

## **Chapter 2: Planning and Requirements**

## 2.1 Planning

### 2.1.1 Scope Initiation (WBS)

The following figures will contain the Work Breakdown Structure (WBS), which demonstrates the activities needed to finish the project, in addition to the resources used in each of the activities:

	WBS	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	0	🚀	📌 SUP Project	50 days	Wed 17-03-21	Mon 24-05-21		
2	1	🚀	📌 Project Initiation	4 days	Wed 17-03-21	Sun 21-03-21		
3	1.1	🚀	Problem definition	4 days	Wed 17-03-21	Sun 21-03-21		Wael Khaled,Microsoft Word[1]
4	1.2	🚀	Project Objectives	2 hrs	Thu 18-03-21	Thu 18-03-21		Microsoft Word[1],Mohammad Alfaqeeh,Wael Khaled
5	1.3	🚀	current and existing systems	3 hrs	Thu 18-03-21	Thu 18-03-21		Mohammad Alfaqeeh,Microsoft Word[1]
6	1.4	🚀	Literature Review	4 days	Wed 17-03-21	Sun 21-03-21		Wael Khaled,Microsoft Word[1]
7	1.5	🚀	Stakeholder list	1 hr	Fri 19-03-21	Fri 19-03-21		Microsoft Word[1],Mohammad Alfaqeeh
8	1.6	🚀	Proposed Scope	1 hr	Fri 19-03-21	Fri 19-03-21		Wael Khaled,Microsoft Word[1]
9	1.7	🚀	Process Model	8 hrs	Fri 19-03-21	Fri 19-03-21	8	Wael Khaled,Microsoft Word[2]
10	1.8	🚀	Scope Excluded	2 hrs	Sat 20-03-21	Sat 20-03-21		Mohammad Alfaqeeh,Microsoft Word[1]
11	1.9	🚀	Project Constraints	1 hr	Sun 21-03-21	Sun 21-03-21		Microsoft Word[1],Mohammad Alfaqeeh
12	2	🚀	📌 Project Planning	8 days	Thu 25-03-21	Fri 02-04-21		
13	2.1	🚀	Scope Initiation	3 days	Thu 25-03-21	Sat 27-03-21	8	Microsoft Project[1],Mohammad Alfaqeeh,Wael Khaled
14	2.1.1	🚀	Activity Revision and study	1 day	Sun 28-03-21	Sun 28-03-21	13	Mohammad Alfaqeeh,Wael Khaled
15	2.2	🚀	Resource Planning	1 hr	Mon 29-03-21	Mon 29-03-21	13	Microsoft Project[1],Wael Khaled
16	2.3	🚀	Risk List and Analysis	4 hrs	Mon 29-03-21	Mon 29-03-21	13	Microsoft Word[1],Mohammad Alfaqeeh
17	2.4	🚀	Cost Estimation	1 hr	Mon 29-03-21	Mon 29-03-21	13	Wael Khaled
18	2.5	🚀	Information Gathering	2 days	Thu 25-03-21	Fri 26-03-21		Wael Khaled,Mohammad Alfaqeeh
19	2.6	🚀	Requirement List (initial)	2 hrs	Sat 27-03-21	Sat 27-03-21	18	Microsoft Word[1],Wael Khaled
20	2.7	🚀	📌 Increment 1	5 days	Sat 27-03-21	Thu 01-04-21	18	
21	2.7.1	🚀	Functional Requirements Specification	2 days	Sat 27-03-21	Mon 29-03-21		Microsoft Word[1],Mohammad Alfaqeeh
22	2.7.2	🚀	Non-Functional Requirements Specification	1 day	Sat 27-03-21	Sat 27-03-21		Microsoft Word[1],Mohammad Alfaqeeh
23	2.7.3	🚀	Requirement Analysis	1 day	Tue 30-03-21	Tue 30-03-21	21,22	Mohammad Alfaqeeh,Wael Khaled

