Group 4 – Communications System

Software Requirements Specification

Revision History

Date	Revision	Description	Author
02/19/2022	1.0	Initial Version	Owen Casebeer
02/20/2022	1.1	Added Requirments	Anthony Lopez
02/20/2022	1.1.2	Updated Overall Description	Matthew Baron
02/21/2022	1.1.3	Updated non-functional requirements	Owen Casebeer
02/21/2022	1.1.4	Added Sequential Diagrams	Anthony Lopez
2/25/2022	1.1.5	Added internal/external reqs	Owen Casebeer
2/25/2022	1.1.6	Added Class Diagrams	Matthew Baron

Table of Contents

1. PURPOSE 4

1.1. SCOPE 4
1.2. DEFINITIONS, ACRONYMS, ABBREVIATIONS 4
1.3. REFERENCES 4
1.4. OVERVIEW 4

2. OVERALL DESCRIPTION 5

2.1. PRODUCT PERSPECTIVE 5
2.2. PRODUCT ARCHITECTURE 5
2.3. PRODUCT FUNCTIONALITY/FEATURES 5
2.4. CONSTRAINTS 5
2.5. ASSUMPTIONS AND DEPENDENCIES 5

3. SPECIFIC REQUIREMENTS 6

3.1. FUNCTIONAL REQUIREMENTS 6
3.2. EXTERNAL INTERFACE REQUIREMENTS 6
3.3. INTERNAL INTERFACE REQUIREMENTS 7

4. NON-FUNCTIONAL REQUIREMENTS 8

4.1. SECURITY AND PRIVACY REQUIREMENTS 8
4.2. ENVIRONMENTAL REQUIREMENTS 8
4.3. Performance Requirements 8

1. Purpose

1.1. Scope

This document will catalog the user, system, and hardware requirements for the Communications System. It will not however, document how these requirements will be implemented.

1.2. Definitions, Acronyms, Abbreviations

CS – Communications System

1.3. References

Use Case ID: 1000U

Use Case Name: Create a chat

Relevant Requirements: 3.1.2

Primary Actor: User

Pre-conditions: User is logged in.

Post-conditions: Chat is created.

Basic Flow or Main Scenario:

1. The user selects participants from the user directory to create a chat.
2. The system responds by creating a chatroom on the server.

Extensions or Alternate Flows:

1. The user creates an empty chatroom.
2. The user invites users from the user directory to join the room.

Exceptions: None.

Related Use Cases: 2000U

Use Case ID: 2000U

Use Case Name: Send a message

Relevant Requirements: 3.1.2

Primary Actor: User

Pre-conditions: User is logged in.

Post-conditions: The message is delivered to the chat.

Basic Flow or Main Scenario:

1. The user opens an existing chatroom.
2. The user inputs the desired message and sends it.
3. The server receives the message and delivers it to the chat's participants.

Extensions or Alternate Flows: None.

Exceptions: The message fails to send.

Related Use Cases: 1000U

Use Case ID: 3000U

Use Case Name: Access chat history

Relevant Requirements: 3.1.1

Primary Actor: Supervisor

Pre-conditions: Supervisor is logged in.

Post-conditions: The supervisor is granted access to all chat logs.

Basic Flow or Main Scenario:

