|  |
| --- |
| CSC 340 Computer Ethics and Software Engineering |
| Meeting Coordinator |
| Final Report |

|  |
| --- |
| Aidan Barbour, Jeff Davidson, and John Holtzworth  4-20-2020 |

Contents

[I. Introduction 2](#_Toc38276841)

[A. Problem Statement 2](#_Toc38276842)

[B. Proposal 2](#_Toc38276843)

[II. System Description 2](#_Toc38276844)

[III. System Requirements 2](#_Toc38276845)

[A. Functional Requirements 2](#_Toc38276846)

[IV. Use Case Diagram 3](#_Toc38276847)

[V. Class Diagram 3](#_Toc38276848)

[VI. Sequence Diagram 3](#_Toc38276849)

[VII. Activity Diagram 8](#_Toc38276850)

[VIII. State Diagram 13](#_Toc38276851)

[IX. Database Design 14](#_Toc38276852)

[X. Conclusion 14](#_Toc38276853)

[XI. Data Dictionary 15](#_Toc38276854)

# Introduction

## Problem Statement

* + Multiple companies need to allow managers or team leaders to coordinate meetings with their subordinates

## Proposal

* + We propose a calendar system which will allows for managers to track events and add employees to the events

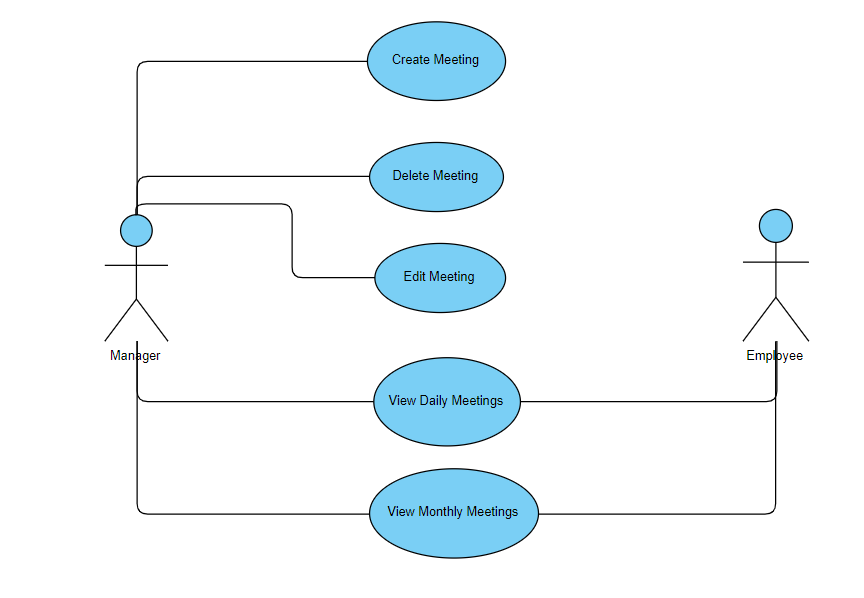
# System Description

* + The system shall provide the following services to the user:
* Add a meeting to the calendar of themselves and their employees
  + - Check employee availability
    - Check conference room availability
* Delete a meeting from the calendars of all attendees.
* Edit a meeting
* View a daily list of meetings
* View a monthly-based meeting list.

