Assignment 1

Software Requirements Specification for MyBankUML

COMP 354
Introduction to Software Engineering

Mykyta Onipchenko (40281057)
Bogdan Ivan (40283137)
Mate Barabas (40285661)
Veronika Pontolillo (40286664)
Kimberley Kam Sing (40319425)
Gorden Quach (40263250)

Due: October 5th, 2025

Table of Contents

(1) Introduction

The purpose of this document is to analyze and define high-level needs for features to be added to the BankUML application. BankUML is a Java-based application simulating banking operations. This application is to be extended into MyBankUML by adding 3 features detailed below. This document details the functionalities required by stakeholders and target users,along with the reasons behind these needs. The specific ways in which the system addresses them are described in the use cases and supplementary specifications, which help guide the organization's overall strategy.

The numbers in the diagram titles correspond to their numbers in the SRS example document and thus are not sequential since we chose to omit certain less relevant diagrams.

(2) Positioning

(2.1) Problem Statement

Category	Description
The problem of	Developing a platform that makes it easy for customers to perform their banking operations.
Affects	Customers who want to consult their information, transfer money, or look up other people's accounts.
The impact of which	That it's difficult to manage banking accounts.
A successful solution would be	A banking platform allowing aforementioned operations.

(2.2) Product Position Statement

Category	Description
For	Customers needing an easily managed banking account.
Who	Are looking for one with previously mentioned features.
MyBankUML	Is a Java-based application
That	Enables customers to check their account information and search for other account information.
Unlike	The current version of MyBank doesn't have any user interactivity, meaning it does not provide meaningful and relevant accessibility to users.
Our product	Our newer version allows bank employees to consult client information (name, account number, type, etc.); allows the search and filtering of the accounts with the aid of a database; and allows for more interactivity and security with the implementation of role-based permissions.

(3) Stakeholders and Users

(3.1) Stakeholder and User Summary

We decided to merge these diagrams together because users are self-represented

Name	Description	Responsibilities	Stakeholder
Customer	A customer that wants to consult their own account information.	Consult all their information. Transfer between their accounts.	Self-represented.
System Administrator	An administrator responsible for the stability of the bank.	See the state of the bank.	Self-represented.
Bank employee	An employee of a branch.	See information about customers in the branch. Transfer between accounts.	Self-represented.

(3.2) User Environment

Customers need
- PC (laptop or desktop) with a stable network
- Correctly installed JVM
- Graphically capable computer (graphical OS, video drivers) with peripherals (screen, keyboard, and mouse)
- Email they have access to (for account creation and log in)

(3.3) Key Stakeholder and User needs

Need	Priority	Concerns	Current Solution	Proposed Solution
Manage and view account information	High	Principle need for application, security purposes and tracking	Manual management	Graphical interface
Manage bills and payments	High	Ease task of paying bills	Manual management	Sending reminder notifications
Maintain tools and application status	High	Prevent system outage with tools to easily fix any problems	None	Generate the tools required to maintain and fix the application easily
Manage customer accounts	High	Surveil malicious or fraudulent activity, track payments that are due	Manual management	Maintain customer information with updated tools status in the system
Manage employee activity	High	Security and integrity	None	Create a logging system to track activity
Manage branch activity	Medium	Security and integrity	Manual management	Create a logging system to track activity and send to a repository for easy checking
Apply permissions based on type of user	High	Security and integrity, prevent malicious users or use of services	None	Create permission hierarchy to apply to everyone using the application
Search tools	Medium	Flexible search, response time and accuracy	None	Provide flexibility to search tools and equipment

(5) Product Features

ID	Features	Description
1	Create an account / sign-up	Customers can open a bank account and sign in with their card number and password
2	Login	In order to perform functions related to account management and bill payments users must be able to login with their card number and password.
3	Password recovery	Allow users to create a new password through verification.
4	User account dashboard	Provides users with information related to accounts and allows them to access sections for payment.
5	Credit card / bill payment	Allow users to input a sum of money to pay off their cards or bills
6	Manage savings account	Transfer money into savings or out of savings
7	Add / remove branch	Allows administrator to add or remove branches from system
8	Add / remove employees	Allows admin to manage employees
9	Assign role based permissions	Allows admin to assign roles to employees and what they have access to.
10	Lock or unlock customer accounts	Allows employees to lock/unlock customer accounts for security purposes
11	Lock or unlock card	Allows employee or customer to lock/unlock their card for security reasons
12	Tools and branch information	Displays tool status and branch status to employee and administrator.
13	Tool customer account management	Tool that allows employees to access customer accounts and manage it when needed for security purposes.
14	Search tool	Tool that allows to search for a specific customer account throughout the database

(7) Use Cases

ID: UC-01

Use case: Locking or Unlocking a card

Description: This use case happens when the customer has its bank card lost or stolen. The user can lock its card using this feature by selecting the option to lock/unlock the card using the user interface of MyBankUML then selecting whether they want to lock or unlock it.