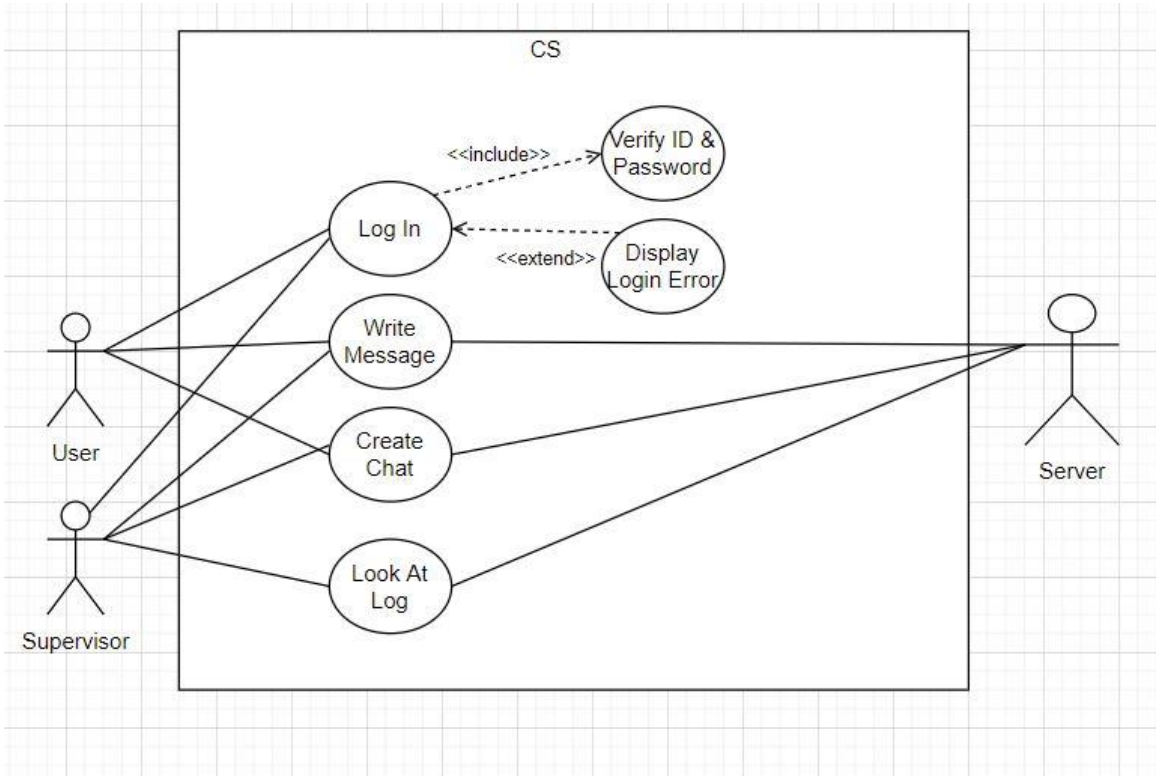
1. The supervisor attempts to access user chat logs.
2. The system responds by granting a directory of logs.

Extensions or Alternate Flows: None.

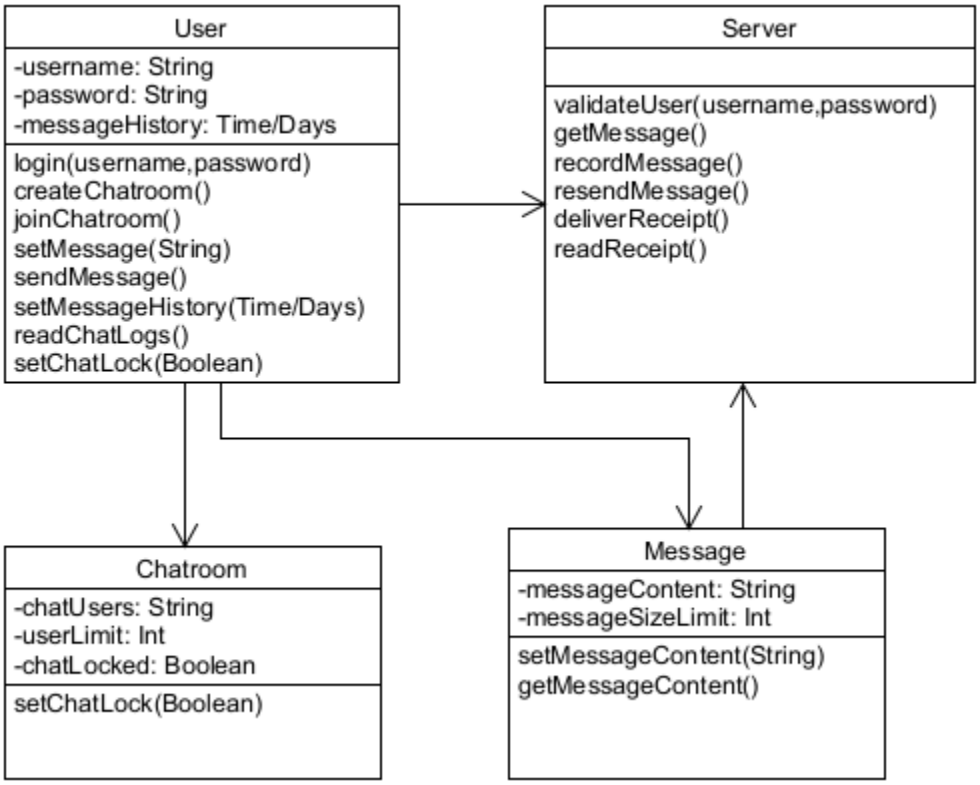
Exceptions: None.