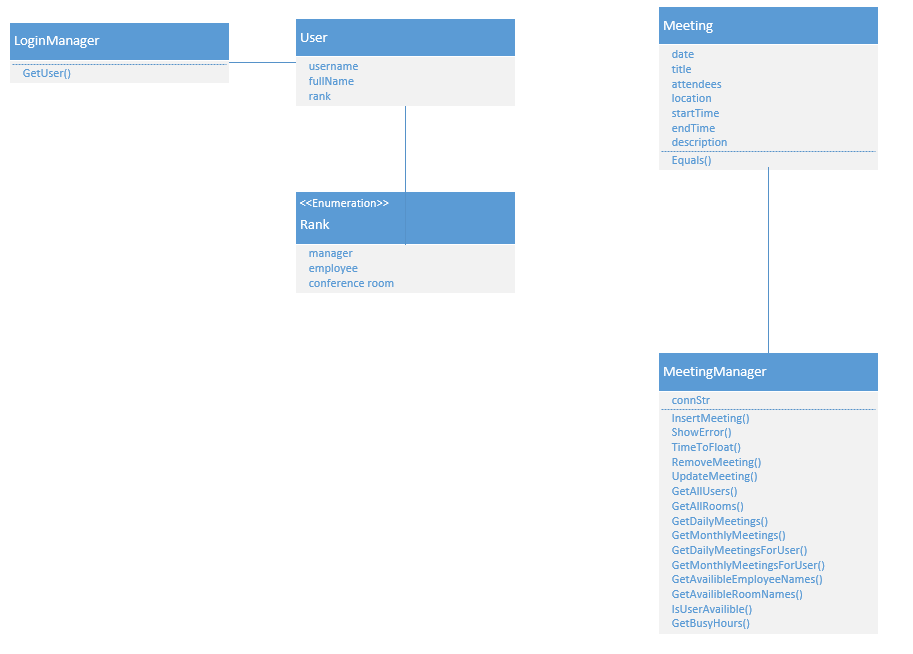
# System Requirements

## Functional Requirements

# Use Case Diagram



# Class Diagram

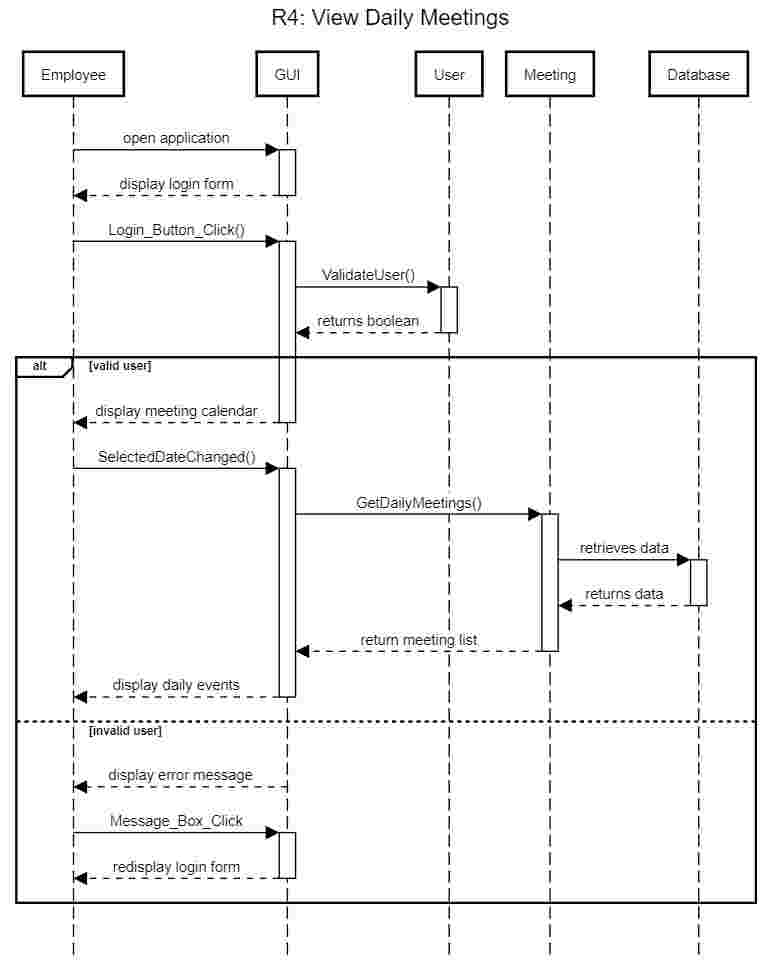


