مرافق

SWE206: Introduction to Software Engineering

Marafiq: Project Phase 3

Software Requirements Specification (SRS)

Section 53, Group 07

Name	ID
MOHAMMED ALYOUSIF	202166710
ABDULLAH AL ABBAS	202156190
MAHDI AL HASSAN	202271080

Table of Contents

Table of Contents 2

Revision History 4

Detailed description of all tasks performed and work distribution: 5

1. Introduction 6

1.1 Purpose 6

1.2 Document Conventions 6

1.3 Intended Audience and Reading Suggestions 6

1.4 Project Scope 6

1.5 References 7

2. Overall Description 8

2.1 Product Perspective 8

2.2 Product Features 8

2.3 User Classes and Characteristics 8

2.4 Operating Environment 8

2.5 Design and Implementation Constraints 9

2.6 User Documentation 9

2.7 Assumptions and Dependencies 9

3. System Features 10

3.1 User Authentication and Role-Based Access Control 10

3.1.1 Description and Priority 10

3.1.2 Functional Requirements 10

3.2 Facility Reservation Management 10

3.2.1 Description and Priority 10

3.2.2 Functional Requirements 10

3.3 Prevention of Gender-Specific Policy Violations 10

3.3.1 Description and Priority 10

3.3.2 Functional Requirements 10

3.4 Group Reservation Initiation 11

3.4.1 Description and Priority 11

3.4.2 Functional Requirements 11

3.5 Reservation Confirmation and Cancellation 11

3.5.1 Description and Priority 11
3.5.2 Functional Requirements 11
4. Detailed Requirements 12
4.1 UML use case diagram 12
4.2 UML use case description 13
4.2.1 User Authentication 13
4.2.2 Reserve Facility 14
4.2.3 Waitlist Management 15
4.2.4 Send Notifications 16
4.2.5 Generate Usage Reports 17
4.2.6 Holding an Event 18
4.3 UML Class Diagram 19
4.4 UML Activity Diagram 20
4.5 UML Sequence Diagram 24
4.6 Non-functional Requirements 29
4.6.1 Reliability 29
4.6.2 Performance 29
4.6.3 Scalability 29
4.6.4 Security 29
4.6.5 Usability 29
4.6.6 Compatibility 29
4.6.7 Maintainability 30
4.6.8 Performance 30
4.6.9 Compliance 30
5. Prototypes and External Resources 31
5.1 Figma Prototype: 31
5.2 Repository Link: 31

Revision History

Name	Date	Reason For Change	Version
Mahdi Al Hassan	March 23, 2024	4.2.2 and 4.2.3 from users to participant and coordinator 4.2.4 actor: system only	1.1 (Phase 1)
Mohammed Alyousif	April 27, 2024	Adding 4.3, 4.4 and 4.5.	2.1 (Phase 2)
Mohammed Alyousif	May 14, 2024	Adding 5.1 and 5.2	3.1

Detailed description of all tasks performed and work distribution:

Task	Performed by:	
Logo	ABDULLAH AL ABBAS	MOHAMMED ALYOUSIF
File Formatting	MOHAMMED ALYOUSIF	
1	MOHAMMED ALYOUSIF	MAHDI AL HASSAN
2	MOHAMMED ALYOUSIF	MAHDI AL HASSAN
3.1	MOHAMMED ALYOUSIF	
3.2	MOHAMMED ALYOUSIF	
3.3	MOHAMMED ALYOUSIF	MAHDI AL HASSAN
3.4	MOHAMMED ALYOUSIF	MAHDI AL HASSAN
3.5	MOHAMMED ALYOUSIF	
4.1	MOHAMMED ALYOUSIF	MAHDI AL HASSAN
4.2.1	ABDULLAH AL ABBAS	
4.2.2	ABDULLAH AL ABBAS	MAHDI AL HASSAN
4.2.3	ABDULLAH AL ABBAS	MAHDI AL HASSAN
4.2.4	ABDULLAH AL ABBAS	MAHDI AL HASSAN
4.2.5	ABDULLAH AL ABBAS	
4.2.6	MOHAMMED ALYOUSIF	
4.3	MOHAMMED ALYOUSIF	
4.4	ABDULLAH AL ABBAS	
4.5	MOHAMMED ALYOUSIF	ABDULLAH AL ABBAS
4.6	MOHAMMED ALYOUSIF	MAHDI AL HASSAN
5	MOHAMMED ALYOUSIF	

1. Introduction

1.1 Purpose

The purpose of this document is to outline the functional and non-functional requirements for the development of a reservation management system for the Resources Management department at KFUPM. This system aims to streamline the process of reserving classrooms, labs, and various facilities within the university, ensuring efficient utilization and preventing scheduling conflicts.

1.2 Document Conventions

- Font: Times New Roman (Headings), Arial (Body)
- Font Size: 22 for Section Title, 16 for Headings, 14 for Sub-Headings and 12 for Body
- Bold Text for Section Title, Headings and Sub-Headings

Term	Definition
Admin	Chairmen and Deans
Coordinator	Faculty Members and Club Presidents
Participant	Students and University Employees

1.3 Intended Audience and Reading Suggestions

This document is intended for stakeholders involved in the development, implementation, and usage of Marafiq. Stakeholders include software developers, project managers, system administrators, and end-users such as faculty members, and students. For a comprehensive understanding, it is recommended that developers and project funders read the entire document. For end-users, such as faculty members, and students, it is advised to focus on reading the introduction, overall description, and system features sections to gain an understanding of Marafiq's functionalities and how it can benefit them.

1.4 Project Scope

The scope of the Marafiq project encompasses the design and development of a web-based reservation management system tailored to the specific needs of the Resources Management department at KFUPM. Marafiq will allow users to reserve classrooms, labs, and various facilities such as swimming pools, sports courts, and the gym. It will facilitate reservation requests, prevent double bookings, enforce access restrictions based on user roles and gender-specific policies, and provide functionality for group reservations and event management. Additionally, Marafiq will include features for administrators to manage reservations, send notifications, and maintain system reliability, security, and scalability.

1.5 References

"IEEE Guide for Software Requirements Specifications," in IEEE Std 830-1984, vol., no., pp.1-26, 10 Feb. 1984, doi: 10.1109/IEEESTD.1984.119205.

2. Overall Description

2.1 Product Perspective

In the current manual system utilized by the Resources Management department at KFUPM, such as Excel sheets, managing reservations for classrooms, labs, and facilities is inefficient and error-prone, leading to issues like double bookings and underutilization of resources. Marafiq presents a superior alternative, offering a centralized and automated reservation management system. Unlike manual methods, Marafiq streamlines the process through its user-friendly interface and automated functionalities, ensuring real-time availability checks, prevention of double bookings, and user authentication. It enhances transparency and accessibility by providing detailed information about available facilities and reservation statuses, reducing uncertainties and cancellations. Marafiq represents a significant advancement over manual methods, improving operational efficiency and enhancing the overall experience for administrators and end-users alike at KFUPM.

2.2 Product Features

Marafiq contain the following key features:

- User authentication and role-based access control.
- Facility reservation for classrooms, labs, swimming pools, sports courts, and the gym.
- Prevention of double bookings and enforcement of gender-specific policies.
- Group reservation initiation for collaborative activities.
- Reservation confirmation and cancellation functionalities.
- Email notifications for reservations, updates and cancellations.
- Administrative tools for managing reservations and user accounts.

2.3 User Classes and Characteristics

- Software Developers: Responsible for designing, developing, and maintaining Marafiq.
- Project Managers: Oversee the implementation and deployment of Marafiq.
- System Administrators: Manage user accounts, permissions, and system configurations.
- Faculty Members: Reserve academic buildings and facilities for events and activities.
- Club Presidents: Initiate group reservations and coordinate activities for club members.
- Students: Reserve non-academic facilities and participate in group reservations for various activities.

2.4 Operating Environment

Marafiq is a web-based application that operates on modern web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge. It requires an internet connection and access to the university network for authentication and data synchronization.

2.5 Design and Implementation Constraints

- Marafiq must comply with university policies and regulations regarding facility reservations and access restrictions.
- The system's design should prioritize usability and accessibility to accommodate users with varying levels of technical proficiency.
- Implementation must consider scalability to support a growing user base and reservation workload over time.

2.6 User Documentation

Comprehensive user documentation will be provided to guide users through the process of using Marafiq. This documentation will include instructions for registration, reservation, cancellation, and other system functionalities. Additionally, troubleshooting tips and frequently asked questions (FAQs) will be included to assist users in navigating the system.