Related Use Cases: None.

Use Case Diagram

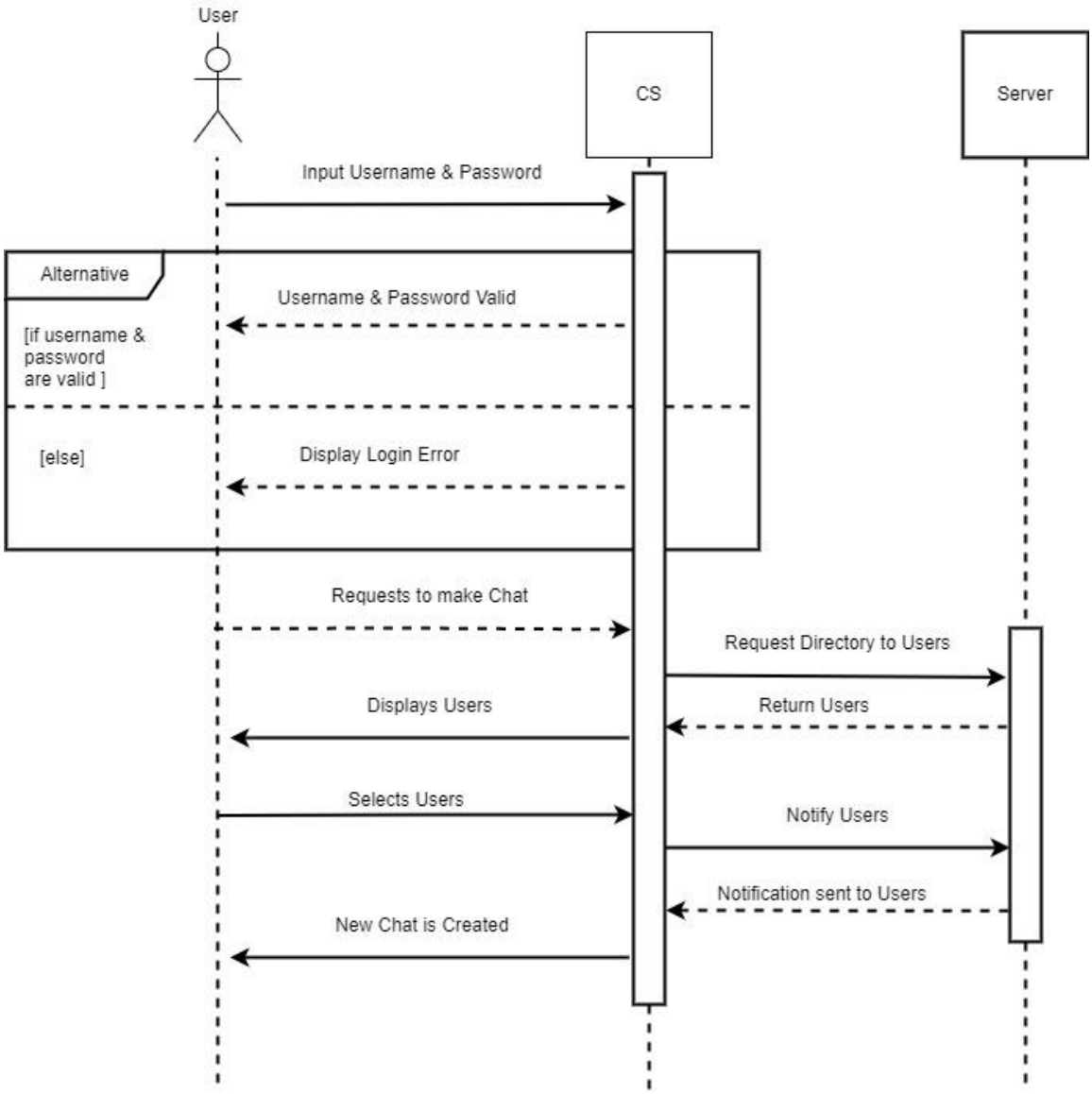


Class Diagrams

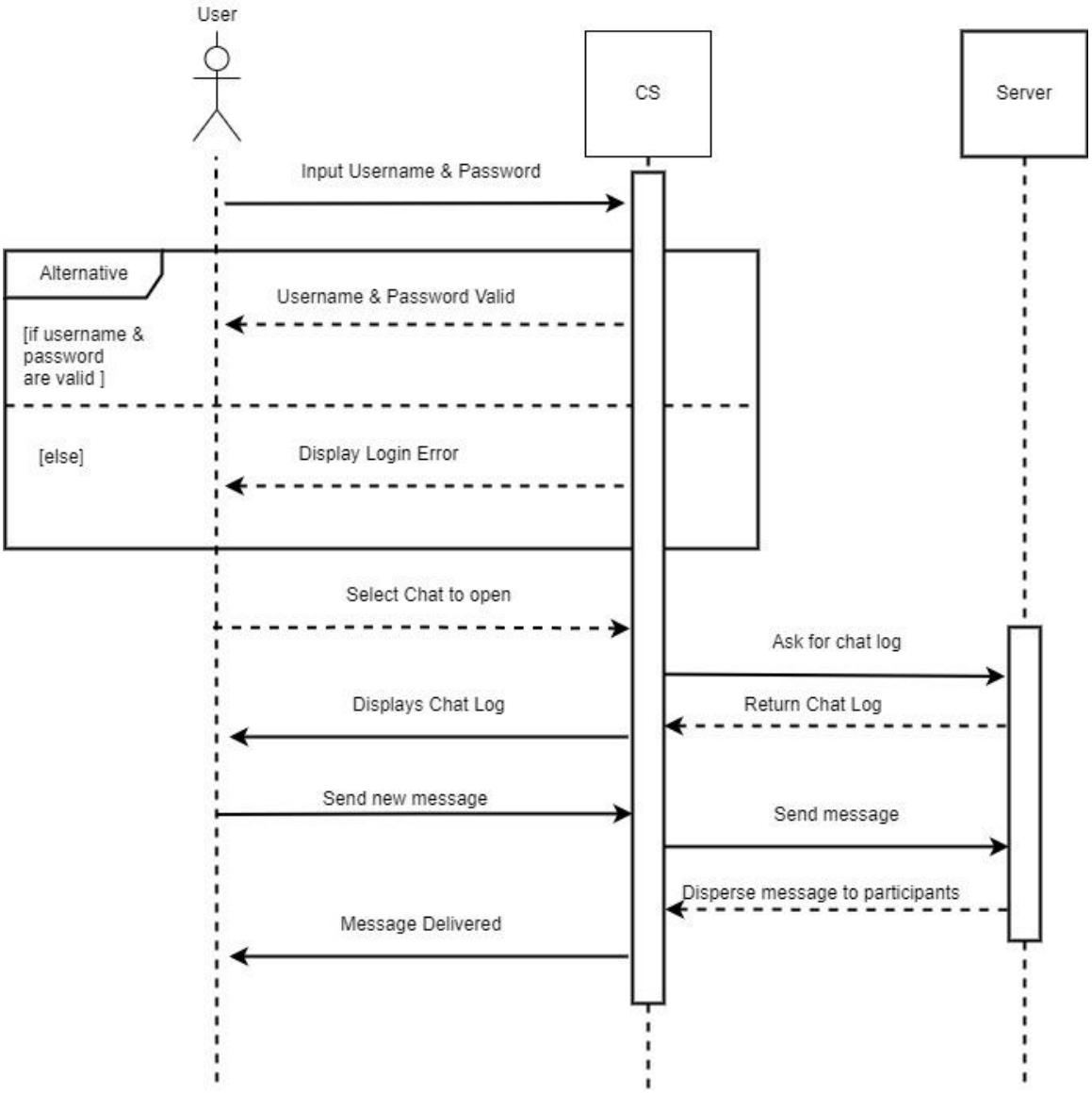


Sequential Diagrams

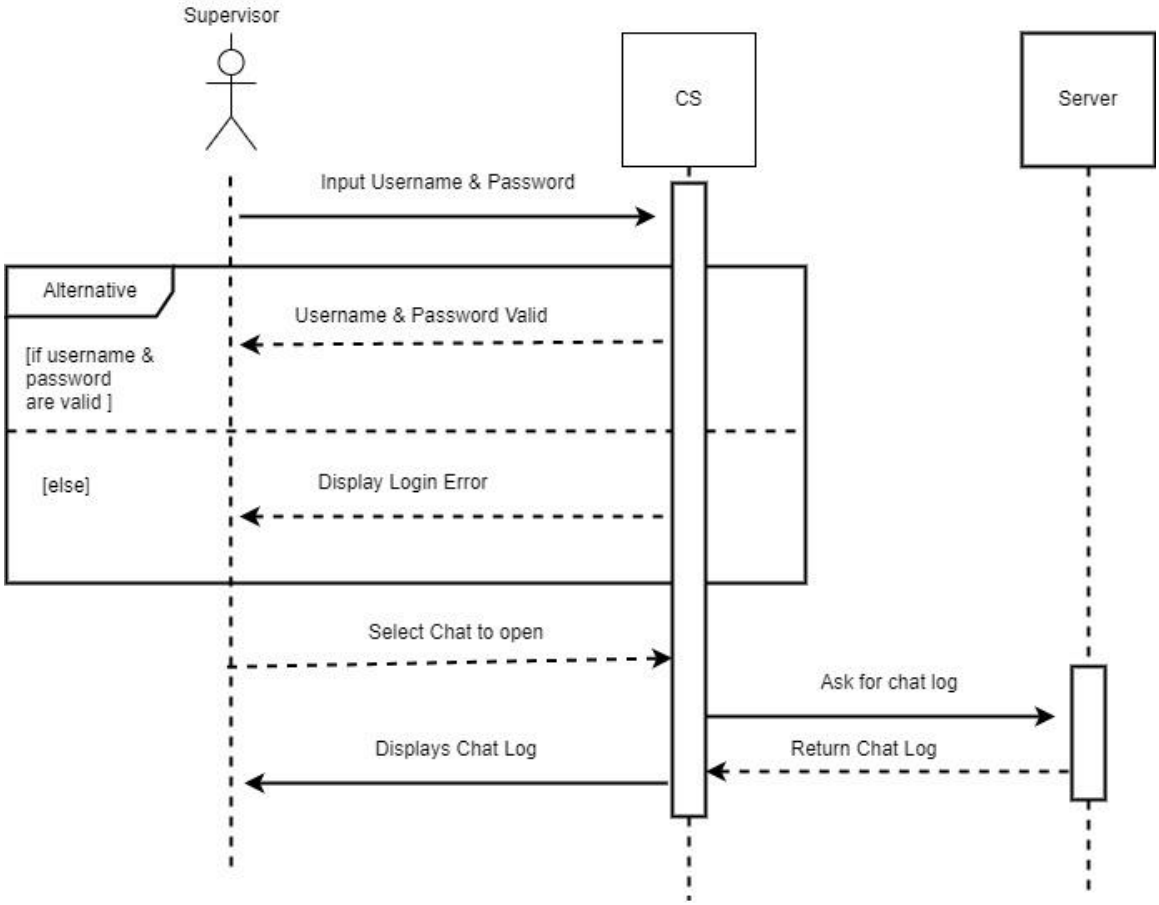
Sequential Diagram Use Case #1000



Sequential Diagram Use Case #2000



Sequential Diagram Use Case #3000



1.4. Overview

The Communications System (CS) is designed to interconnect corporate communications nationwide through a text-based messaging system.

2. Overall Description

2.1. Product Perspective

2.2. Product Architecture

The system will be organized into 2 major modules: the client module and the server module.