# Sequence Diagram

1. Add Meeting
   1. A screenshot of text

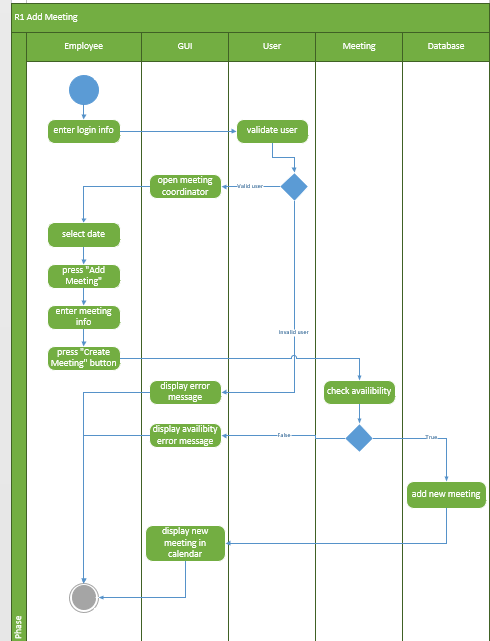
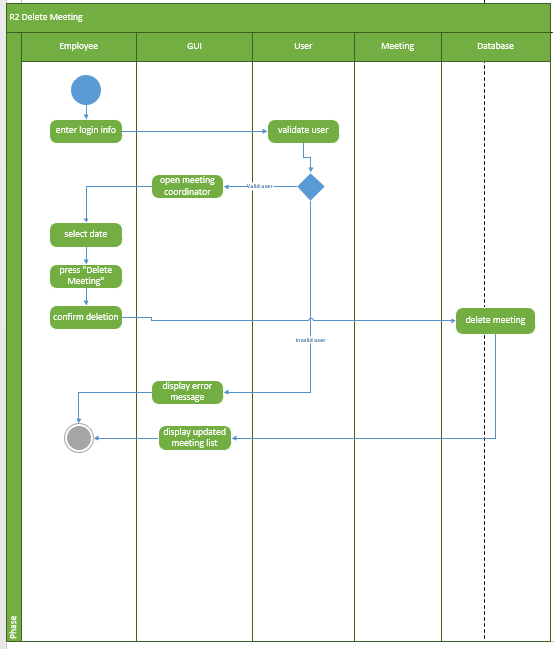
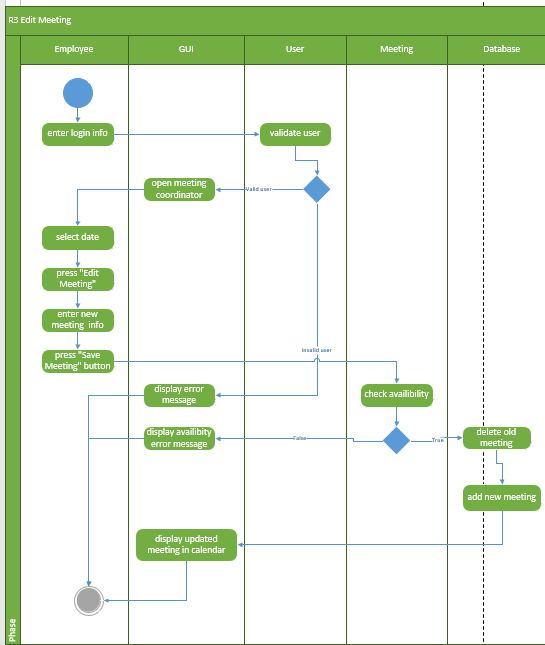