2.7 Assumptions and Dependencies

- Marafiq assumes the availability of university databases for user authentication and facility information.
- The successful implementation and adoption of Marafiq depends on the support provided by the university administration.
- Dependencies include access to the university network and compliance with data security and privacy regulations.

3. System Features

3.1 User Authentication and Role-Based Access Control

3.1.1 Description and Priority

This feature involves authenticating users and assigning appropriate roles to control access to system functionalities. It is of high priority as it ensures that only authorized users can interact with the system and perform actions based on their roles.

3.1.2 Functional Requirements

- Users must be able to log in securely using their credentials.
- Upon successful authentication, the system should assign the appropriate role (i.e., admin, coordinator or participant) to the user.
- Access permissions should be enforced based on the user's role, allowing access to specific features and functionalities.

3.2 Facility Reservation Management

3.2.1 Description and Priority

This feature enables users to reserve classrooms, labs, and various facilities within the university campus. It is of high priority as it forms the core functionality of the system, facilitating efficient utilization of resources and preventing scheduling conflicts.

3.2.2 Functional Requirements

- Users should be able to view available facilities and their corresponding time slots.
- The system should prevent double bookings by enforcing availability checks.
- Users should be able to submit reservation requests for specific facilities and time slots.
- Admins should have the ability to accept, reject, or modify reservation requests.

3.3 Prevention of Gender-Specific Policy Violations

3.3.1 Description and Priority

This feature ensures compliance with gender-specific policies for certain facilities, such as separate swimming pools for male and female students. It is of high priority to uphold university regulations and maintain a safe and inclusive environment.

3.3.2 Functional Requirements

- The system should enforce gender-specific restrictions for facilities where applicable.

- Users should be restricted from reserving facilities that are designated for the opposite gender.
- Admins should be able to configure and manage gender-specific facilities policies.

3.4 Group Reservation Initiation

3.4.1 Description and Priority

This feature allows users to initiate group reservations for collaborative activities or events. It is of high priority as it promotes community engagement and facilitates coordinated use of facilities.

3.4.2 Functional Requirements

- Users should be able to initiate group reservations and specify the minimum and maximum number of participants.
- The system should confirm group reservations once the required number of participants is met.
- Participants should be able to join existing group reservations initiated by other users.

3.5 Reservation Confirmation and Cancellation

3.5.1 Description and Priority

This feature provides users with the ability to confirm or cancel their reservations as needed. It is of high priority to ensure flexibility and responsiveness in managing reservations.

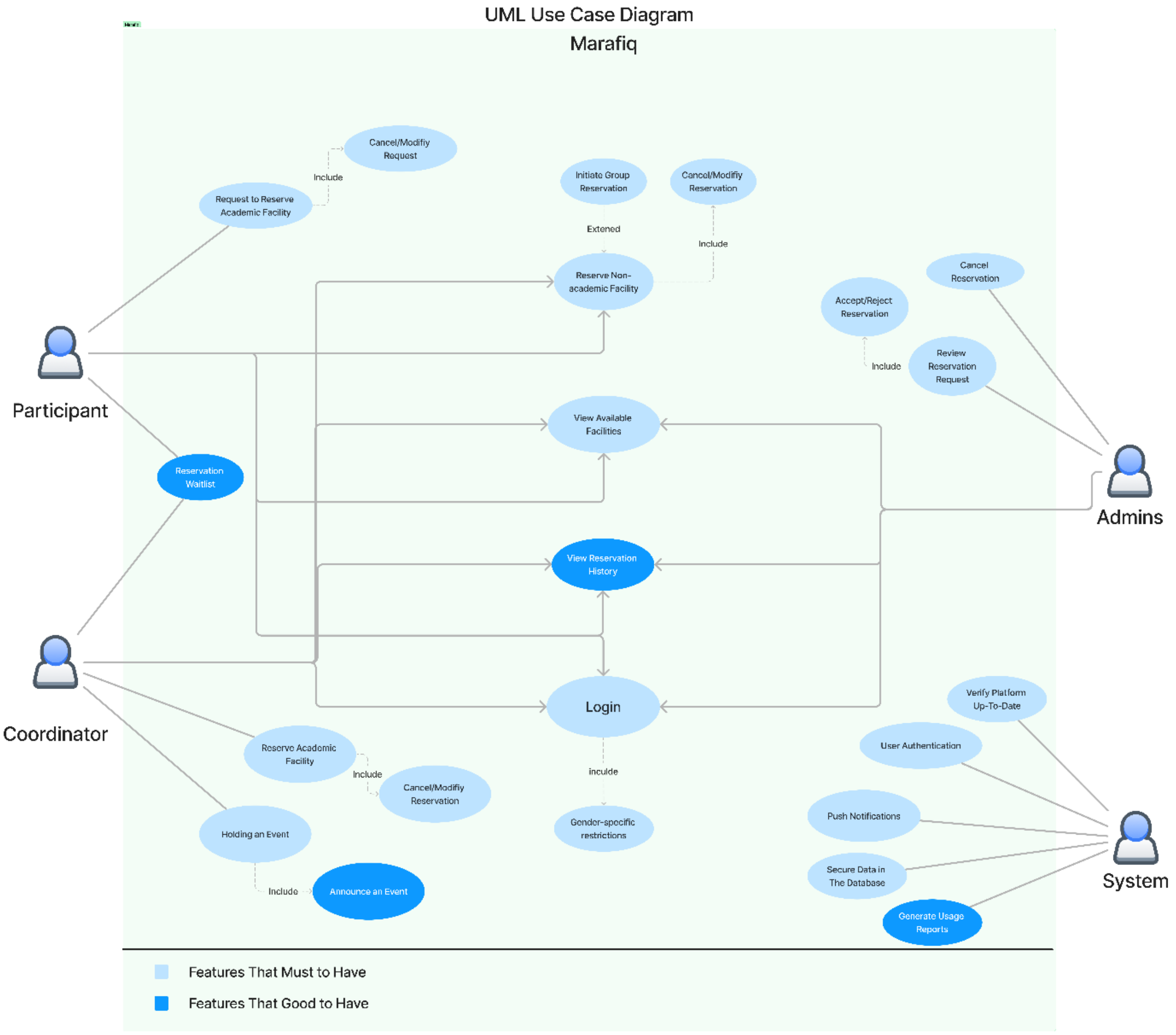
3.5.2 Functional Requirements

- Users should receive confirmation notifications upon successful reservation.
- Users should be able to cancel their reservations within a specified time frame.
- Admins should have the ability to cancel reservations and notify affected users accordingly.

4. Detailed Requirements

4.1 UML use case diagram

For a better resolution click here or on the figure.


UML Use Case Diagram
Marafiq
Cancel/Modifiy Request
Include
Request to Reserve Academic Facility
Initiate Group Reservation
Cancel/Modifiy Reservation
Extened
Include
Reserve Non-academic Facility
Cancel Reservation
Accept/Reject Reservation
Include
Review Reservation Request
Participant
View Available Facilities
Reservation Waitlist
Admins
View Reservation History
Coordinator
Login
Verify Platform Up-To-Date
User Authentication
Reserve Academic Facility
Include
Cancel/Modifiy Reservation
inculde
Holding an Event
Gender-specific restrictions
Push Notifications
Include
Announce an Event
Secure Data in The Database
System
Generate Usage Reports
Features That Must to Have
Features That Good to Have

4.2 UML use case description

4.2.1 User Authentication

Use case ID:	1.1		
Use Case Name:	User Authentication		
Created by:	Abdullah Al Abbas	**Last Updated by:**	
Date Created:	March 23, 2024	**Last Revision Date:**	
Actors:	Admins, Participant, Coordinator and System		
Description:	The process of authenticating users to access Marafiq		
Trigger:	The user press login button		
Preconditions:	The Device has an internet connection. The user has a valid username and password.		
Postconditions:	If authentication is successful, the user gains access to Marafiq and its functionalities. If authentication fails, the user is notified and unable to access Marafiq.		
Normal Flow:	The user navigates to the login page of Marafiq. The user enters their username and password. The system verifies the credentials provided by the user. If the credentials are valid, the system grants access to the user. If the credentials are invalid, the system denies access and prompts the user to re-enter the credentials.		
Alternative Flows:	If the user forgets their password: The user clicks on the "Forgot Password" button. The system prompts the user to enter their email or username for password recovery. The system sends a password reset link or instructions to the user's registered email. The user follows the instructions to reset their password and gain access.		
Exceptions:			
Assumptions:	Users have valid credentials registered in the system.		
Notes and Issues:	Users should be from KFUPM		