Figure 1- WBS 1

	WBS	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
GANNT CHART	24	2.7.4	Requirement Specification	1 day	Wed 31-03-21	Wed 31-03-21	23	Microsoft Word[1],Wael Khaled
	25	2.8	<b>Increment 2</b>	5 days	Fri 02-04-21	Thu 08-04-21		
	26	2.8.1	Discuss Requirement Specification	2 hrs	Fri 02-04-21	Fri 02-04-21	24	Mohammad Alfaqeeh,Wael Khaled
	27	2.8.2	Specify functional Requirements	1 day	Sat 03-04-21	Sat 03-04-21		Wael Khaled
	28	2.8.3	Specify non-functional Requirements	1 day	Sun 04-04-21	Sun 04-04-21		Wael Khaled
	29	2.8.4	Requirement Analysis	2 days	Mon 05-04-21	Tue 06-04-21		Mohammad Alfaqeeh,Wael Khaled
	30	2.8.5	Requirement Specification(final)	2 days	Wed 07-04-21	Thu 08-04-21	29	Microsoft Word[1],Mohammad Alfaqeeh
	31	3	<b>Increment 3 (Design)</b>	8 days	Fri 09-04-21	Tue 20-04-21		
	32	3.1	Design Sign up Screen	2 days	Fri 09-04-21	Mon 12-04-21		Android Studio V.4.1[1],Wael Khaled
	33	3.2	Design Login Screen	1 day	Mon 12-04-21	Mon 12-04-21		Android Studio V.4.1[1],Wael Khaled
	34	3.3	Design Chat Screen	1 day	Tue 13-04-21	Tue 13-04-21		Android Studio V.4.1[1],Mohammad Alfaqeeh
	35	3.4	Design Friend Request and Friend List Screen	1 day	Thu 15-04-21	Thu 15-04-21		Android Studio V.4.1[1],Mohammad Alfaqeeh
	36	3.5	Design Scanner Screen	1 day	Fri 16-04-21	Fri 16-04-21		Android Studio V.4.1[1],Wael Khaled
	37	3.6	Design Reminder and settings page	1 day	Sat 17-04-21	Sat 17-04-21		Android Studio V.4.1[1],Wael Khaled
	38	3.7	Test and bug fixes	3 days	Sat 17-04-21	Tue 20-04-21		Android Studio V.4.1[1],Mobile Phone (Android Based)[1],Mohammad Alfaqeeh,Wael Khaled
	39	4	<b>Increment 4 (Implementation)</b>	8 days	Wed 21-04-21	Fri 30-04-21		
	40	4.1	Sign up page implementation	2 days	Wed 21-04-21	Thu 22-04-21		Android Studio V.4.1[1],Google Firebase[1],Wael Khaled
	41	4.2	Login Page Implementation	1 day	Fri 23-04-21	Fri 23-04-21	40	Android Studio V.4.1[1],Google Firebase[1],Wael Khaled
42	4.3	Scanner Page Implementation	1 day	Wed 21-04-21	Wed 21-04-21		Android Studio V.4.1[1],Mohammad Alfaqeeh	

Figure 2- WBS 2

	WBS	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
GANNT CHART	43	4.4	Reminder Page Implementation	2 days	Thu 22-04-21	Fri 23-04-21		Android Studio V.4.1[1],Mohammad Alfaqeeh
	44	4.5	Bug fixes and error handling	2 days	Sat 24-04-21	Mon 26-04-21		Android Studio V.4.1[1],Mohammad Alfaqeeh,Wael Khaled
	45	4.6	Unit Testing and Integration	3 days	Mon 26-04-21	Wed 28-04-21		Android Studio V.4.1[1],Mobile Phone (Android Based)[1],Mohammad Alfaqeeh,Wael Khaled
	46	4.7	System Tests	2 days	Thu 29-04-21	Fri 30-04-21	45	Android Studio V.4.1[1],Mobile Phone (Android Based)[1],Mohammad Alfaqeeh,Wael Khaled
	47	5	<b>Increment 5 (design improvements)</b>	5 days	Sat 01-05-21	Thu 06-05-21		
	48	5.1	Color Palette improvements	2 days	Sat 01-05-21	Mon 03-05-21		Android Studio V.4.1[1],Mohammad Alfaqeeh,Wael Khaled
	49	5.2	Item Positioning Improvements	2 days	Tue 04-05-21	Wed 05-05-21		Android Studio V.4.1[1],Mohammad Alfaqeeh,Wael Khaled
	50	5.3	Tests and bug fixes	1 day	Thu 06-05-21	Thu 06-05-21		Android Studio V.4.1[1],Mobile Phone (Android Based)[1],Mohammad Alfaqeeh,Wael Khaled
	51	6	<b>Increment 6 (Code Improvements)</b>	7 days	Fri 07-05-21	Mon 17-05-21		
	52	6.1	Sign up Page code cleanup	2 days	Fri 07-05-21	Mon 10-05-21		Android Studio V.4.1[1],Wael Khaled
	53	6.2	Login Page code cleanup	1 day	Mon 10-05-21	Mon 10-05-21		Android Studio V.4.1[1],Wael Khaled
	54	6.3	Chat page code cleanup	2 days	Tue 11-05-21	Wed 12-05-21		Android Studio V.4.1[1],Mohammad Alfaqeeh
	55	6.4	Reminder page code cleanup	1 day	Thu 13-05-21	Thu 13-05-21		Android Studio V.4.1[1],Mohammad Alfaqeeh
	56	6.5	Scanner and settings page code cleanup	1 day	Fri 14-05-21	Fri 14-05-21		Android Studio V.4.1[1],Mohammad Alfaqeeh
	57	6.6	Final Application Tests	2 days	Sat 15-05-21	Mon 17-05-21		Android Studio V.4.1[1],Mobile Phone (Android Based)[1],Mohammad Alfaqeeh,Wael Khaled
	58	7	<b>Increment 7 (Deployment)</b>	5 days	Tue 18-05-21	Mon 24-05-21	57	
59	7.1	Database control tests	2 days	Tue 18-05-21	Wed 19-05-21		Android Studio V.4.1[1],Google Firebase[1],Wael Khaled	
60	7.2	Security tests	2 days	Thu 20-05-21	Fri 21-05-21		Mobile Phone (Android Based)[1],Mohammad Alfaqeeh	

