UNIVERSITI TEKNOLOGI MALAYSIA

SECJ2203 SOFTWARE ENGINEERING

SYSTEM DOCUMENTATION

EVENT MANAGEMENT SYSTEM FOR K01, KOLEJ TUN RAZAK

25th MAY 2024

FACULTY OF COMPUTING

LECTURER: DR. SIM HIEW MOI

PREPARED BY:

GROUP MEMBERS	MATRIC NUMBER
VENNISE NGOH YAN LENG	B23CS8032
ARIEF NAJMI BIN BAKIYUDIN	A23CS8004
OMAR ABDELMONEM HANAFY	A22EC4012
AINA ZAFIRAH BINTI AHMAD MUZAMIR	A23CS8014
SADIK AL MAHMUD	A20EC4049

REVISION PAGE

a. Overview

This Software Development document outlines the process, methodology, and timeline for the development of the Event Management System for Kolej Tun Razak. It includes details on the project's goals, scope, chosen software process model, and project duration. Additionally, it provides insights into the technological aspects, user requirements, and the iterative approach adopted for the development.

b. Target Audience

The Software Development (SD) document is intended for project stakeholders, including developers, administrators, and users involved in the development, deployment, and use of the Event Management System for Kolej Tun Razak.

c. Project Team Members

Member Name	Role	Task	Status
Omar Abdelmonem	**Introduction**	Introduction, Purpose, Scope & 3 user cases	Complete
Aina Zafirah	**Introduction**	Definitions, Reference & Overview	Complete
Vennise Ngoh	**Specific Requirements**	System Features & Use case	Complete
Arief Najmi	**Specific Requirements**	User Characteristics	
Sadik Al Mahmud	**Specific Requirements**	Software System Attributes & Design Constraints	

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
	Team Leader (Omar Abdelmonem)	Complete introduction & Purpose	10/04/2024
	Omar Abdelmonem	Complete Scope	14/04/2024
		Added sequence diagram for UC001, UC002 and UC003	25/05/2024
		Added activity diagram for UC001, UC002 and UC003	25/05/2024
	Vennise Ngoh Yan Leng	Complete User Characteristics, with Use Case diagram ,Listed all Use Case	15/04/2024
	Vennise Ngoh Yan Leng	Complete Use Case AU 001 - AU003	28/04/2024
	Vennise Ngoh Yan Leng	Complete Use Case AU 004 and AU005	07/05/2024
	Vennise Ngoh Yan Leng	Added domain model and description of domain model.	18/05/2024
	Vennise Ngoh Yan Leng	Added sequence diagram for AU 001 - AU 003	24/05/2024

	Vennise Ngoh Yan Leng	**Added activity diagram for AU 001 - AU 003**	**25/05/2024**
	Arief Najmi Bin Bakiyudin	**Complete Use Case Descriptions 1,2,3,4,5**	**12/05/2024**
	Arief Najmi Bin Bakiyudin	**Complete Use Case Diagram 4,56**	**15/05/2024**
	Arief Najmi Bin Bakiyudin	**Complete Activity Diagram 4,5,6**	**23/05/2024**
	Aina Zafirah Binti Ahmad Muzamir	**Complete Definition**	**07/05/2024**
	Aina Zafirah Binti Ahmad Muzamir	**Complete use case description for UC006 and UC007**	**11/05/2024**
	Aina Zafirah Binti Ahmad Muzamir	**Complete Use case description for AC006, AC007 and AC008**	**12/05/2024**
	Aina Zafirah Binti Ahmad Muzamir	**Complete sequence diagram and activity diagram for AC006, AC007 and AC008**	**25/05/2024**

TABLE OF CONTENTS

1 INTRODUCTION

The Event Management System for Kolej Tun Razak is a digital tool designed to streamline event planning, organization, and execution within the college premises. By integrating a range of features for both organizers and administrators, the system aims to enhance efficiency and effectiveness in managing events. This document serves as an overview of the system development process, comprising the System Requirements Specification (SRS), System Design Document (SDD), and System Testing Document (STD).

1.1 Purpose

The purpose of this System Documentation (SD) is two-fold: to provide a comprehensive overview of the development process for the Event Management System for Kolej Tun Razak and to serve as a reference document for stakeholders involved in the system development process, including developers, testers, project managers, and clients. The SD aims to ensure clarity, coherence, and accuracy in documentation, guiding the development process and fostering alignment among team members. It delineates the purpose by providing a comprehensive overview of the system development process, including requirements, design, and testing phases. The intended audience for the SD includes developers, testers, project managers, and clients involved in the system development process.

1.2 Scope

The Event Management System for Kolej Tun Razak encompasses the following aspects:

- Software Product

  The system facilitates event planning, venue booking, and activity management within Kolej Tun Razak.