Level: User Goal / Subfunction

Primary Actor: Customer

Supporting Actor: Bank Employee

Stakeholders and interests:
- Customer: Unlocks or locks their card if needed.
- Bank Employee: Access status of the customer's card and can lock/unlock it depending on the customer's needs.
- Developer: Develops the user interface for the clients to be able to intuitively lock or unlock their card. Also, develops the feature of locking the client's card where all transfers and access to the card is blocked until it is unlocked.
- Tester: Tests the UI to see if it meets the requirements and tests if the lock feature blocks all access and use of the client's card.

Pre-conditions:
- Customers must have a card and an account at MyBankUML
- Customers must be 18 years old or older in order to have a credit card.
- Customers must login into their account before accessing this feature.

Post-conditions:
1. **Successful Scenario:**
    1.1. Customers successfully log into their account in MyBankUML.
    1.2. Customers select the option of locking/unlocking their card via the user interface.
    1.3. The terminal or the GUI returns the confirmation that their card has been locked/unlocked and the card blocks all access and transfers if locked.
2. **Failure Scenario:**
    2.1. The card that has been locked still allows transfers and access to the card.
    2.2. The card is still locked after requesting to unlock the card.

ID: UC-02

Use case: Transferring Money Between Accounts

Description: This use case happens when customers have multiple accounts in MyBankUML (i.e checking and savings account) and want to transfer money between these accounts. It allows for more accessibility and flexibility to the clients.

Level: User Goal

Primary Actor: Customer

Supporting Actor: Bank Employee

Stakeholders and interests:
- Customer: transfers their money between their accounts
- Bank Employee: Access the client's accounts and transfers the amount requested by the clients to their more secured accounts (i.e investment/wealth management accounts).
- Developer: Develops the user interface for the clients as well as the functions to be able to intuitively and securely move money between the user's accounts.
- Tester: Tests the UI to see if it meets the requirements and tests if the money has been safely transferred between different accounts.

Pre-conditions:
- Customers must multiple accounts at MyBankUML
- Customers must login into their account before accessing this feature.
- Bank employees must have been given authorization by the clients using a two-factor authentication from the clients in order for them to transfer the client's money between more secured accounts.

Post-conditions:
1. **Successful Scenario:**
    1.1. Customers successfully log into their account in MyBankUML.
    1.2. Customers select the option of transfer funds between accounts via the user interface.
    1.3. The terminal or the GUI returns the confirmation that the transfer has been successful and the accounts show the respective new amounts after the transfer.
    1.4. All of the above but for bank employees while given access to high-security accounts.
2. **Failure Scenarios:**
    2.1. The money has not been transferred and the money was lost during the transfer.
    2.2. The money has been transferred but the accounts have not been updated with the new amounts for low-security and high-security accounts.

(8) User Stories

Display account information		Search for other user accounts		Transfer between accounts	
Log in	Access account	Search	Access account	Find destination account	Transfer
Enter email	Navigate to details page	Navigate to search page	Navigate to other users details age	Navigate to detail page of the destination account (could be your own or someone else's)	Navigate to transfer page
Enter password	See all account attributes (based on permissions)	Compose a search query into the search bar	See some account attributes (based on permissions)	Copy the account ID	Select source account from the drop down
Press login button	Go back to home page	Click on a relevant account in the search results	Go back to home page	Navigate to home page	Paste destination account ID
Navigate to home page					Enter the amount
					Press transfer button
					Navigate back to home page