Figure 3- WBS 3

	WBS	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
	44	4.5	Bug fixes and error handling	2 days	Sat 24-04-21	Mon 26-04-21		Android Studio V.4.1[1],Mohammad Alfaqeeh,Wael Khaled
	45	4.6	Unit Testing and Integration	3 days	Mon 26-04-21	Wed 28-04-21		Android Studio V.4.1[1],Mobile Phone (Android Based)[1],Mohammad Alfaqeeh,Wael Khaled
	46	4.7	System Tests	2 days	Thu 29-04-21	Fri 30-04-21	45	Android Studio V.4.1[1],Mobile Phone (Android Based)[1],Mohammad Alfaqeeh,Wael Khaled
	47	5	<b>Increment 5 (design improvements)</b>	<b>5 days</b>	<b>Sat 01-05-21</b>	<b>Thu 06-05-21</b>		
	48	5.1	Color Palette improvements	2 days	Sat 01-05-21	Mon 03-05-21		Android Studio V.4.1[1],Mohammad Alfaqeeh,Wael Khaled
	49	5.2	Item Positioning Improvements	2 days	Tue 04-05-21	Wed 05-05-21		Android Studio V.4.1[1],Mohammad Alfaqeeh,Wael Khaled
	50	5.3	Tests and bug fixes	1 day	Thu 06-05-21	Thu 06-05-21		Android Studio V.4.1[1],Mobile Phone (Android Based)[1],Mohammad Alfaqeeh,Wael Khaled
	51	6	<b>Increment 6 (Code Improvements)</b>	<b>7 days</b>	<b>Fri 07-05-21</b>	<b>Mon 17-05-21</b>		
	52	6.1	Sign up Page code cleanup	2 days	Fri 07-05-21	Mon 10-05-21		Android Studio V.4.1[1],Wael Khaled
	53	6.2	Login Page code cleanup	1 day	Mon 10-05-21	Mon 10-05-21		Android Studio V.4.1[1],Wael Khaled
	54	6.3	Chat page code cleanup	2 days	Tue 11-05-21	Wed 12-05-21		Android Studio V.4.1[1],Mohammad Alfaqeeh
	55	6.4	Reminder page code cleanup	1 day	Thu 13-05-21	Thu 13-05-21		Android Studio V.4.1[1],Mohammad Alfaqeeh
	56	6.5	Scanner and settings page code cleanup	1 day	Fri 14-05-21	Fri 14-05-21		Android Studio V.4.1[1],Mohammad Alfaqeeh
	57	6.6	Final Application Tests	2 days	Sat 15-05-21	Mon 17-05-21		Android Studio V.4.1[1],Mobile Phone (Android Based)[1],Mohammad Alfaqeeh,Wael Khaled
	58	7	<b>Increment 7 (Deployment)</b>	<b>5 days</b>	<b>Tue 18-05-21</b>	<b>Mon 24-05-21</b>	57	
	59	7.1	Database control tests	2 days	Tue 18-05-21	Wed 19-05-21		Android Studio V.4.1[1],Google Firebase[1],Wael Khaled
	60	7.2	Security tests	2 days	Thu 20-05-21	Fri 21-05-21		Mobile Phone (Android Based)[1],Mohammad Alfaqeeh
	61	7.3	Deployment Testing	2 days	Fri 21-05-21	Mon 24-05-21		Android Studio V.4.1[1],Mobile Phone (Android Based)[1],Mohammad Alfaqeeh,Wael Khaled

Figure 4- WBS 4

## 2.1.2 Gantt Chart

The following figures will contain the Gantt chart, which will demonstrate the project activities (Y-axis) with the project's timetable (X-axis) and will also show the links and dependencies between the different tasks and increments.

The Gantt chart will also emphasize the incremental process model that we decided to adopt in this project.

