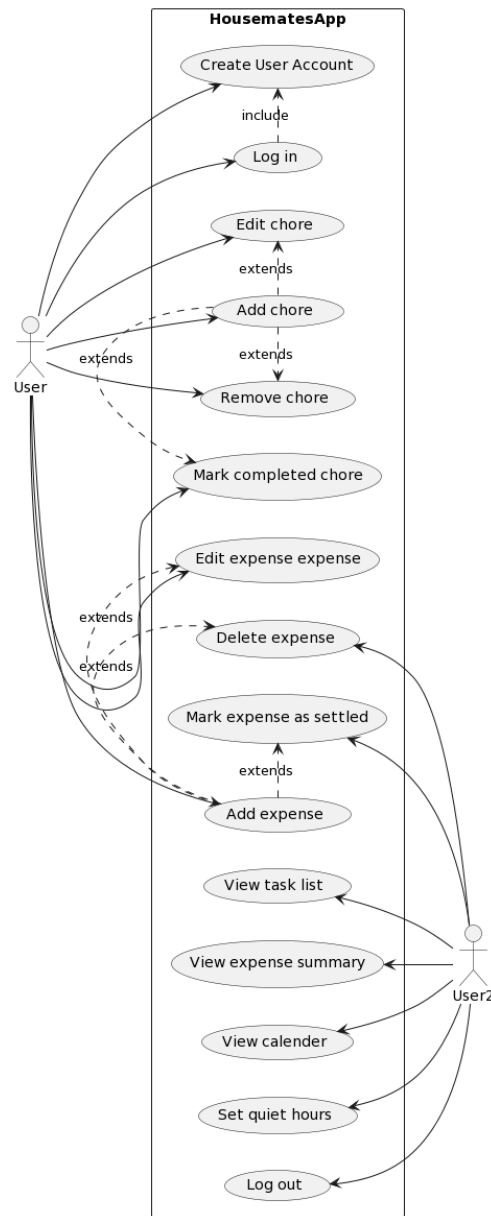


Figure 2: Usecase Diagram

8 Functional Requirements

8.1 Task Management System

- TM1: – *Description*: User tasks will be displayed in task management page.
- *Rationale*: This is to be able to clearly view the tasks which the user must complete.
- TM2: – *Description*: Users are able to create new tasks.
- *Rationale*: Users need to be able to add new tasks as needed.
- *Fit Criterion*: User able to see task added to bill management page.
- TM3: – *Description*: Users are able to assign tasks to roommates.
- *Rationale*: This feature enables users to divide tasks among roommates.
- *Fit Criterion*: Other users are able to see task on their end.
- TM4: – *Description*: 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*: Task is removed from current tasks and moved to completed tasks.

8.2 Bill Management System

- BM1: – *Description*: Users are able to create a new bill.
- *Rationale*: This requirement allows users to create new bills as needed.
- BM2: – *Description*: Users are able to assign bill to a roommate and split it amongst them evenly.
- *Rationale*: This functionality will allow users to keep track of bill splitting easily.

- *Fit Criterion:*
 1. The bill amount is correctly split evenly amongst the roommates.
 2. The total bill amount is shown on the screen and the amount required to pay by each roommate 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.
- 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.
- BM6:
 - *Description:* Users are able to assign bill to a roommate and split it amongst them unevenly.
 - *Rationale:* This functionality will allow users to split unevenly if required. E.g. Splitting a dinner bill where each user ordered different items with different prices.
 - *Fit Criterion:*
 1. The bill amount is correctly split amongst the roommates.
 2. The total bill amount is shown on the screen and the amount required to pay by each roommate is displayed correctly.
- BM7:
 - *Description:* Users are able to settle expenses from bills.

- *Rationale*: This functionality will allow users to remove the expense after they paid it.
- *Fit Criterion*:
 1. Once their portion of the bill is paid off the it should show on screen that the amount the user owes for the bill should be 0.

8.3 Scheduling System

- SS1:
- *Description*: Users are able to create new events and schedule them within the scheduling feature.
 - *Rationale*: This functionality allows users to plan and coordinate activities efficiently.
 - *Fit Criterion*: Assigned users are able to see event on scheduling page.
- SS2:
- *Description*: Users can view all the events scheduled by other users on their calendar within the scheduling feature.
 - *Rationale*: Event visibility ensures that all users are aware of shared activities and can plan accordingly.
 - *Fit Criterion*: Users can clearly see and distinguish between their personal events and shared user events on the calendar.

8.4 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 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.~~

8.5 Formal Specification of Bill Management System

A critical aspect of this project involves calculating expenses associated to every roommate. In order to represent all bills and expense associated to every roommate 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.
- U be the set of all users.
- 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 . If a user u does not share the cost of some item i , that is $u \notin S(i)$, then $P(i, u) = 0$.