- Functionality

  It offers features for organizers to request event places, purchase activities, and schedule events, while administrators can manage event places, activities, and approvals.

- Application

  The system is designed to reduce the workload for both organizers and administrators, improve efficiency in event planning, and provide data for performance analysis and decision-making.

The scope for the Event Management System for Kolej Tun Razak varies based on specific needs and requirements of the users:

- **Responsiveness and Accessibility**

  The system shall be designed as a responsive website, easily accessible by all users using different devices. It should be operational at all times, considering that most students are busy during office hours because of study commitments.

- **Administrator Website**

  Administrators should have their own website that can manage all the places, events, facilities, and manpower supported in Kolej Tun Razak. Users' website accounts and administrator website accounts should be separated for security purposes.

- **Administrator Features**

  The system for administrators must provide features for adding, updating, and deleting event places, facilities, activities, and activities packages.

- **User Website Features**

  For users' websites, it must have the following features:

  - View existing places

  - Search by date and time

  - View activities and facilities provided for each event

  - Sending booking requests

  - Renting facilities if needed

  - Buying activities if needed

Technologies Implemented:

NO.	TECHNOLOGY	DESCRIPTION
1.	Website programming	Flexible, and high capability on any device and browser.
2.	AJAX	High performance and increase the UX by reducing or preventing reloading the page while transfering a data.
3.	JSON	Stores all data in an array that makes data transfer much easier, and higher performance.
4.	Bootstrap	Have many features and libraries provided for increasing the user interface experience.

Table 1.1

1.3 Definitions, Acronyms And Abbreviations

TERM	DEFINITION
SRS	System Requirement Specification - a document that outlines the requirements and objectives of the software product to be developed.
SDD	System Design Documentation - a document that describes the system architecture, components and interfaces in detail.
STD	System Testing Documentation - a document that outlines the testing process and procedures for the software product to be developed.
UX	User Experience - the overall experience a user has while interacting with a product, including ease of use, efficiency and satisfaction.

1.4 References

1.5 Overview

The System Documentation (SD) provides a comprehensive description of the Event Management System for Kolej Tun Razak to be developed. This document comprises three main sections which are the System Requirements Specification (SRS), System Design Documentation (SDD) and System Testing Documentation (STD).

In the specific requirements, we brief about the use case we have in our system with a use case diagram. And the object in the system using UML diagrams. By this diagram, users can have a clear understanding of the system functionality for each type of user. Other than that, in each use case listed, we also use a use case table, sequence diagram and activities diagram to show the data flow and the activity flow in the system.

We also listed the operating environment for the system hardware, software and others systems that will interact with our system. For non-functional requirements, we listed it at system constraint.

The System Requirements Specification (SRS) serves as a contract between the stakeholders and the development team, guiding the development process and ensuring that the final product meets the stakeholders' needs and expectations. It provides a roadmap for the design, implementation, testing and validation for the Event Management System.

2 SPECIFIC REQUIREMENTS

The Event Management System will need to have several characteristics and features to provide a smooth experience for user and administrator. In this section, we will provide a clear image on the system user characteristics, functionality and user cases in every scenario input and output.

2.1 User Characteristics

In this section, we will describe the users of this system with Use Case and their characteristics for the development team to have a clear understanding of the whole system, what is the system design for and to provide a ruler on the needs of the system.

The Event Management System will have two main types of users: UTM students and KTR administrators.

2.1.1 UTM students

- The UTM students who will use this system are expected to have a need for finding places, ways of organizing an event in KTR.
- They may have a range of decided dates for organizing their event.
- Some students might have no ideas on what activities they can put in their event, and find guidance for their event.
- Some students may just want to book for a section of hours for their event and some may want to book for several days.
- All UTM students can create their account on the user page system.

2.1.2 KTR administrators

- The KTR administrators are expected to have professional training and experience on organizing an event.
- They must know all the places information, address, and what activities they provide for event organizers.
- They will use this software to approve student place booking requirements and replying to their questions.
- Only one administrator has the permission to create a new account for other KTR administrators.

Both users will be required to provide some personal information to create their account in the system, for example: username, email address, and bank account if online transfer is needed. The software must be designed with high security measures to protect every user information from being seen by others.

2.2 System Features

Event Management System is a website application that can be accessed by different devices and browsers. The system is created for giving faster and easier ways for UTM students to organize their events for their course project and at the same time provide KRT administrators with a better system for checking the event schedule and place booking status. Both users will be able to generate a report of their activities or performance of the month.