4.2.2 Reserve Facility

Use case ID:	1.2		
Use Case Name:	Reserve Facility		
Created by:	Abdullah Al Abbas	**Last Updated by:**	
Date Created:	March 23, 2024	**Last Revision Date:**	
Actors:	Participant, coordinator		
Description:	The process of reserving a facility using Marafiq		
Trigger:	The user clicks on the thing that they want to reserve		
Preconditions:	The user is authenticated and logged into Marafiq. The user has selected a facility they want to reserve. The facility is available for reservation at the desired date and time slot.		
Postconditions:	The facility is successfully reserved for the user at the specified date and time. The reservation details are stored in the system for future reference.		
Normal Flow:	The user navigates to the facility reservation section. The user selects the type of facility they want to reserve. The user specifies the date and time slot for the reservation. The system checks the availability of the selected facility for the specified date and time. If the facility is available, the user confirms the reservation by submitting the request. The system records the reservation details, including the user, facility, date, and time.		
Alternative Flows:	If the selected facility is not available for the specified date and time: The system notifies the user of the unavailability and suggests alternative options if applicable. The user selects a different date or time slot or chooses a different facility for reservation. The user repeats the reservation process with the updated information.		
Exceptions:			
Assumptions:	The reservation system maintains up-to-date availability information for all facilities.		
Notes and Issues:	The system should provide real-time feedback on facility availability to avoid double bookings.		

4.2.3 Waitlist Management

Use case ID:	1.3		
Use Case Name:	Waitlist Management		
Created by:	Abdullah Al Abbas	**Last Updated by:**	
Date Created:	March 23, 2024	**Last Revision Date:**	
Actors:	Participant, coordinator		
Description:	The process of managing waitlists for fully booked facilities.		
Trigger:	The user clicks on join the waitlist button.		
Preconditions:	The user is authenticated and logged into Marafiq. The desired facility is fully booked for the specified date and time. The user chooses to join the waitlist for the facility.		
Postconditions:	The user is added to the waitlist for the facility. If a spot becomes available, the user is automatically confirmed for the reservation.		
Normal Flow:	The user selects a fully booked facility for reservation. The system checks the availability and determines that the facility is fully booked. The system prompts the user to join the waitlist for the facility. The user confirms their decision to join the waitlist. The user is added to the waitlist queue for the facility.		
Alternative Flows:	If a spot becomes available on the waitlist due to a cancellation: The system notifies the next user on the waitlist about the available spot. The user is automatically confirmed for the reservation. The system updates the reservation status and notifies the user accordingly.		
Exceptions:			
Assumptions:	Users are informed about their waitlist status and potential reservation updates via notifications.		
Notes and Issues:	The system should prioritize waitlist confirmations based on the order of joining the waitlist (first-come, first-served). Users on the waitlist should have the option to cancel their waitlist request if they no longer wish to wait for a spot.		

4.2.4 Send Notifications

Use case ID:	1.4		
Use Case Name:	Send Notifications		
Created by:	Abdullah Al Abbas	**Last Updated by:**	
Date Created:	March 23, 2024	**Last Revision Date:**	
Actors:	System		
Description:	The process of sending notifications to users		
Trigger:	An event happens that requires sending a notification		
Preconditions:	The system has access for sending notifications. Users have provided valid addresses during registration.		
Postconditions:	Users receive notifications with relevant information.		
Normal Flow:	An event triggers the need for a notification, such as a reservation being confirmed or cancelled. The system retrieves the necessary information for the notification, including the user's address and the details of the reservation event. The system composes the notification with a suitable message, including details like reservation date, time, facility, and reason. The system sends the notification to the user's registered address. The user receives the notification.		
Alternative Flows:	If the email server is temporarily unavailable: The system queues the notification for sending later. Once the server is accessible, the system sends the queued notifications.		
Exceptions:	If the address provided by the user is invalid or inaccessible, the system can provide an option for changing the address.		
Assumptions:	Users have provided valid and accessible email addresses during registration.		
Notes and Issues:	Email notifications should be clear, concise, and informative to keep users informed.		

4.2.5 Generate Usage Reports

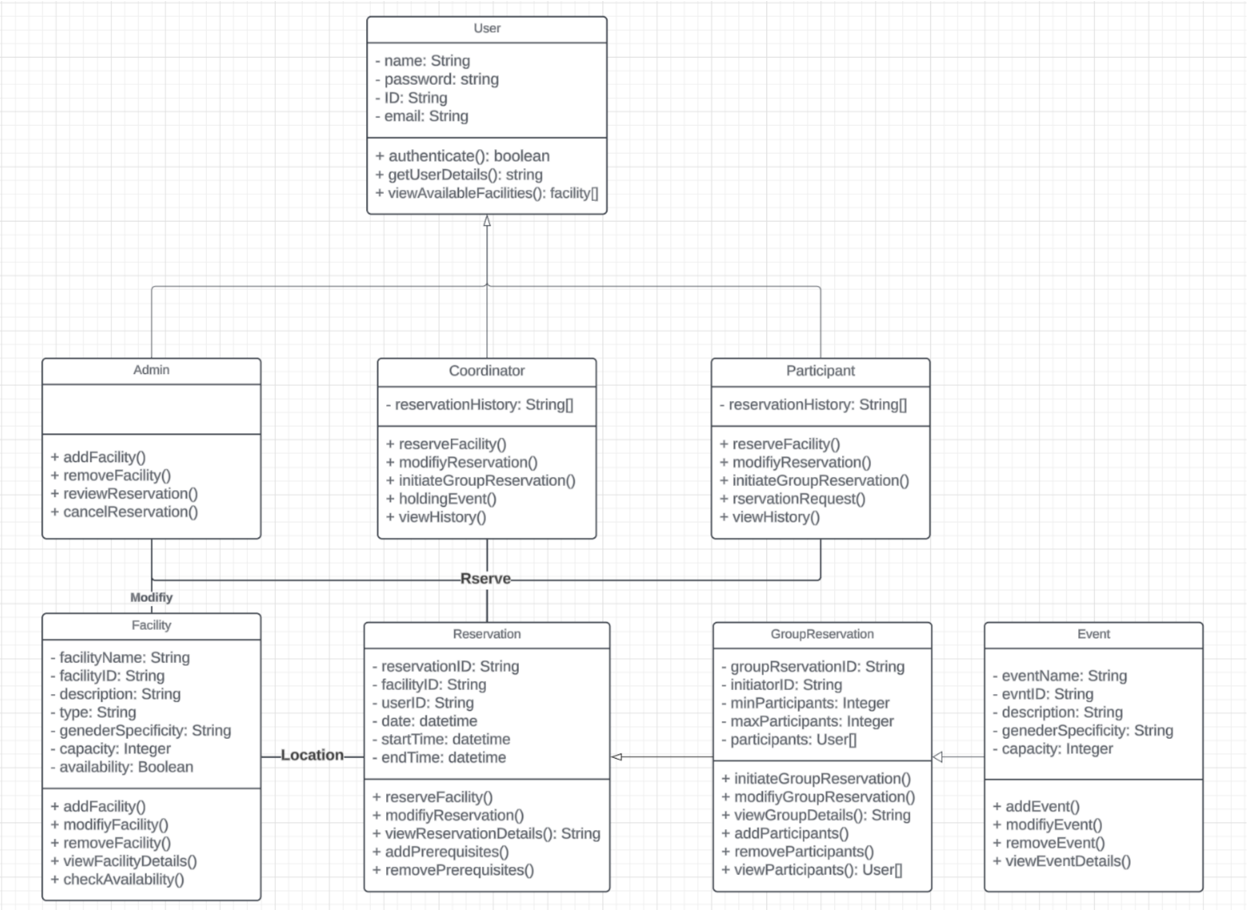
Use case ID:	1.5		
Use Case Name:	Generate Usage Reports		
Created by:	Abdullah Al Abbas	**Last Updated by:**	
Date Created:	March 23, 2024	**Last Revision Date:**	
Actors:	System, Administrators		
Description:	The process of generating usage reports for facilities and reservations within Marafiq.		
Trigger:	Admit click on Generate Report button.		
Preconditions:	The admin is authenticated and logged into the admin panel of the reservation system. Marafiq has collected sufficient data on facility reservations and usage.		
Postconditions:	Marafiq generates usage reports based on specified criteria and parameters. The admin can view the generated reports for analysis.		
Normal Flow:	The admin navigates to the reporting or analytics section within the admin panel. The admin presses on the button that's generates the report. Marafiq processes the request and retrieves relevant data. Marafiq generates the usage report. The admin reviews the generated report.		
Alternative Flows:	If the generated report contains sensitive or confidential information: The admin can specify access permissions for viewing the sensitive report data.		
Exceptions:			
Assumptions:	Marafiq collects comprehensive data on facility reservations, usage patterns, and user activities.		
Notes and Issues:	Admins should be able to schedule automated report generation at regular intervals for ongoing monitoring and analysis.		