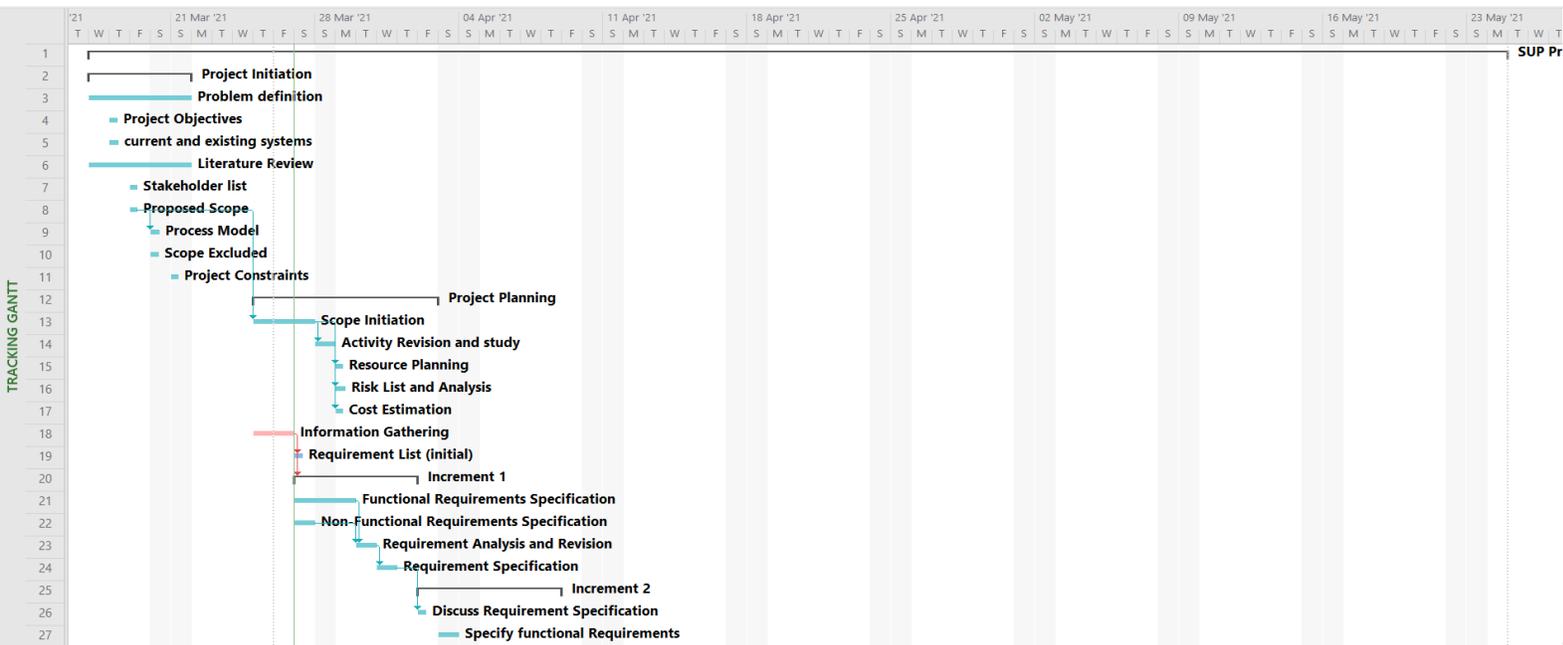


Figure 5- Gantt 1

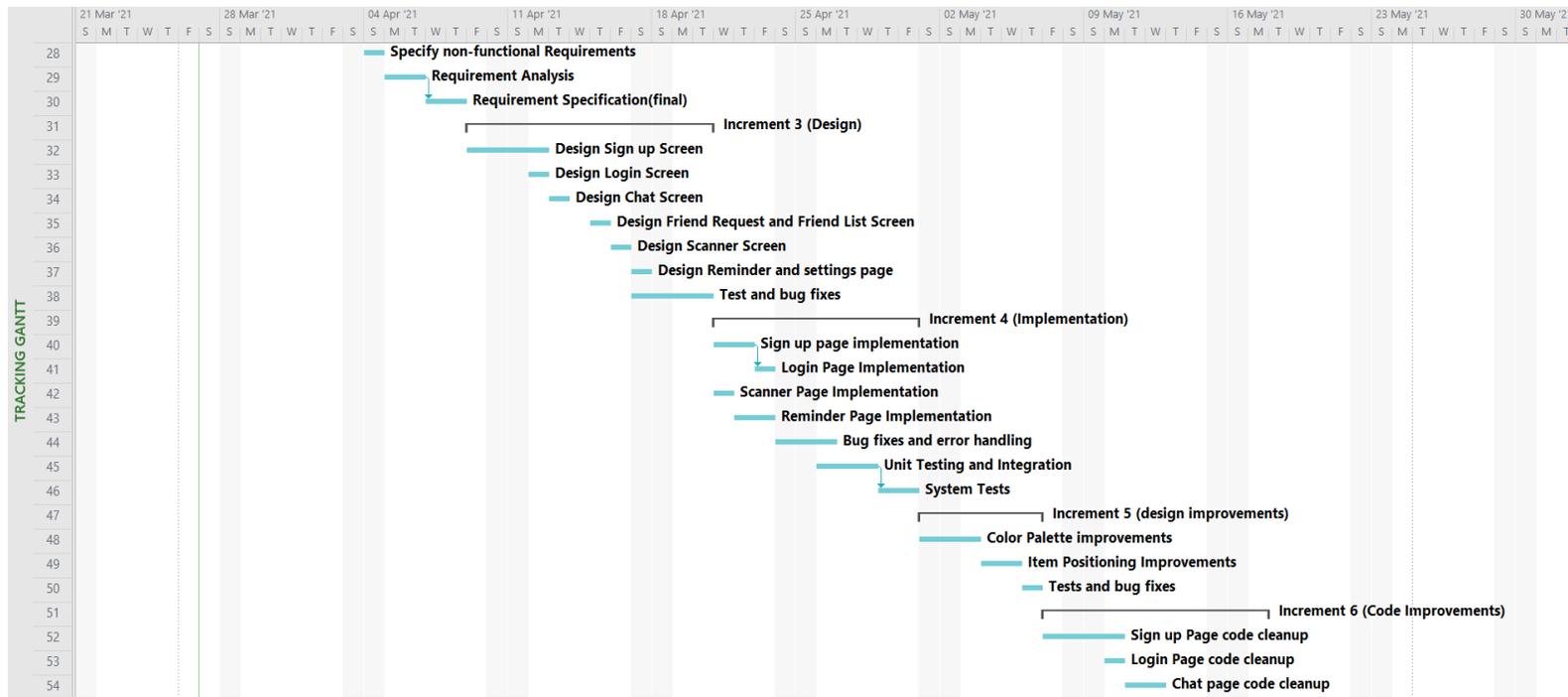


Figure 6- Gantt 2

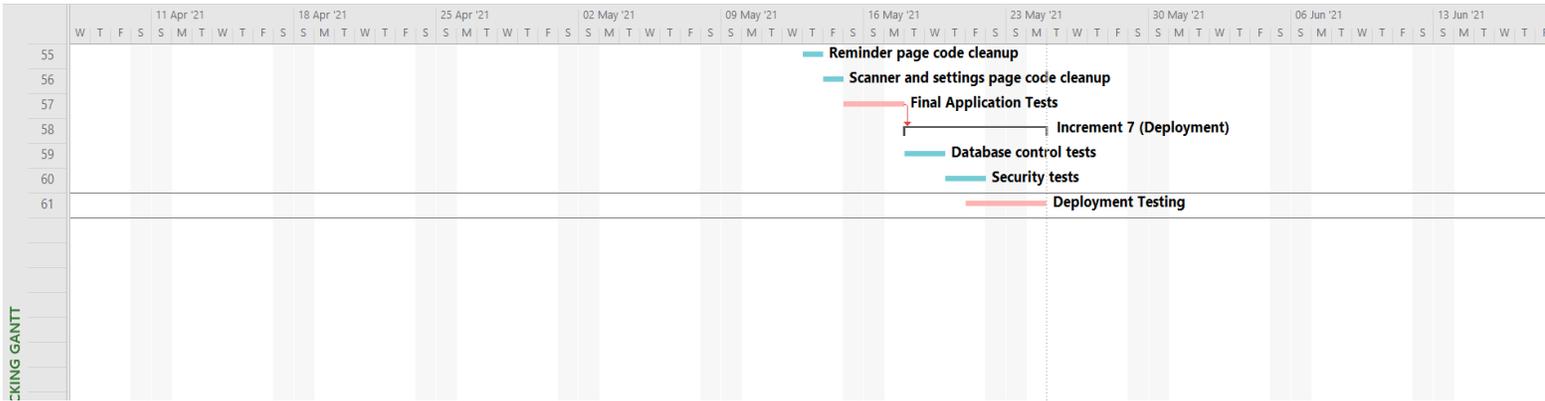


Figure 7- Gantt 3

### 2.1.3 Resource Sheet

Resource Name	Resource Type	Cost
Developer 1	Work	500\$/month
Developer 2	Work	500\$/hour
Android Studio V.4.1	Material	Free
Microsoft Word	Material	Free
Microsoft Visio	Material	Free
Microsoft Project	Material	10\$ (monthly subscription)
Google Firebase	Material	Free
Laptop	Material	Free
Smartphone (Android based)	Material	Free

Table 1- Resource Sheet

### 2.1.4 System Development Requirements

The following table will demonstrate the resources that will be required throughout the development process of this project, including Human Resources, Software, and Hardware, in addition to a cost overview of the project.

Resource Type	Resources
Human Resources	<ul style="list-style-type: none"><li>• Wael Khaled</li><li>• Mohammad alfaqeeh</li></ul>
Software	<ul style="list-style-type: none"><li>• Android Studio V.4.1</li><li>• Google Firebase</li><li>• Microsoft Word</li><li>• Microsoft Project</li><li>• Microsoft Visio</li></ul>
Hardware	<ul style="list-style-type: none"><li>• Laptop</li><li>• Smartphone (Android based)</li></ul>

Table 2- System Development Requirements

### 2.1.5 Cost Estimation and Budgeting

The following table will contain cost estimations for the different resources that will be used in the development process of this project:

Human Resources			
	Name	Monthly Cost	Entire Project
1	Wael Khaled	500\$	1,500\$
2	Mohammad Alfaqeeh	500\$	1,500\$
	<b>Total Cost</b>	<b>1,000\$</b>	<b>3,000\$</b>

Materials		
	Name	Cost
1	Android Studio V.4.1	Free
2	Google Firebase	Free
3	Microsoft Word	Free
4	Microsoft Project	10\$/month
5	Microsoft Visio	Free
6	Laptop	Free
7	Smartphone (Android Based)	Free
	<b>Total Cost</b>	<b>30\$</b>
	<b>Net Total</b>	<b>3,030\$</b>