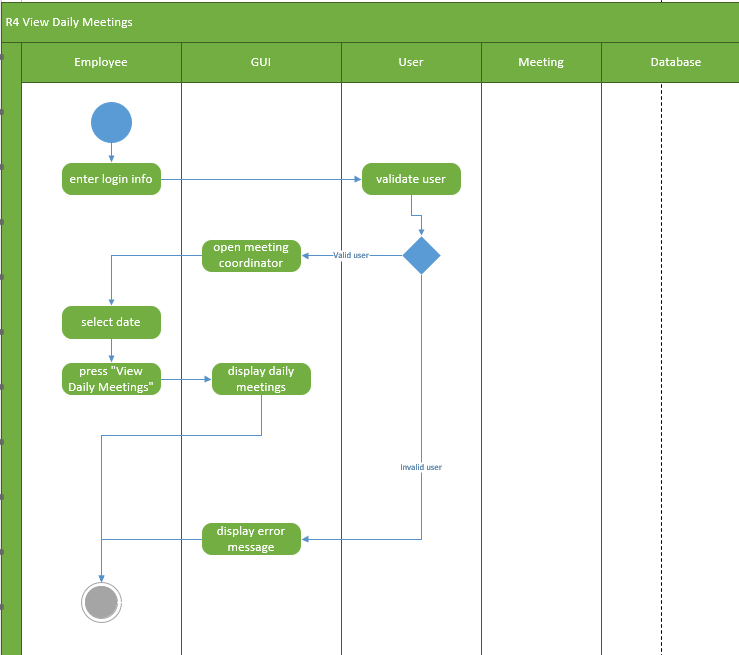
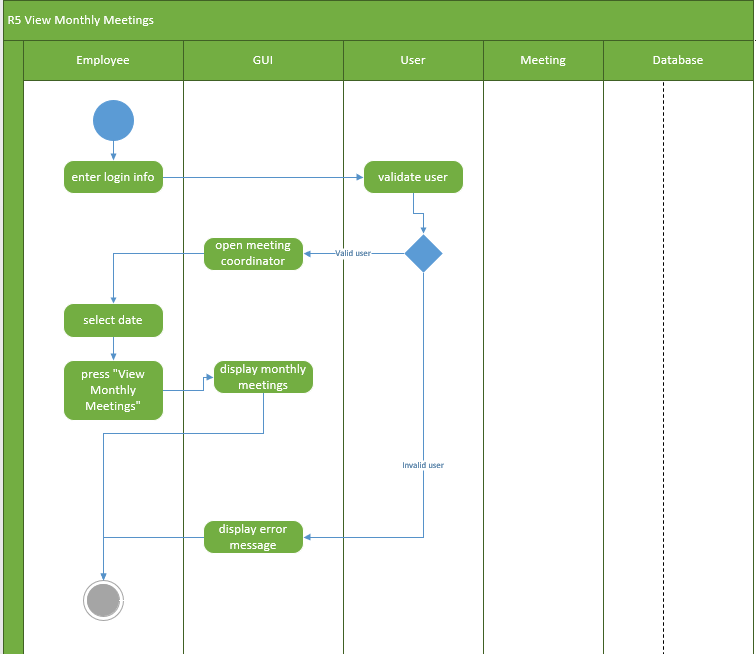
      Description automatically generated
2. Delete Meeting
   1. A close up of a piece of paper