4.2.6 Holding an Event

Use case ID:	1.6		
Use Case Name:	Holding an Event		
Created by:	Mohammed Alyousif	**Last Updated by:**	
Date Created:	March 23, 2024	**Last Revision Date:**	
Actors:	Coordinator		
Description:	The process of coordinating and holding an event, such as a workshop, seminar, or sports activity, within the university premises.		
Trigger:	Coordinator initiates the process of holding an event.		
Preconditions:	Coordinator is authenticated and logged into Marafiq. Coordinator has a reserved facility.		
Postconditions:	The event is successfully scheduled and confirmed, and participants are notified.		
Normal Flow:	Coordinator navigates to a reserved facility within my reservation panel. Coordinator presses on the Holding an Event button. Coordinator specifies description and audience of the event. Marafiq announce the event to the desired audiences.		
Alternative Flows:	If the Admin cancels the event: Marafiq notifies Coordinator with the reasons.		
Exceptions:	If the Coordinator fails to provide all necessary event details, the system prompts them to complete the required information.		
Assumptions:	Participants are responsible for attending the event at the specified date, time, and location.		
Notes and Issues:			

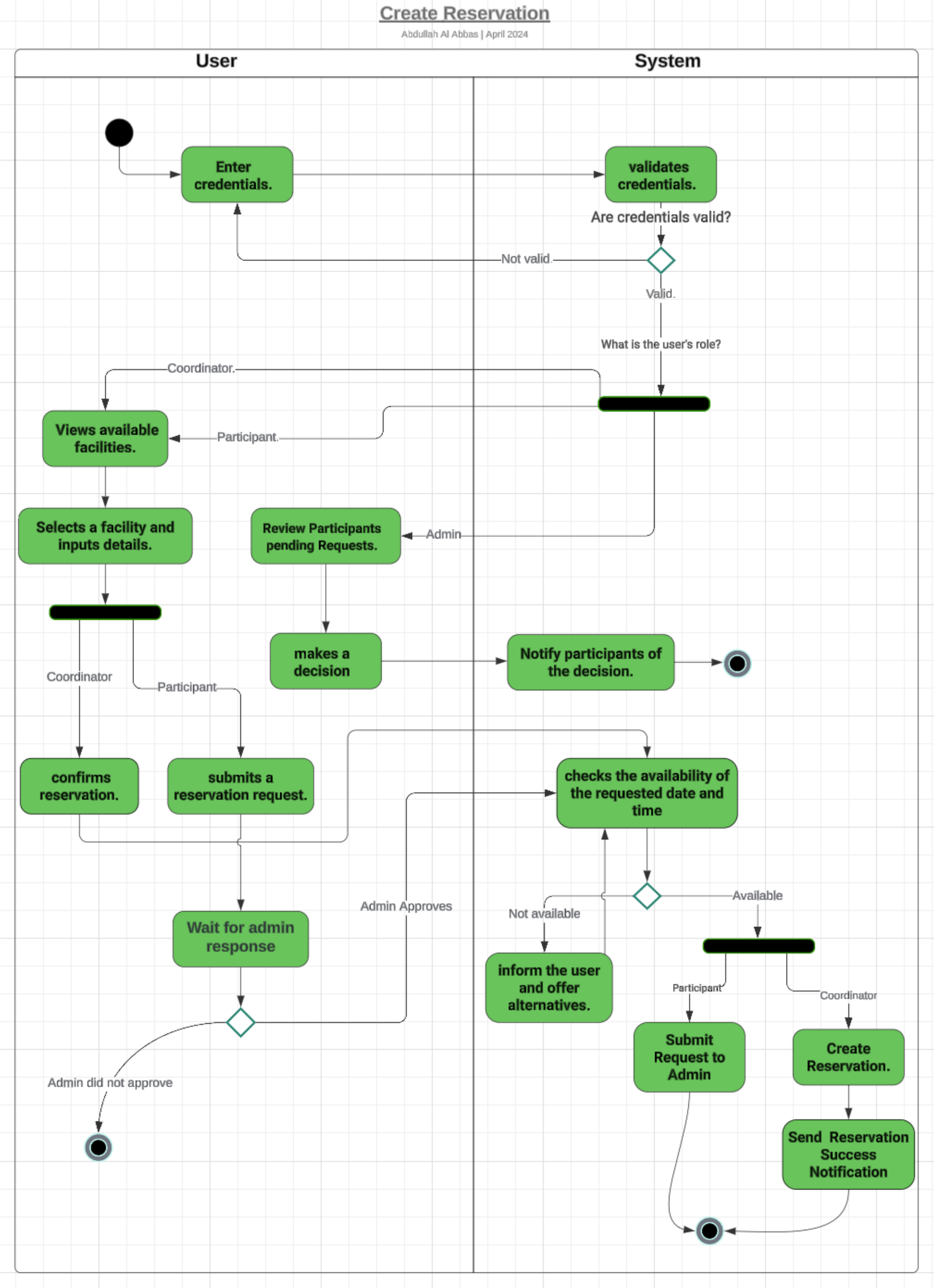
4.3 UML Class Diagram

For a better resolution click here or on the figure.