Table 3- Cost Estimation

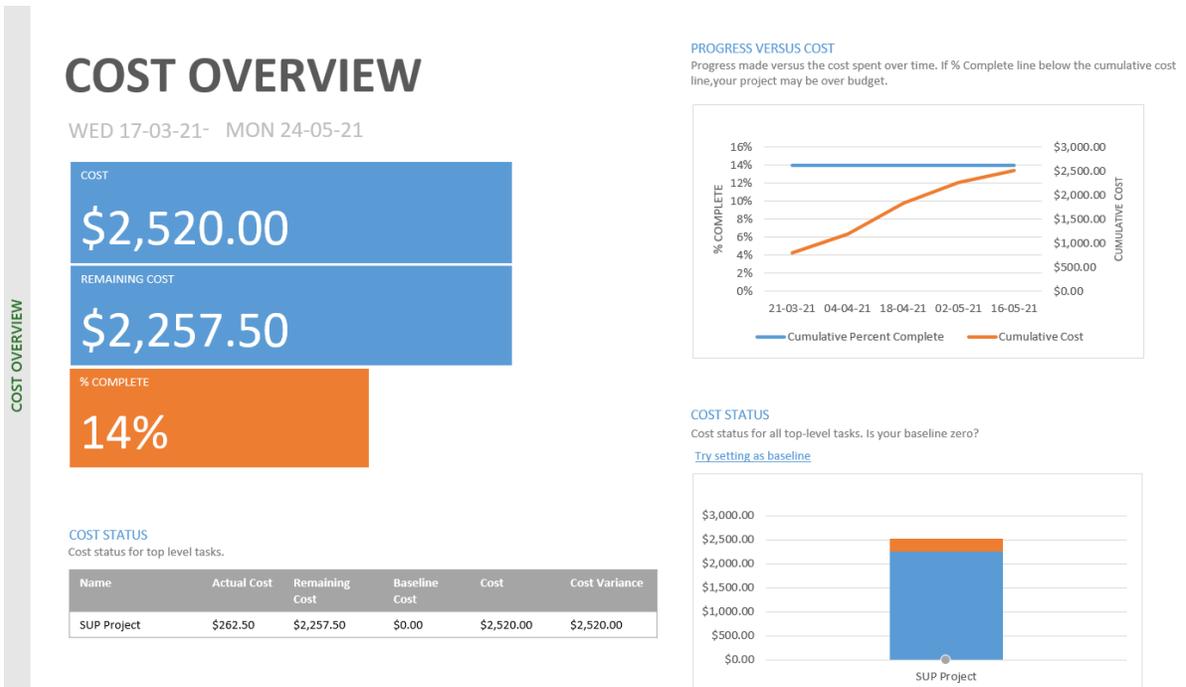


Figure 8- Cost Overview

### **2.1.6 Risk List**

1. There is a risk of not being able to keep up with the delivery dates.
2. Some advanced features may prove to be difficult to implement in the given timeframe.
3. Inaccurate cost estimation.
4. Requirements could evolve as the development process continues.
5. Application maintenance after delivery.
6. Lack of end-user testing and engagement due to health and safety measures.

## **2.2 Requirements**

### **2.2.1 Information Gathering**

#### **Brainstorming**

In this project, we decided to start information gathering with simple brainstorming sessions aiming to get to a better understanding of the problem we are set out to deal with in this project and the requirements that we need to implement during the development process, reaching a solid base of requirement specification that we could build on as we go.

Brainstorming sessions seemed like an appropriate method, especially because we as team members are students who are directly affected by the problem that our project is aiming to solve.

#### **Interviews**

Based on our belief that those affected by the problem are the most capable of describing it, multiple interviews were conducted with fellow students especially those who are not

majoring in IT-related topics, with the main goal of understanding the problems that they are facing with the current educational process.

These interviews were carried out using video conferencing platforms to try to interview as many students as possible while respecting the safety measures implemented and without endangering public safety.

### **2.2.2 Functional Requirements**

1. The system shall allow the user to create an account.
2. The system shall allow the user to sign in.
3. The system shall allow the user to sign out.
4. The system shall allow the user to chat with other users.
5. The system shall allow the user to send and receive friend requests.
6. The system shall allow the user to accept or reject friend requests.
7. The system shall allow the user to create reminders for exams, project submission deadlines, or homework.
8. The system shall allow the user to convert images to PDF files.
9. The system shall allow the user to edit his/her profile information.

### **2.2.3 Non-functional Requirements**

1. The system shall be available 80% (Availability).
2. The system shall be accurate 95% in the process of fetching, querying, and adding data to the database. (Accuracy).
3. The user chats shall be encrypted (Security).
4. The user information shall be secure (Security).
5. The system shall be easy to use by first-time users (Usability).
6. The system shall be able to handle wrong user inputs (Robustness).

7. The system should not any form of delay in sending or receiving data from the database (Performance).

## 2.3 Use cases

### 2.3.1 User Signup

When the user opens the application for the first time, he/she must create an account so they can log in with it and use the different functions offered by the application.

The user enters his/her personal information (name, date of birth, profile picture) in addition to an email and password, then the system will verify the information entered and create the user account.

<b>Use Case Name</b>	User Signup
<b>Preconditions</b>	-
<b>Actors</b>	Users, System
<b>Flow of Events</b>	<ol style="list-style-type: none"><li>1. The user clicks on the signup page.</li><li>2. The user enters his/her data (name, date of birth, profile picture, email, and password).</li><li>3. The user clicks the signup button.</li><li>4. The system verifies the data entered (no empty fields, the password is not too short, the email format is correct and the email does not already exist in the database).</li><li>5. The system creates the account.</li><li>6. The system redirects the user to the sign-in page.</li></ol>

<b>Alternatives</b>	<ol style="list-style-type: none"> <li>1. The user chooses the “sign up via Google” option.</li> <li>2. The user picks the Google account he/she wants to use.</li> <li>3. The user is redirected to the signup page with some fields already filled.</li> <li>4. The user clicks the signup button.</li> <li>5. The system creates the account.</li> </ol>
<b>Post Conditions</b>	The system creates the user account and displays the login page.
<b>Exceptions</b>	E2.1: Wrong data entry (missing fields, short password, incorrect email format).

*Table 4- User Signup Use case*

### 2.3.2 User Sign-in

After the user creates an account successfully, he/she can sign in using the email and password they provided and if the data entered matches the data stored in the database, they will be redirected to the application’s homepage, if not, they will be asked to retry.

<b>Use Case Name</b>	User Sign-in
<b>Precondition</b>	The user already has an account.
<b>Actors</b>	Users, System
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. The system displays the sign-in page.</li> <li>2. The user enters his/her email and password.</li> </ol>

	<ol style="list-style-type: none"> <li>3. The user clicks the sign-in button.</li> <li>4. The system checks if the email and password entered exist and match the data in the database and if correct, the system will redirect the user to the homepage.</li> </ol>
<b>Alternatives</b>	-
<b>Post Condition</b>	The system greets the user and transfers him/her to the homepage.
<b>Exceptions</b>	E2.1: The email or password entered is incorrect or does not match the database.