2.3. Product Functionality/Features

The high-level features of the system are as follows:

- All employees have a unique login
- One-on-one and group chats
- Strictly text-based
- Asynchronous messaging
- All chatrooms are logged
- Delivery/Read receipts

2.4. Constraints

2.4.1 The project must be completed by 5/4/2022.

2.5. Assumptions and Dependencies

No current assumptions.

3. Specific Requirements

3.1. Functional Requirements

3.1.1. Common Requirements:

3.1.1.1 All chats are recorded and stored in a database.

3.1.2. Client Module Requirements:

3.1.2.1 Users should be allowed to log in using their issued ID and password.

3.1.2.2 Users should be allowed to create chats/chatrooms.

3.1.2.3 Users should be allowed enter and leave chatrooms as they please.

3.1.2.4 Chatrooms should have a participants list and security settings to limit who can join the chat.

3.1.2.5 Users should be able to set how long messages are stored before they are automatically deleted from their personal chat history.

3.1.2.6 New messages will be labeled as read if the chat is opened.

3.1.2.7 Users should have access to a directory of users.

3.1.3. Server Module Requirements:

3.1.3.1 Messages are asynchronous.

3.1.3.2 Messages will be labeled as delivered if successfully sent.

3.2. External Interface Requirements

3.2.1 The system must provide an GUI to allow users to access communications features.

3.3. Internal Interface Requirements

3.3.1 The system must process outgoing messages from the client to ensure messages are delivered to all applicable parties.
3.3.2 The system must process requests for past chat logs from users with the appropriate access level.

4. Non-Functional Requirements

4.1. Security and Privacy Requirements

4.1.1 The CS may only be accessed with a successful log in with a username and password.
4.1.2 The CS will be stored on a private corporate server.

4.2. Environmental Requirements

4.2.1 The CS will be built with the Java programming language.

4.3. Performance Requirements

There are no Performance Requirements for this project

Group 4 Minutes

1/26 @ 4:25pm

Introduction Client Meeting:

Attendance: Owen, Maxwell, Matty, Brianna, Anthony

Meeting Time: 5 minutes

1. Internal business communication network
2. >=2 people per chat (one-on-one and group messaging)
3. Must work across multiple offices
4. All communications are stored

1/31 @ 4:20pm

Extended Requirement Client Meeting

Attendance: Owen, Maxwell, Matty, Anthony

Meeting Time: 20 minutes

1. Text only chats
2. Users have usernames and passwords – accounts are assigned by employer
3. Messaging is asynchronous
4. Records are kept for all chats
5. 2-part system: server and client
6. All users should have access to a directory of all other users

Optional/time permitting – profile pictures/avatars, broadcasts

2/9 @ 4:00pm

Q&A Client Meeting

Attendance: Owen, Maxwell, Matty, Anthony

Meeting Time: 5 minutes

1. There should be a settings GUI which allows users to decide how long messages are stored in their personal "copy" of the chat
2. Rooms should have a participant list
3. There should be both system and user delivery results (ex. Message successfully sent, message read by another participant)