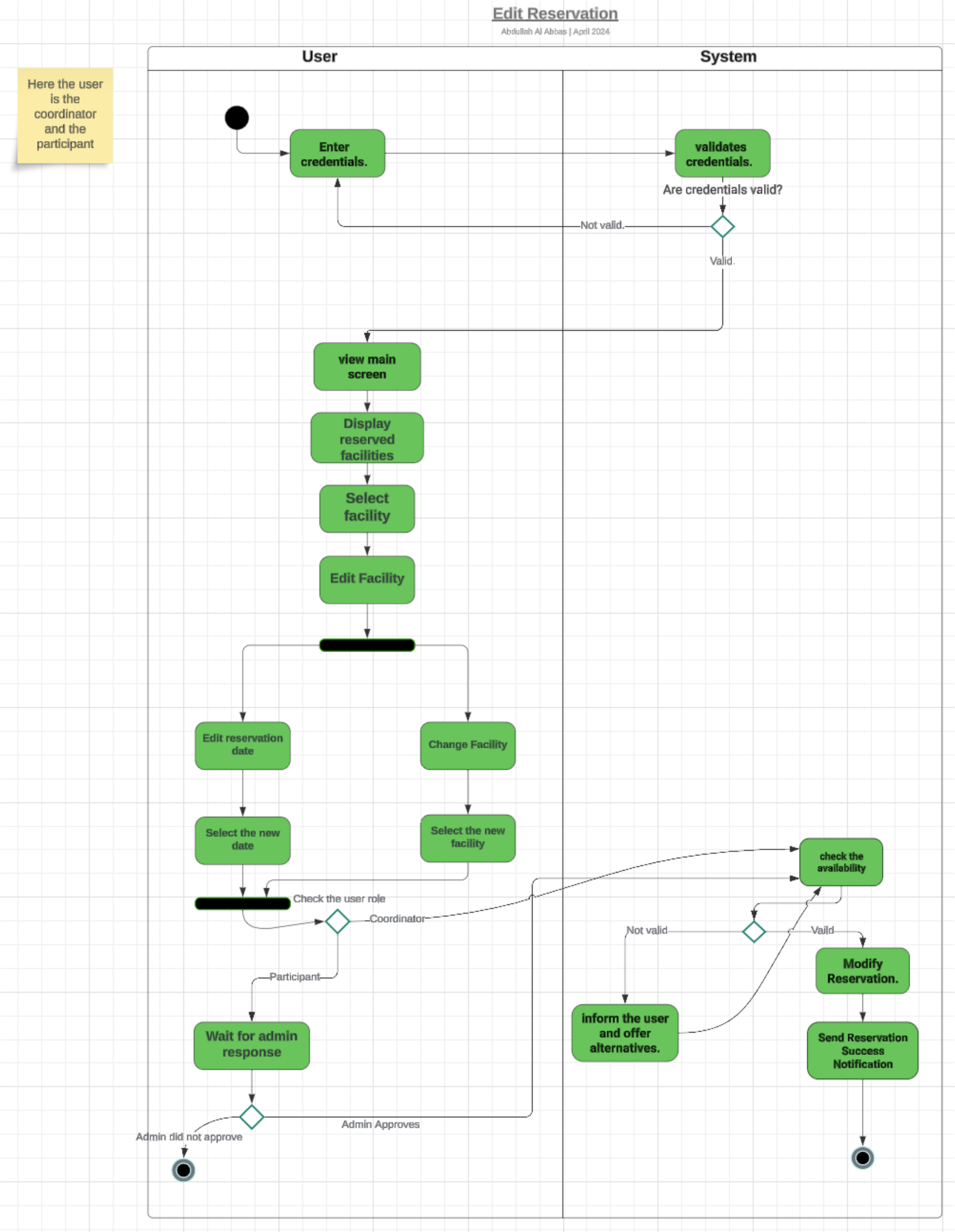
4.4 UML Activity Diagram

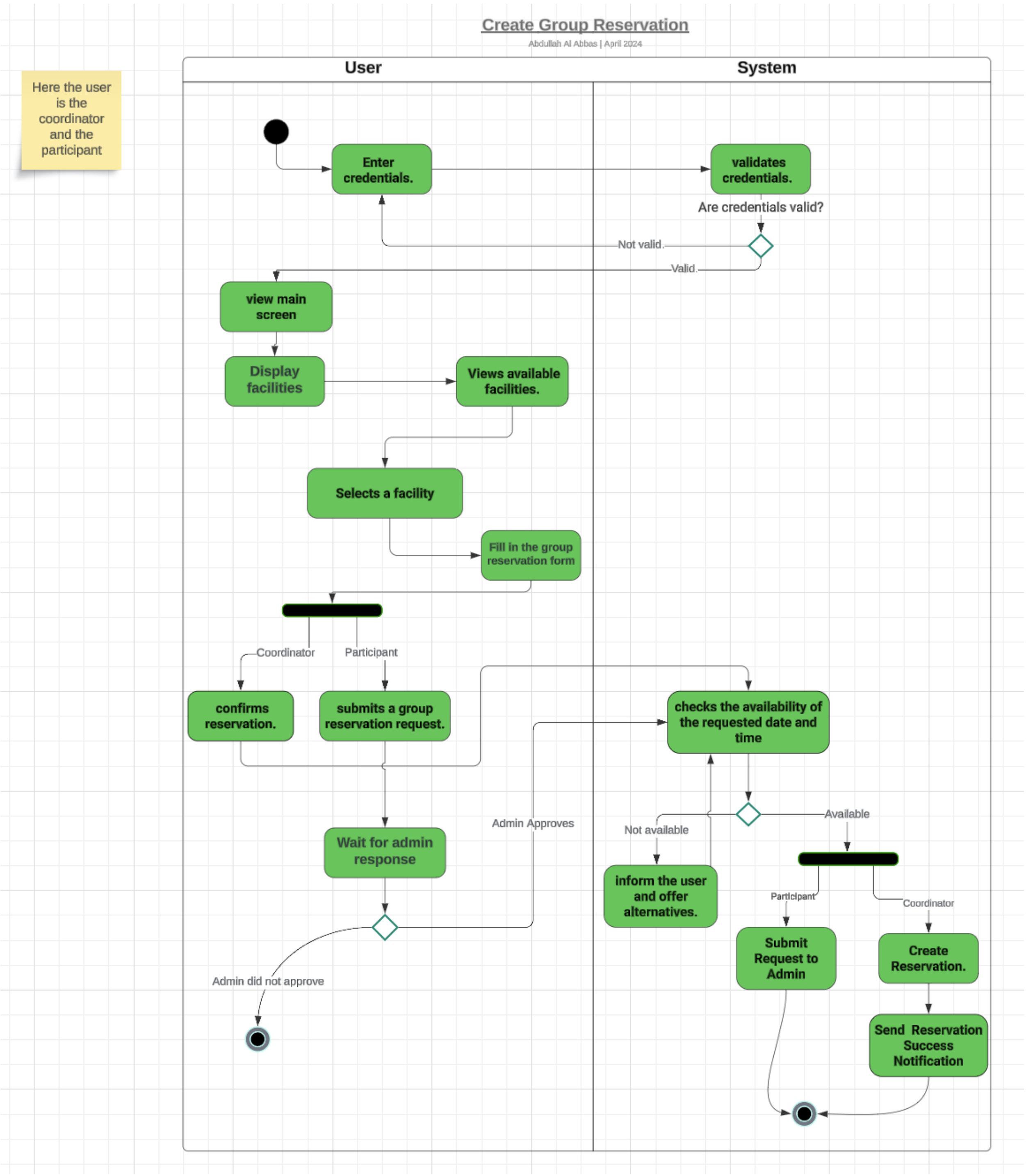
For a better resolution click here or on the figure.