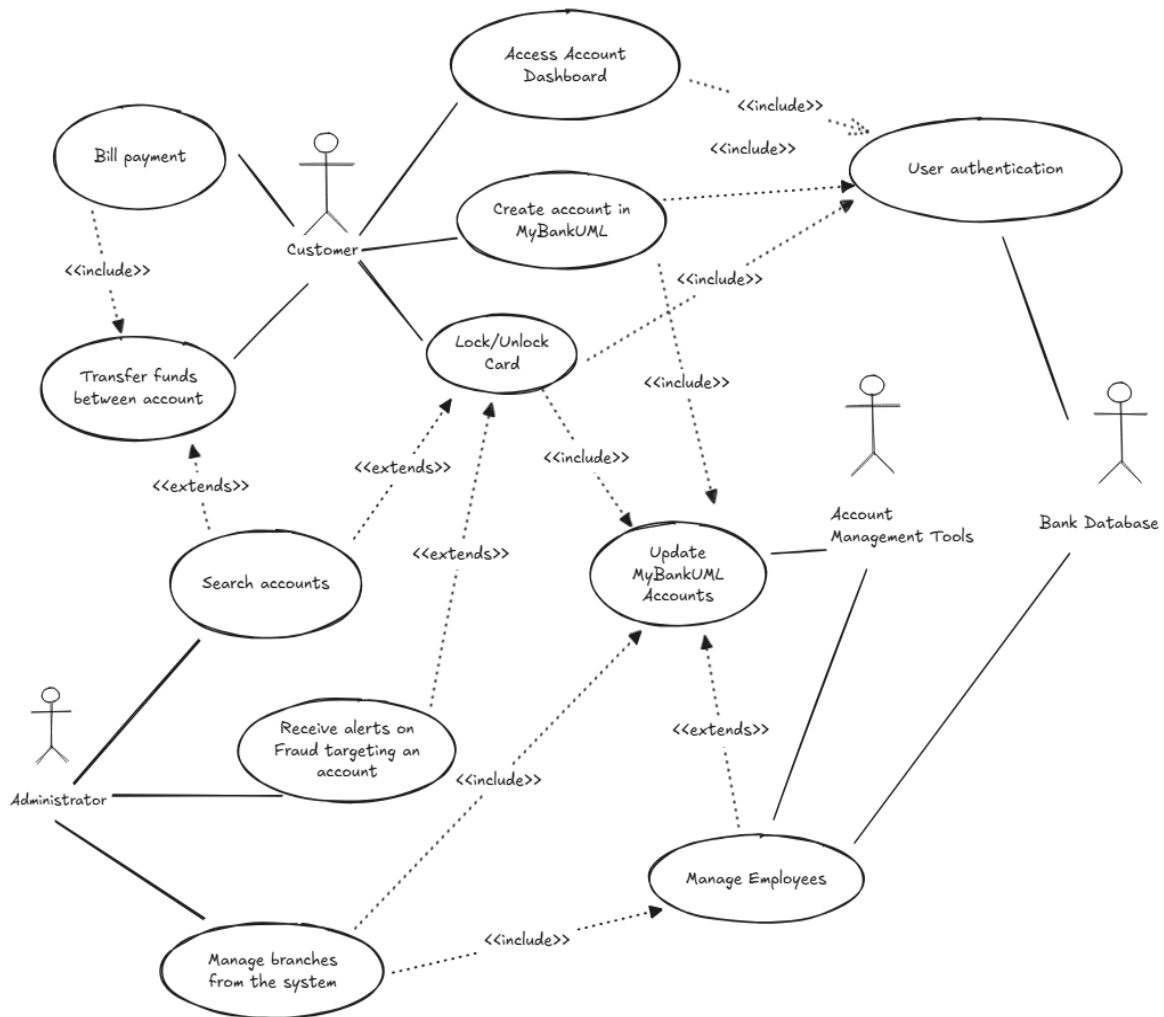# (9) UML Diagrams

(9.1) Use Case Diagram

Figure 1: Use Case Diagram