Table 5- User Sign-in Use Case

### 2.3.3 Send Friend Requests

If the user wants to communicate with his/her friends using the chatting function, they must send them a friend request, where the user opens the friend search tab and types the username of the person in the search bar, and if the system verifies that the username exists in the database, the user would be able to send them a friend request.

<b>Use Case Name</b>	Send Friend Requests
<b>Precondition</b>	<p>The user already has an account.</p> <p>The user must be signed in.</p>
<b>Actors</b>	Users, System
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. The user selects the friend search tab found on the homepage.</li> <li>2. The system displays the friend search tab.</li> <li>3. The user types the username that he/she is looking for in the search bar.</li> </ol>

	<ol style="list-style-type: none"> <li>4. The system checks if the username exists in the database and if it does, the system displays it to the user.</li> <li>5. The user clicks the “send friend request” button.</li> </ol>
<b>Alternatives</b>	-
<b>Post Condition</b>	A friend request will be sent to the specified user.
<b>Exceptions</b>	E3.1: The username entered does not exist in the database.

Table 6- Send Friend Request Use Case

### 2.3.4 Receive Friend Requests

The friend requests sent to the user will be displayed in the friend requests tab, where the user can accept or reject the friend requests he/she received, if the user accepts the request it will disappear from the request tab and the user will be added to the friends' list, in the other hand if the user rejects it, it will simply be removed from the request tab.

<b>Use Case Name</b>	Receive Friend Request
<b>Precondition</b>	The user already has an account.
<b>Actors</b>	Users, System
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. The user selects the friend requests tab found on the homepage.</li> <li>2. The system displays the page which contains the list of requests received.</li> <li>3. The user clicks the “accept” button.</li> <li>4. The system removes the request and adds the user to the friend list.</li> </ol>
<b>Alternatives</b>	<p>If the user selects the “reject” button</p> <p>The system will delete the request.</p>

<b>Post Condition</b>	The system will remove the friend request and add the user to the friend list (if the request is accepted)  The system will remove the friend request (if the request is rejected)
<b>Exceptions</b>	-

*Table 7- Receive Friend Request Use Case*

### 2.3.5 User chat

The user can chat with his/her friends that were added previously, to chat for the first time the user clicks on the friend that he/she wants to talk with from the friend list tab, then the system will automatically redirect them to the chat page where the user could start sending messages, after that the user can directly access recent chats from the homepage.

<b>Use Case Name</b>	User Chat
<b>Precondition</b>	The user must be signed in. The user must be added to the friend list.
<b>Actors</b>	Users, System
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. The user selects the friend list tab from the homepage.</li> <li>2. The system displays the friend list page.</li> <li>3. The user selects a friend to chat with from the list.</li> <li>4. The system redirects the user to the chat page.</li> <li>5. The user sends a message to his/her friend.</li> </ol>
<b>Alternatives</b>	-
<b>Post Condition</b>	The chat will be initiated and the message will be sent.

<b>Exceptions</b>	E5.1: The message could not be sent due to a connection error.
-------------------	----------------------------------------------------------------

Table 8- User Chat Use Case

### 2.3.6 Add Reminder

The user can add reminders for upcoming exams, homework, or project submissions, then the system will send notifications reminding the user of the activity needed to be done.

The user can create reminders by clicking the “Add reminder” button on the reminder page, then he/she will be asked to enter some information (name of reminder, type of reminder, due date) and when the user is done, the system will create the reminder.

<b>Use Case Name</b>	Add Reminder
<b>Precondition</b>	The user must be signed in.
<b>Actors</b>	Users, System
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. The user selects the reminder page.</li> <li>2. The system displays the reminder page.</li> <li>3. The user clicks the “Add Reminder” button.</li> <li>4. The system displays a list of data fields that the user needs to fill (reminder name, reminder type, due date).</li> <li>5. The user clicks the “create reminder” button.</li> <li>6. The system verifies the data (no empty fields) then creates the reminder</li> </ol>
<b>Alternatives</b>	-

<b>Post Condition</b>	The system creates a reminder
<b>Exceptions</b>	E4.1: The user forgets to fill some required fields.

Table 9- Add Reminder Use Case

### 2.3.7 Edit User Information

The user will be able to edit the information he/she provided when first creating the account, the user selects the “edit profile” option and the system displays a page containing the information stored in the database.

The user edits the information, then submits the changes and the system commits the updated information to the database.

<b>Use Case Name</b>	Edit User Information
<b>Precondition</b>	The user must be signed in.
<b>Actors</b>	Users, System
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. The user selects the “edit profile information” option.</li> <li>2. The system displays the edit profile information page.</li> <li>3. The user enters the updated information.</li> <li>4. The user clicks the “submit changes” button.</li> <li>5. The system verifies the information entered (no missing fields or incompatible data type) then commits the changes to the database.</li> </ol>
<b>Alternatives</b>	-

<b>Post Condition</b>	The system commits the changes to the database.
<b>Exceptions</b>	E3.1: empty fields or incompatible data type.

Table 10- Edit User Information Use Case

### 2.3.8 Convert Images

The user should be able to convert images saved on his/her phone's gallery to PDF files, the user navigates to the image scan page.

The system will show a popup asking for permission to access the phone gallery, if access is granted, the user will be asked to choose an image to convert, after selection, the system shows a preview of the conversion result and allows the user to modify the scanning parameters, and after the converting process is done the system will save the converted file in the phone's storage.