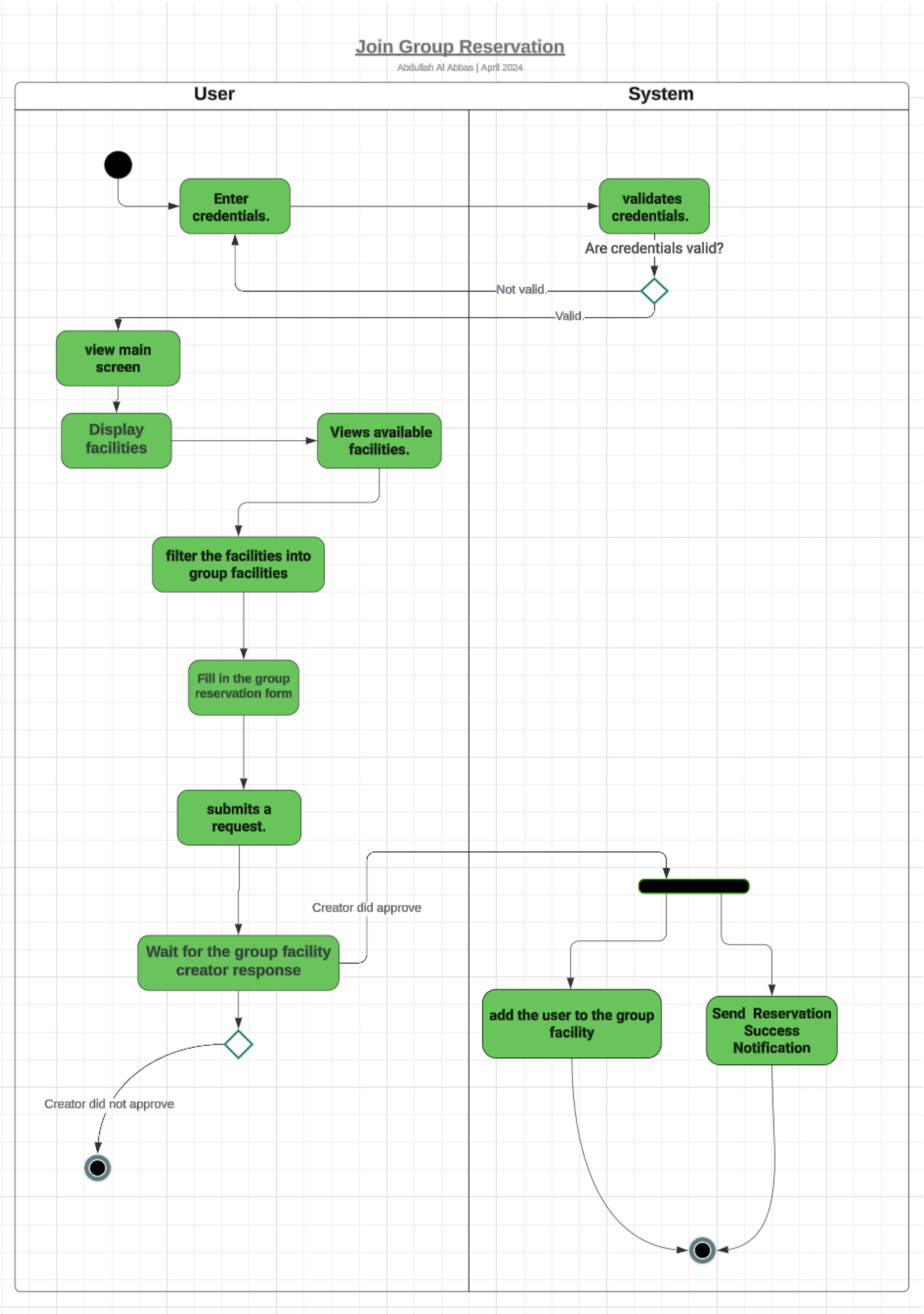


Create Reservation

Abdullah Al Abbas | April 2024

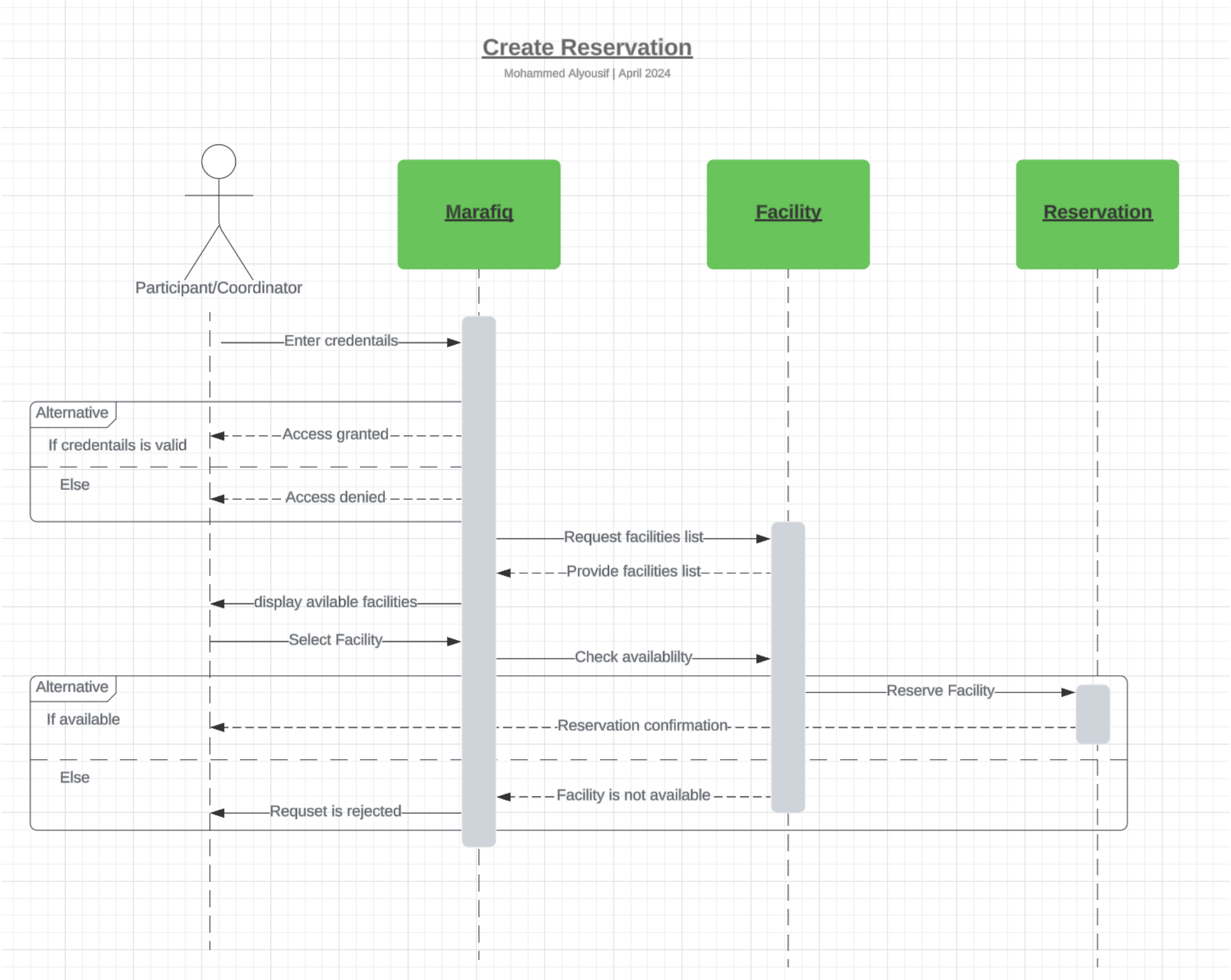

Edit Reservation
Abdullah Al Abbas | April 2024
Here the user is the coordinator and the participant
User
System
Enter credentials.
validates credentials.
Are credentials valid?
Not valid.
Valid.
view main screen
Display reserved facilities
Select facility
Edit Facility
Edit reservation date
Change Facility
Select the new date
Select the new facility
Check the user role
Coordinator
Participant
Wait for admin response
Admin did not approve
Admin Approves
check the availability
Not valid
Vaild
inform the user and offer alternatives.
Modify Reservation.
Send Reservation Success Notification

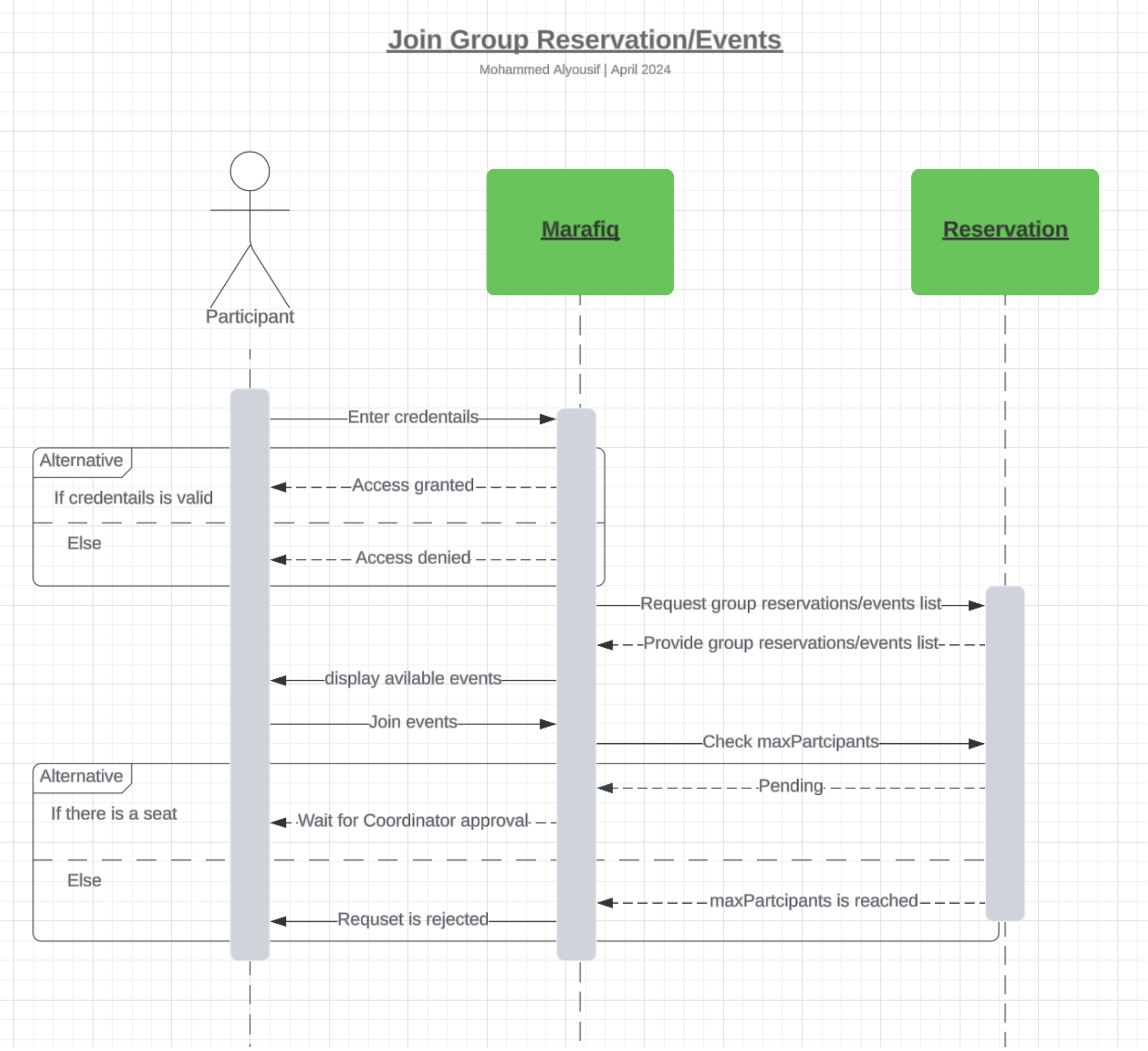

Create Group Reservation
Abdullah Al Abbas | April 2024
Here the user is the coordinator and the participant
User
System
Enter credentials.
validates credentials.
Are credentials valid?
Not valid.
Valid
view main screen
Display facilities
Views available facilities.
Selects a facility
Fill in the group reservation form
Coordinator
Participant
confirms reservation.
submits a group reservation request.
checks the availability of the requested date and time
Wait for admin response
Admin Approves
Not available
Available
inform the user and offer alternatives.
Participant
Coordinator
Submit Request to Admin
Create Reservation.
Send Reservation Success Notification
Admin did not approve