(9.2) Sequence Diagrams

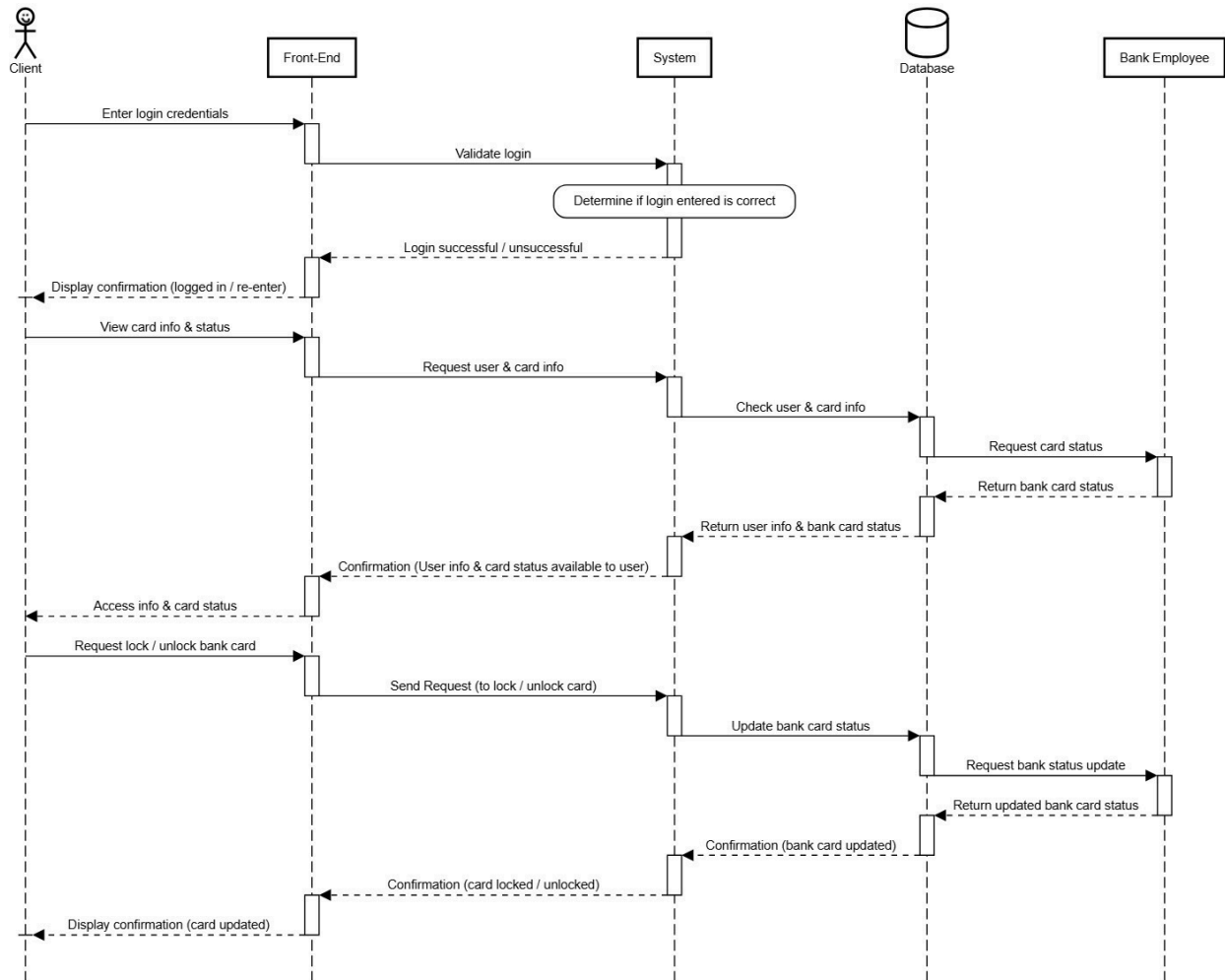## 9.2.1. Locking or Unlocking a Bank Card (In Case of Loss or Theft)