2/19 @ 9:00am

Developer Meeting To Create SRS Document

Attendance: Owen, Maxwell, Matty, Brianna, Anthony

Meeting Time: 2 hours

1. Finished Part 1 as a group
2. Plan to upload all documents to GitHub before Sunday night

2/21 @ 3:34pm

Developer meeting to discuss last minute changes before Phase 1 deadline

Attendance: Owen, Maxwell, Matty, Anthony

Meeting Time: 26 minutes

1. Finalized non-functional requirements
2. Discussed questions for client meeting

2/21 @ 4:09pm

Client Meeting

Attendance: Owen, Maxwell, Matty, Anthony

Meeting Time: 5 minutes

1. Clarified due date for SRS document
2. Clarified the purpose of/what belongs in the project schedule

2/25 @ 9:00am

Client Meeting

Attendance: Owen, Matty, Anthony

Meeting Time: 2 hours

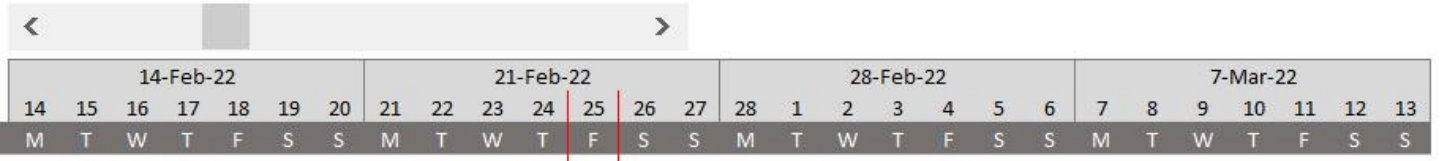
1. Updated information for SRS document.
2. Began preparation for presentation.

PROJECT START: 1/26/2022
Display Week: 1

TASK	START	END
Requirements	1/26/2022	2/14/2022
SRS Documentation	2/7/2022	2/14/2022
Design	2/15/2022	2/28/2022
Implementation	3/1/2022	4/4/2022
Testing	4/5/2022	5/3/2022
Finished Product	5/4/2022	5/4/2022

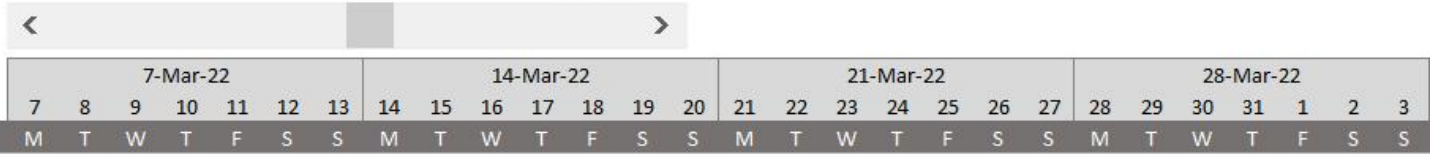
PROJECT START: 1/26/2022
Display Week: 4

TASK	START	END
Requirements	1/26/2022	2/14/2022
SRS Documentation	2/7/2022	2/14/2022
Design	2/15/2022	2/28/2022
Implementation	3/1/2022	4/4/2022
Testing	4/5/2022	5/3/2022
Finished Product	5/4/2022	5/4/2022



PROJECT START: 1/26/2022
Display Week: 7

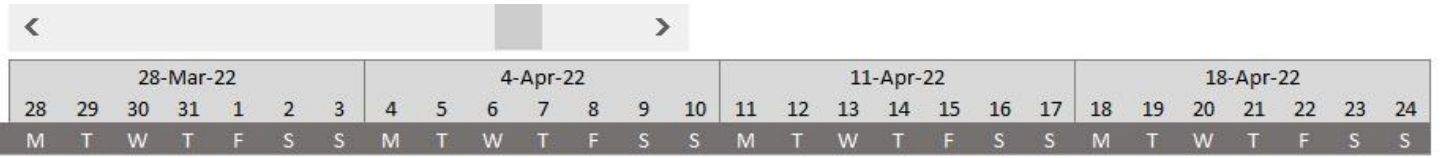
TASK	START	END
Requirements	1/26/2022	2/14/2022
SRS Documentation	2/7/2022	2/14/2022
Design	2/15/2022	2/28/2022
Implementation	3/1/2022	4/4/2022
Testing	4/5/2022	5/3/2022
Finished Product	5/4/2022	5/4/2022



PROJECT START: 1/26/2022

Display Week: 10

TASK	START	END
Requirements	1/26/2022	2/14/2022
SRS Documentation	2/7/2022	2/14/2022
Design	2/15/2022	2/28/2022
Implementation	3/1/2022	4/4/2022
Testing	4/5/2022	5/3/2022
Finished Product	5/4/2022	5/4/2022



PROJECT START: 1/26/2022

Display Week: 12

TASK	START	END
Requirements	1/26/2022	2/14/2022
SRS Documentation	2/7/2022	2/14/2022
Design	2/15/2022	2/28/2022
Implementation	3/1/2022	4/4/2022
Testing	4/5/2022	5/3/2022
Finished Product	5/4/2022	5/4/2022