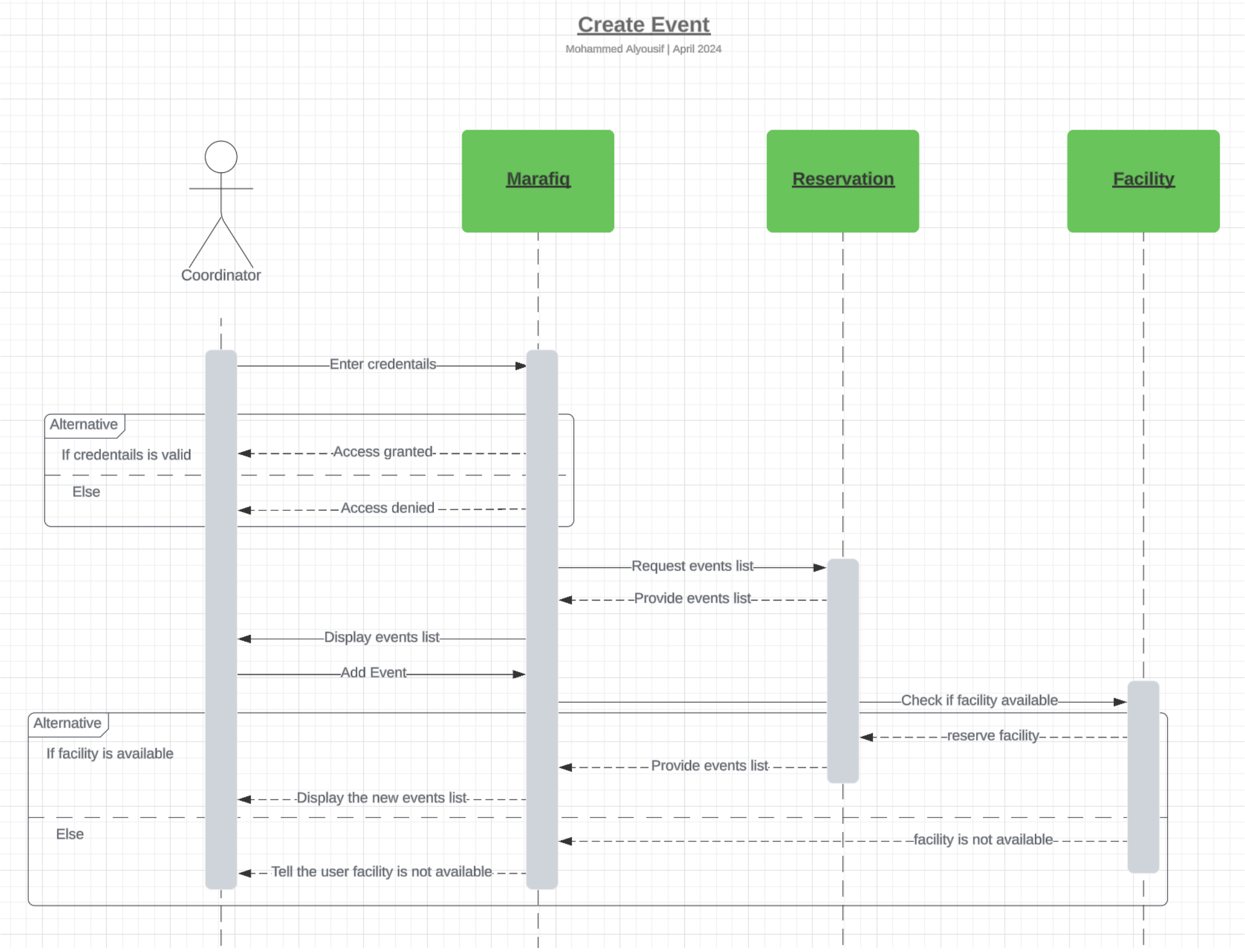

Join Group Reservation
Abdullah Al Abbas | April 2024
User
System
Enter credentials.
validates credentials.
Are credentials valid?
Not valid.
Valid.
view main screen
Display facilities
Views available facilities.
filter the facilities into group facilities
Fill in the group reservation form
submits a request.
Creator did approve
Wait for the group facility creator response
Creator did not approve
add the user to the group facility
Send Reservation Success Notification

4.5 UML Sequence Diagram

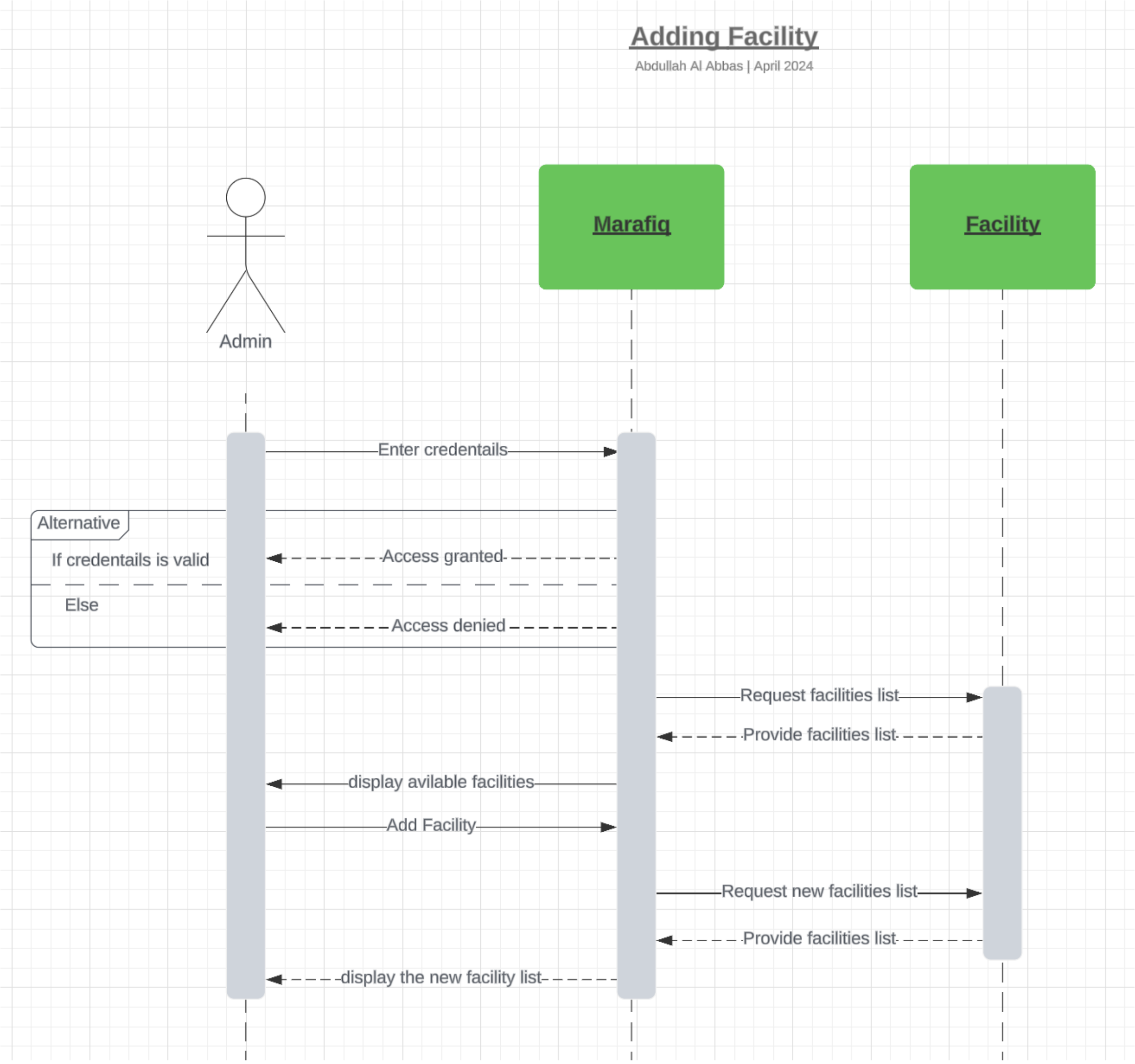
For a better resolution click here or on the figure.