Figure 2: Sequence Diagram of locking or unlocking a bank card

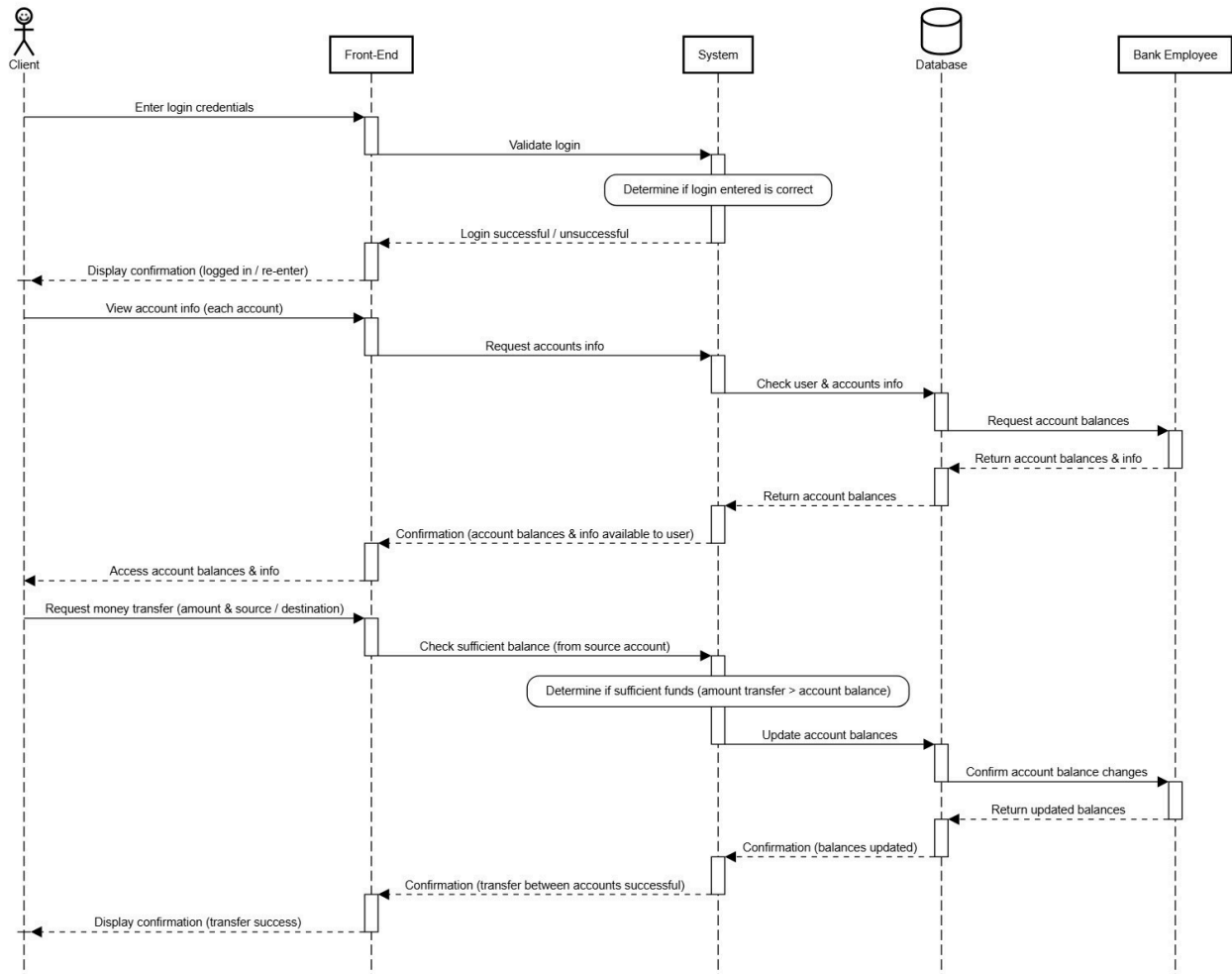## 9.2.2. Transferring Money Between Client's Accounts