      Description automatically generated
3. Edit Meeting
   1. A close up of a map

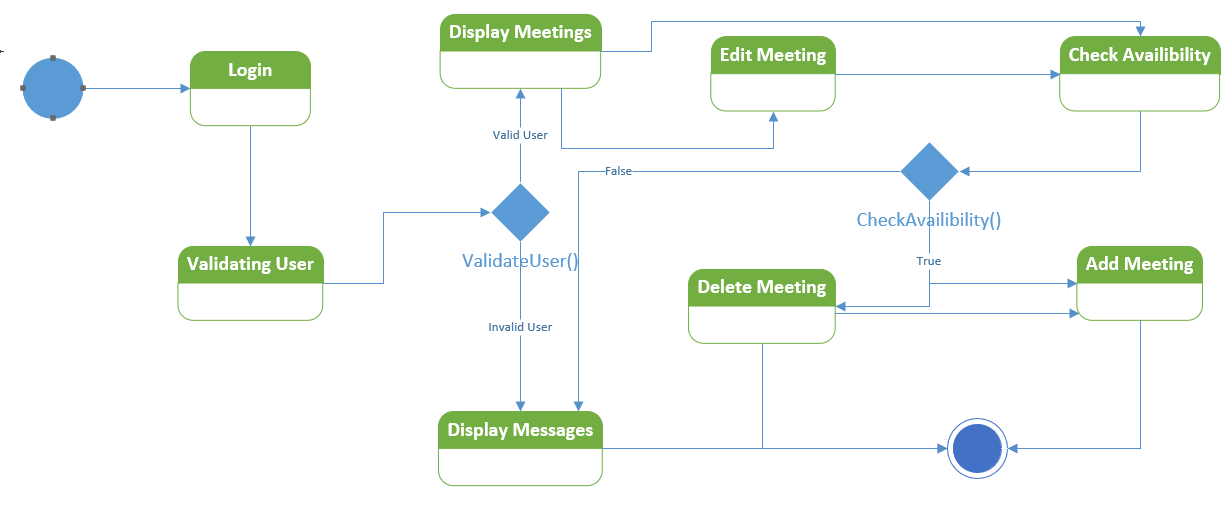
      Description automatically generated
4. View Daily Meetings
   1. 
5. View Monthly Meetings
   1. A close up of a map

      Description automatically generated

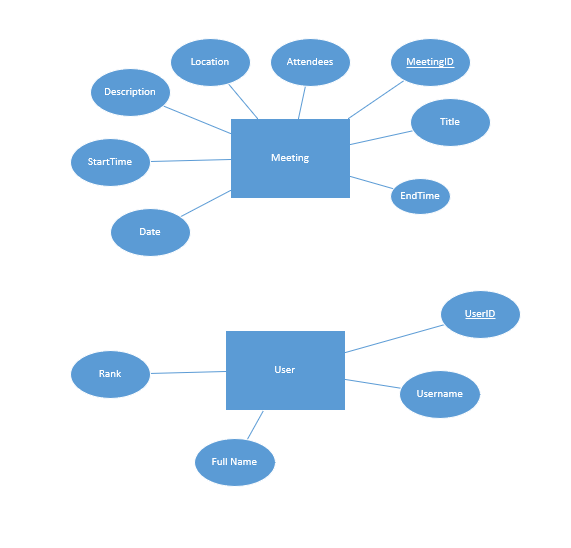
# Activity Diagram

1. Add Meeting
   1. 
2. Delete Meeting
   1. 
3. Edit Meeting
   1. 
4. View Daily Meetings
   1. 
5. View Monthly Meetings
   1. 

# State Diagram



# Database Design



Meeting

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| meetingID | date | title | startTime | endTime | location | attendees | description |

User

|  |  |  |  |
| --- | --- | --- | --- |
| userID | fullName | username | rank |

# Conclusion

* 1. We have developed a meeting coordination system which implements the required specifications. This will solve the personal and company problems as listed.
  2. We have included a functional requirements section which shows a step-by-step process of how each functional requirement of the system works.
  3. We have included a use case diagram which shows the relationship between the users and the functions that different users can perform.
  4. We have included a class diagram which describes the basic structure and operations of the meeting and event classes.
  5. We have included sequence diagrams for each functional requirement which show the chronological order of how the system processes information and fulfills a certain duty.
  6. We have included activity diagrams which describes the different user’s process of completing a use case.
  7. We have included a state diagram for the entire coordination system which displays how the system can move from one state to another.
  8. We have included an ER Diagram which shows the basic structure of the database. Below the diagram is a small table showing how the information will be saved in the different tables.

# Data Dictionary