<b>Use Case Name</b>	Image Scanner
<b>Precondition</b>	The user must be signed in. The user must permit to access the gallery.
<b>Actors</b>	Users, System
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. The user navigates to the image scanner page.</li> <li>2. The system displays a popup asking for permission to access the gallery (on first-time use only).</li> <li>3. The user selects the image to convert.</li> <li>4. The system shows a preview of the converted image.</li> <li>5. The user modifies the scanning parameters (if needed).</li> <li>6. The user clicks the convert button.</li> </ol>

	7. The system converts the image and saves the file on the phone's storage.
<b>Alternatives</b>	-
<b>Post Condition</b>	The system saves the converted file on the phone's storage.
<b>Exceptions</b>	E2.1: permission not granted.

Table 11- Convert Image Use Case

### 2.3.9 User sign-out

The user shall be allowed to sign out of his/her account easily by clicking on the sign-out button from the homepage.

After signing out the system will redirect the user to the sign-in page where he/she can sign in again using another account.

<b>Use Case Name</b>	Sign-out
<b>Precondition</b>	The user must be signed in.
<b>Actors</b>	Users, System
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. The user clicks on the sign-out button from the homepage.</li> <li>2. The system ends the user's current session.</li> <li>3. The system redirects the user to the sign-in page.</li> </ol>
<b>Alternatives</b>	-
<b>Post Condition</b>	The user is signed out of his/her account.
<b>Exceptions</b>	-

Table 12- Sign-out Use Case

## 2.4 Domain Diagram

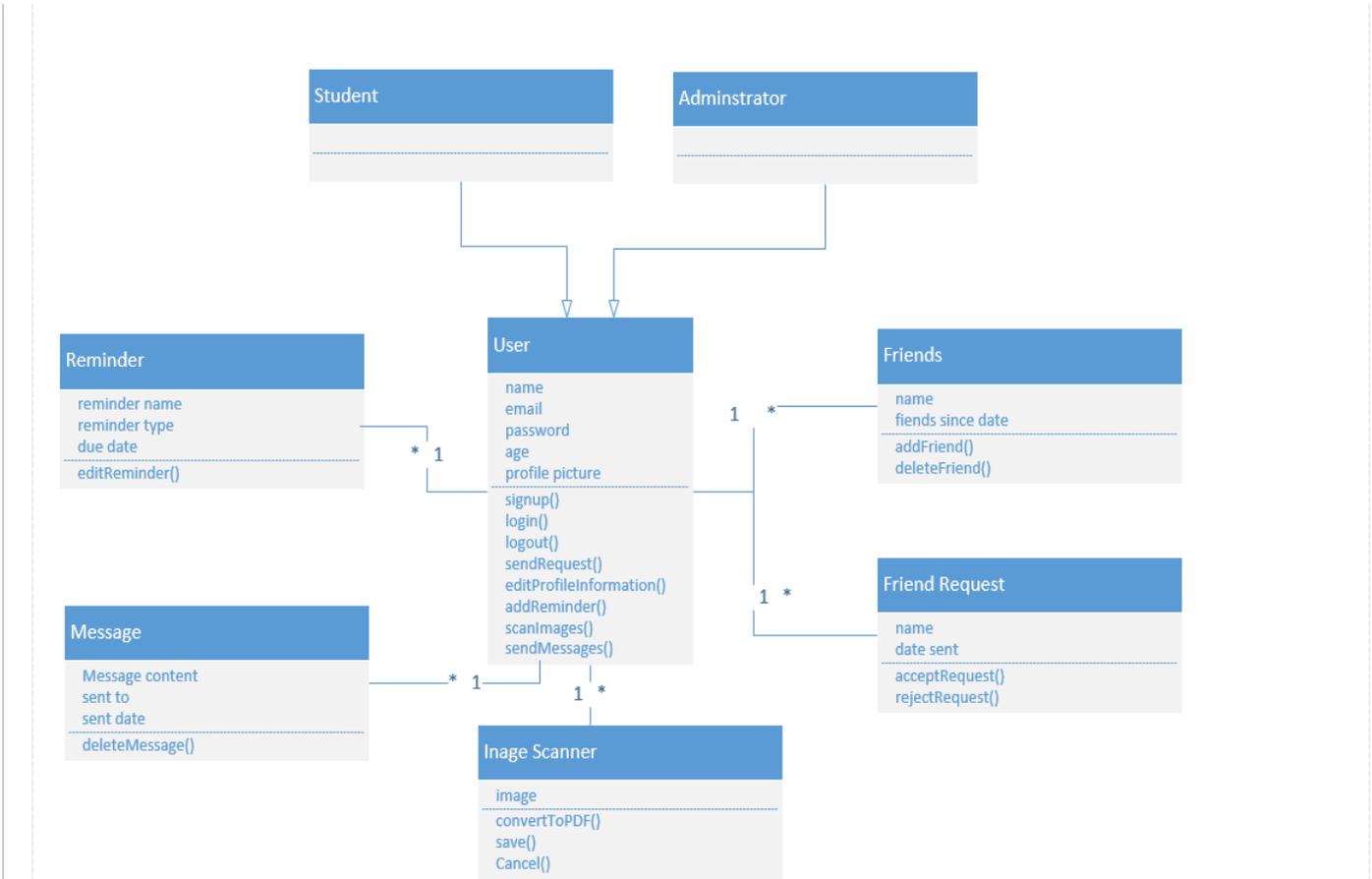


Figure 9- Domain Diagram