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:

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

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

9 Look and Feel Requirements

9.1 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](#).

9.2 Style Requirements

- LF-S1:
- *Description*: All pages in the application must follow a similar user interface
 - *Rationale*: This allows for consistency of style and less ambiguity.
 - *Fit Criterion*: During beta testing, potential users able to determine whether pages looks are consistent.

10 Usability and Humanity Requirements

10.1 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.

10.2 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.

10.3 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 be able to use all of the features of the application within the first 30 minutes of first opening the application.

10.4 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](#).

11 Performance Requirements

11.1 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.

11.2 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.

11.3 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.

11.4 Capacity Requirements

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

11.5 Scalability or Extensibility Requirements

- P-SE1:
- *Description*: The application should horizontally scale when the number of users increase.
 - *Rationale*: To ensure consistent performance among users.
 - *Fit Criterion*: Use performance testing tool to check response times.

12 Operational and Environmental Requirements

12.1 Expected Physical Environment

- OE-PE1:
- *Description*: The system will be able to run on devices that support a browser.
 - *Rationale*: Being available on such devices will allow the application to reach a wide target audience.
 - *Fit Criterion*: The application is able to run on phones, tablets and PCs.

12.2 Productization Requirements

- OE-PR1:
- *Description*: The application should be deployed on a server.
 - *Rationale*: Being deployed entails that anyone on the internet can use the application.
 - *Fit Criterion*: The application is accessible through its unique URL.

13 Maintainability and Support Requirements

13.1 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.

14 Security Requirements

14.1 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.

14.2 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.

14.3 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.

15 Cultural Requirements

15.1 Neutrality Requirements

- CR-N1:
- *Description*: The user interface must not be indicative to any particular culture.
 - *Rationale*: The symbols or colors in the app may offend some users.
 - *Fit Criterion*: During field testing, users from different backgrounds can approve of the design.

16 Compliance Requirements

16.1 Standards Compliance Requirements

- C-SC1:
- *Description*: The application should inform users beforehand about the data being collected and stored from them.
 - *Rationale*: This allows for transparency between provider and users.
 - *Fit Criterion*: A terms and condition page is displayed before they are able to interact with the application.

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](#)

18 Tasks

18.1 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

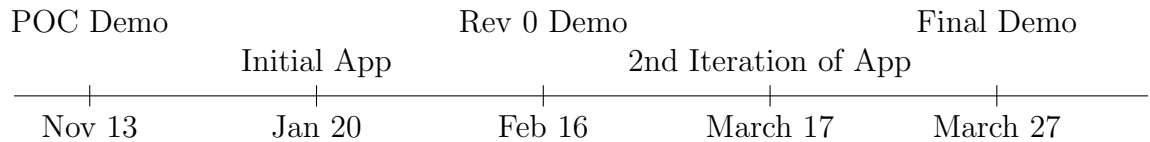
18.2 Planning of the Development Phases

Below is a timeline of the development phases we wish to follow.

Phase 1: Set-up a server in which multiple clients can communicate. This will be presented as POC Demo.

Phase 2: Start Implementing core functionalities such as Task Management and Bill Management by designing a front-end and setting up a database.

Phase 3: After feedback front Rev 0 and carrying out testing, we will incorporate those changes into our application.



19 Traceability Matrices

The following traceability matrix provides a mapping of the functional requirements to the non-functional requirements. Req ID shows the functional requirement IDs and criterion ID shows the non-functional IDs relating to a specific requirement.

Req.ID	Requirement Description	Criterion ID
TM1	Users tasks displayed in task management system.	LF-A1, UH-A1
TM2	Users can create new tasks.	UH-L1, P-SL1
TM3	Users can assign tasks.	UH-L1
TM4	Users can mark tasks as completed.	UH-A1
BM1	Users can create a new bill.	UH-L1, P-SL1
BM2	Users can assign bills among other users and split amongst them evenly.	UH-E1, P-PA1
BM3	Users can modify already inputted bill details	P-RFT1
BM4	Users can categorize the bills.	UH-P1
BM5	Users can keep track of which bills have been paid off.	P-RFT1
BM6	Users can assign bills among other users and split amongst them unevenly.	UH-E1, P-PA1
BM7	Users can settle expenses from bills.	UH-E1, P-PA1
SS1	Users are able to create new events and schedule them within the scheduling feature.	UH-P1
SS2	Users can view all events scheduled by other users on their calendar.	UH-A1, LF-S1
AS1	Users can create new account by providing their personal information.	S-IN1, S-P1
AS2	Users are able to log in to the system by providing their email address and password.	S-A1
AS3	Users are able to update their profile information.	UH-P1,S-IN1
AS4	Users are able to remove their account from the system.	UH-P1

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 as 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.