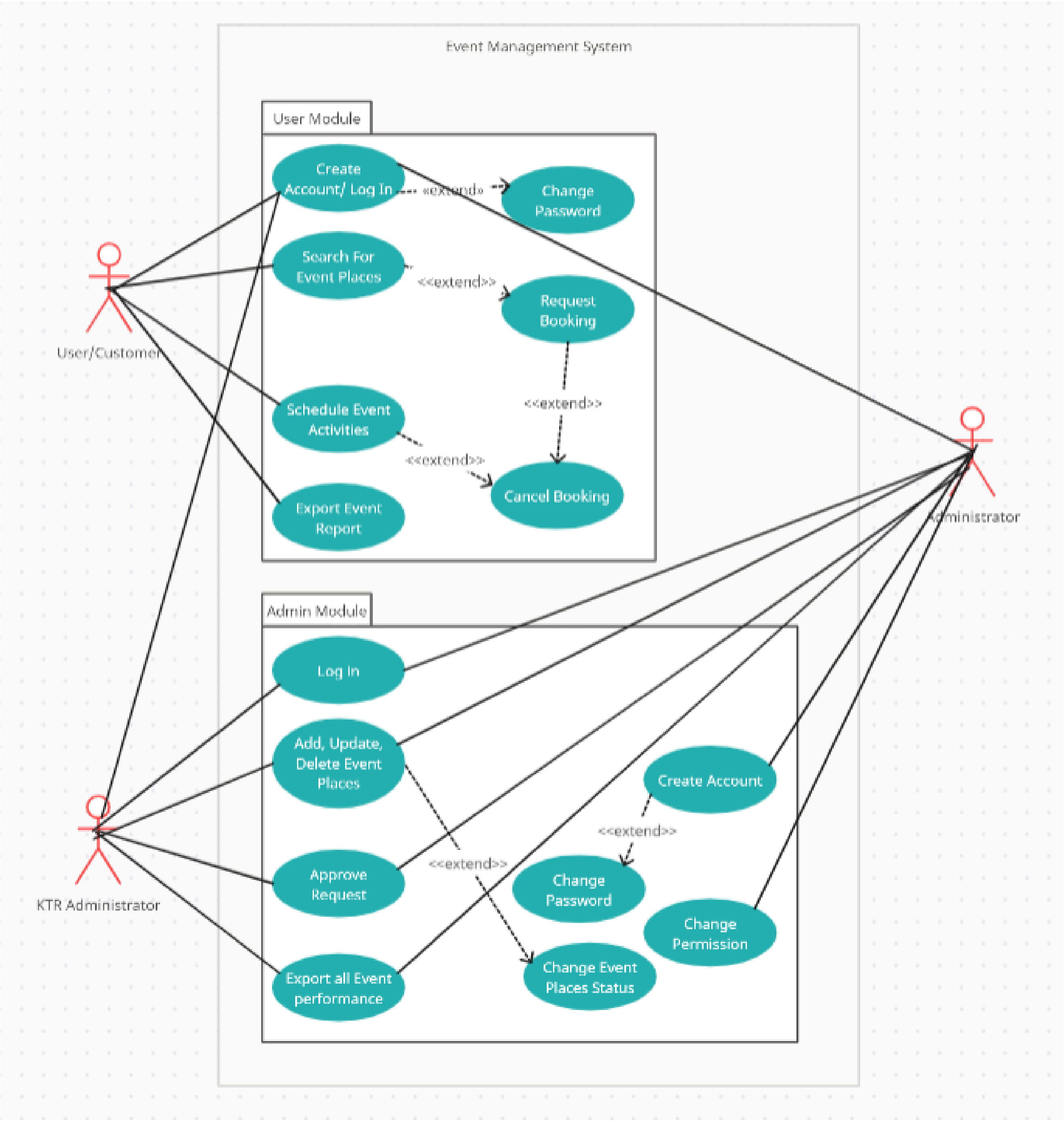


Figure 2.1: Use Case Diagram for Event Management System

Module	Function	Description
User Module	UC001 - Create Account	This use case allows students to sign up as a user for the web application.
	UC002 - Change Password	If students forget their password, they can request to change it after authentication checking using email.
	UC003 - Search for Event Places	This use case allows students to search for available Event Places either with name ,data range or time range.
	UC004 - Request Booking	Students can request for booking an Event Place for a range of time or date.
	UC005 - Cancel Booking	If students want to change the date for their Event or cancel their event, they can cancel their booking and choose another place.
	UC006 - Schedule Event Activities	Students can arrange their own activities timetable or get help from others third party for planning their activities content.
	UC007 - Export Event Report	Students are able to see their Event performance after the event is completed.
Admin Module	AC001 - Create Account	Only System administrator can create admin account in Admin Module
	AC002 - Change Password	Only System administrators can change the admin's account password.
	AC003 - Change Permission	Only System administrators can add, change, delete permission for each admin in a different section/page.
	AC004 - Log in	Administrators need to log in into the system before they can modify or get information from anything.
	AC005 - Manage Event Places	Administrators can modify available event places, activities and packages that can be seen at the user page.
	AC006 - Change Event Places Status	Administrators can change the event place status to prevent multiple booking on a place at the same time or close the area when under construction.