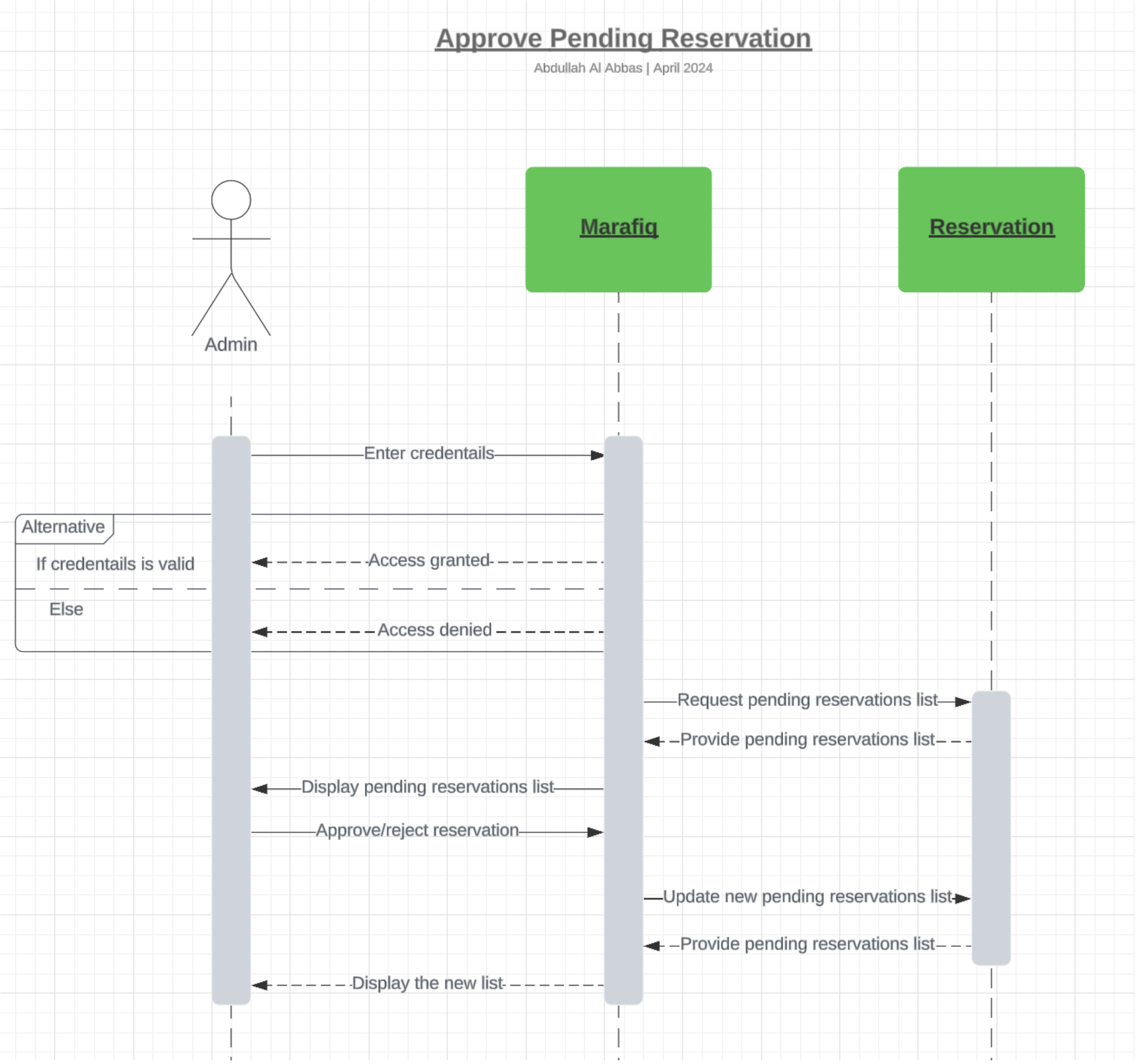



Join Group Reservation/Events
Mohammed Alyousif | April 2024
Participant
Marafiq
Reservation
Enter credentails
Alternative
If credentails is valid
Access granted
Else
Access denied
Request group reservations/events list
Provide group reservations/events list
display avilable events
Join events
Check maxPartcipants
Alternative
If there is a seat
Pending
Wait for Coordinator approval
Else
maxPartcipants is reached
Requset is rejected


Create Event

Mohammed Alyousif | April 2024


Adding Facility
Abdullah Al Abbas | April 2024
Admin
Marafiq
Facility
Enter credentails
Alternative
If credentails is valid
Access granted
Else
Access denied
Request facilities list
Provide facilities list
display avilable facilities
Add Facility
Request new facilities list
Provide facilities list
display the new facility list


Approve Pending Reservation
Abdullah Al Abbas | April 2024
Admin
Marafiq
Reservation
Enter credentails
Alternative
If credentails is valid
Access granted
Else
Access denied
Request pending reservations list
Provide pending reservations list
Display pending reservations list
Approve/reject reservation
Update new pending reservations list
Provide pending reservations list
Display the new list

4.6 Non-functional Requirements

4.6.1 Reliability

- The system shall be highly reliable, with minimal downtime and errors, ensuring uninterrupted access to reservation functionalities.
- It shall be capable of handling large volumes of concurrent users and reservation requests without system failures.

4.6.2 Performance

- The system shall be responsive, providing quick response times for user interactions such as viewing available facilities and making reservations.
- It shall support efficient data processing and retrieval to minimize latency in accessing reservation information.

4.6.3 Scalability

- The system shall be scalable to accommodate a growing number of users, facilities, and reservation requests over time.
- It shall be capable of scaling both vertically and horizontally to handle increased workload and user traffic.

4.6.4 Security

- User authentication and data transmission shall be secured using industry-standard encryption protocols to protect sensitive information.
- Access controls shall be implemented to ensure that users can only access resources and functionalities appropriate for their roles.
- The system shall comply with data privacy regulations and best practices to safeguard user privacy and confidentiality.

4.6.5 Usability

- The user interface shall be intuitive and user-friendly, requiring minimal training for users to navigate and utilize reservation functionalities.
- Accessibility features shall be implemented to ensure that the system is usable by users with disabilities, adhering to accessibility standards such as WCAG.

4.6.6 Compatibility

- The system shall be compatible with a wide range of web browsers and devices, including desktops, laptops, tablets, and smartphones, to ensure accessibility for all users.
- It should support cross-platform compatibility to enable seamless access across different operating systems and devices.

4.6.7 Maintainability

- The system shall be designed with modular and well-structured code to facilitate ease of maintenance and future enhancements.
- Documentation shall be comprehensive and up to date, providing guidance for system administrators and developers on system configuration, deployment, and maintenance tasks.

4.6.8 Performance

- The system shall be capable of handling peak loads and surges in user activity, maintaining consistent performance without degradation in response times.
- Load testing shall be conducted periodically to identify performance bottlenecks and optimize system performance accordingly.

4.6.9 Compliance

- The system shall comply with relevant regulatory requirements and industry standards, including university policies, data protection laws, and security standards.
- Regular audits and compliance assessments shall be conducted to ensure ongoing adherence to regulatory requirements and standards.

5. Prototypes and External Resources

5.1 Figma Prototype:

The Figma prototype provides a visual representation of the user interface and user experience design for the project. It is an essential tool for understanding the layout, navigation, and functionality from a user's perspective. click here

5.2 Repository Link:

The project repository contains all the source code, documentation, and related resources necessary for the development and maintenance of the project. Access to the repository is restricted and requires authentication using the provided token.

Repository URL: https://github.com/MR-Alyousif/Marafiq