Figure 3: Sequence Diagram of transferring money between accounts
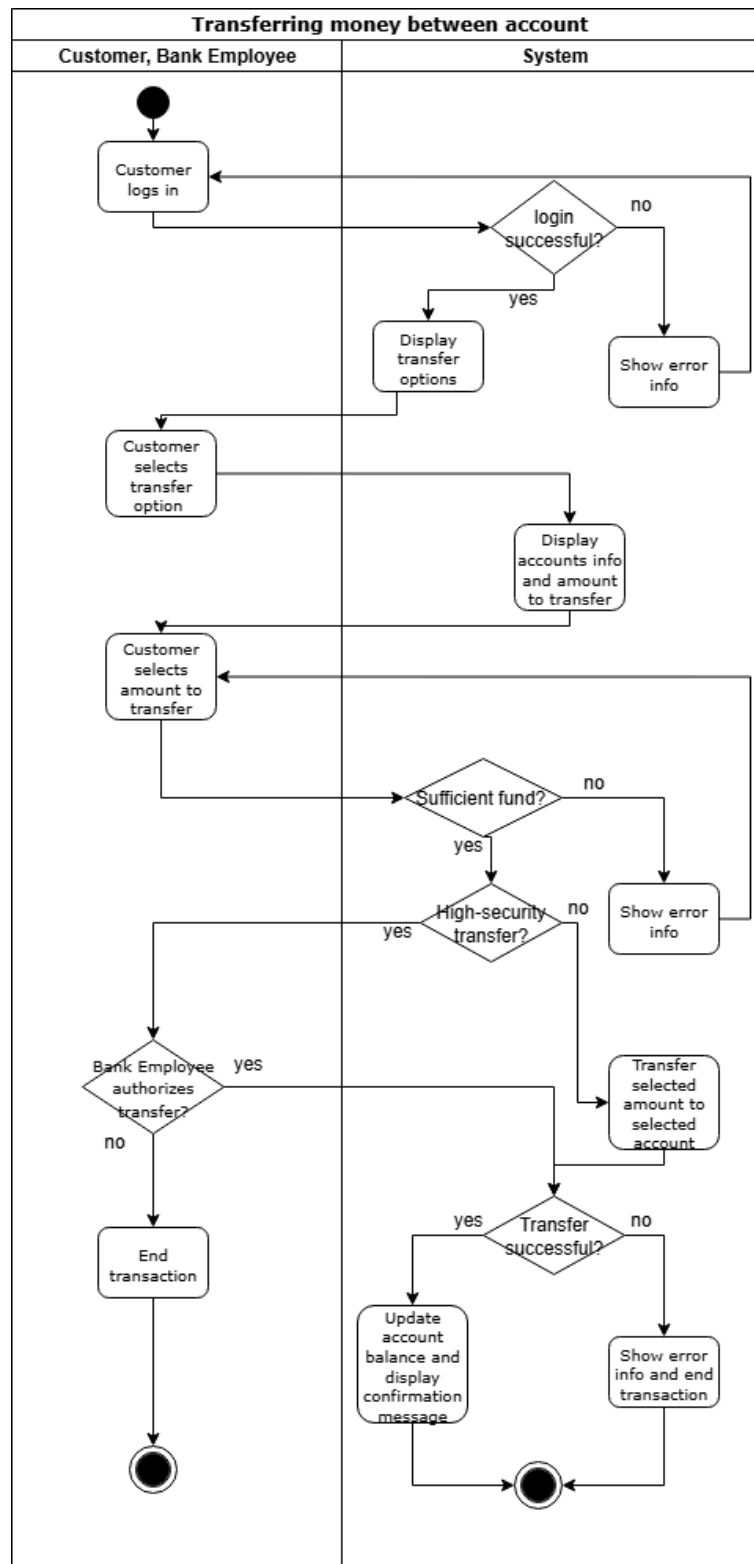
(9.3) Activity Diagram

Figure 4: Activity Diagram of transferring money between accounts

(10) Road Map

Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13
Form Teams	██	██											
Assignment 1			██	██	██								
Assignment 2					██	██	██						
Assignment 3 Tentative								██	██	██			
Implement Features 1,2,3						██	██	██					
Implement Features 5,6,7								██	██				
Implement Features 8,9,10,11										██	██		
Implement Features 12,13,14												██	██
Implement Feature 4						██	██	██	██	██	██	██	██
Team Meetings		██	██	██	██	██	██	██	██	██	██	██	██

Figure 5: Road map for accomplishing MyBankUML project this semester

(11) Appendix

(11.1) Activity Log

Date	Name(s)	Task(s)	Time
2025-09-17	**Entire team**	● Created a GitHub organization, forked the project ● Set up the project	1h
2025-09-26	Mykyta Onipchenko	● Consulted TA during tutorial period to clarify project requirements ● Started problem statement, product position, stakeholder / user summary, and user environment	2h30min
2025-09-28	Gorden Quach	● Completed the Problem Statement table	20mins
2025-09-29	Mykyta Onipchenko	● Started activity log and user stories	1h
2025-09-30	**Entire team**	● Sync between team members ● Clarified work division	15mins
2025-10-01	Bogdan Ivan	● Completed key stakeholder and user needs	1h
2025-10-01	Mate Barabas	● Completed features table ● Completed roadmap	2h
2025-10-02	Veronika Pontolillo	● Completed sequence diagrams ● Contributed to document organization	2h30mins
2025-10-02	Gorden Quach	● Completed use cases ● Completed use cases UML diagram	3h30mins
2025-10-02	Kimberley Kam Sing	● Completed activity diagram	1h30min
2025-10-03	**Entire team**	● Team review	1h
2025-10-04	**Entire team**	● Completed contributions table	15mins
2025-10-05	**Entire team**	● Prepared the document for submission	30mins

(11.2) Contributions

Name	Diagrams
Mykyta Onipchenko	● Problem statement, product position ● Stakeholder / user summaries ● User environment ● User stories
Bogdan Ivan	● Stakeholder needs ● User needs
Mate Barabas	● Features table ● Roadmap
Veronika Pontolillo	● Sequence diagrams
Kimberley Kam Sing	● Activity diagram
Gorden Quach	● Problem statement ● Use cases ● Use cases UML