	AC007 - Approve Request	Administrators can approve or reject user requests on booking a place or ordering an activity.
	AC008 - Export all Event Performance	Administrators can check on their monthly/ yearly event performance and income if available.

Table 2.1: Description of Module and Functions for Event Management System

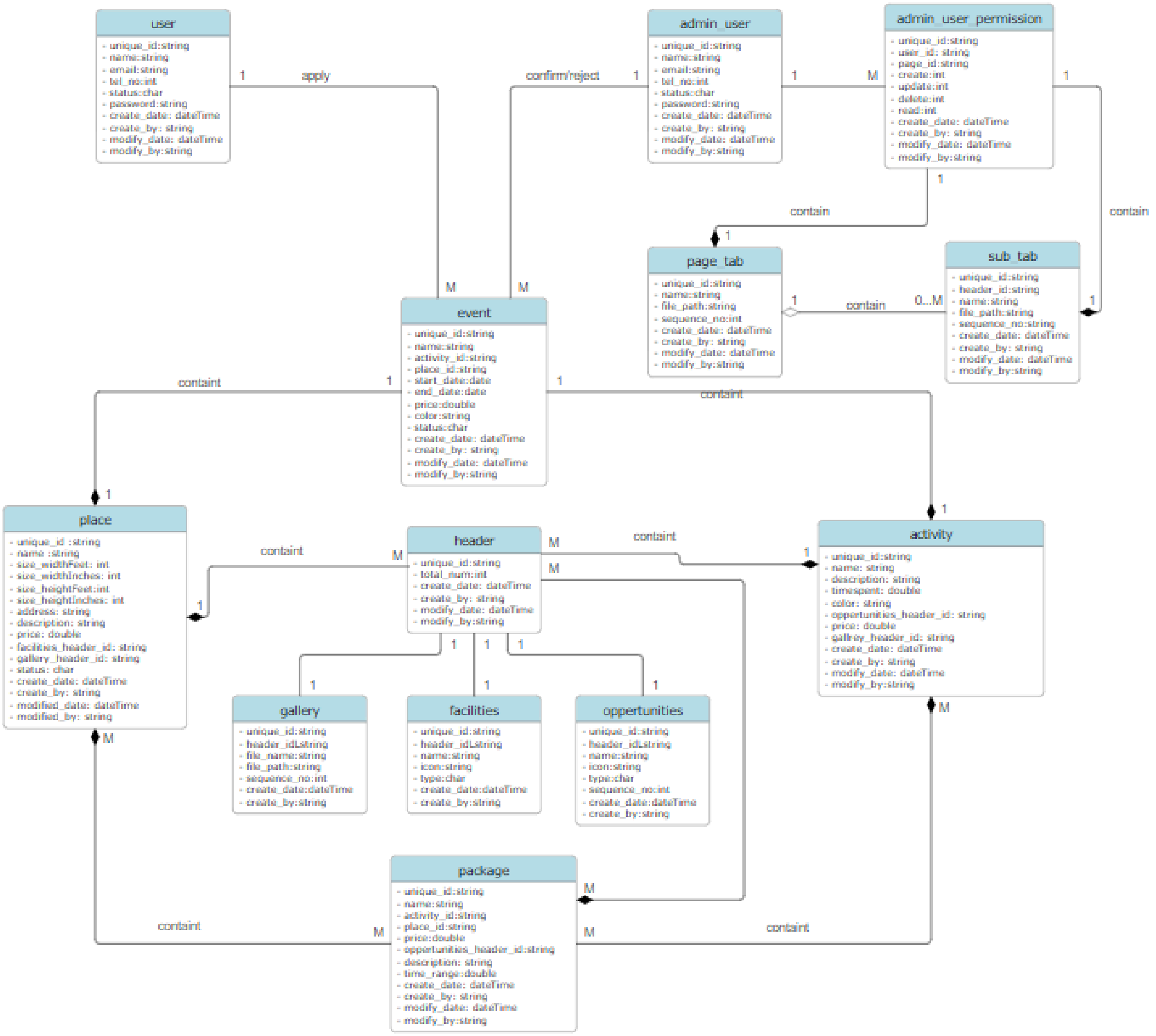


Figure 2.2: Domain Model for Event Management System

The figure above shows the domain model for Event Management System describing the relationship between objects. System Users can apply one or more than one event. After system users apply the event, the System Administrator will approve/reject each event. Every system administrator has different permissions for each page tab and subtab of the page tab. Each user permission contains one user and one page or sub tab. A user can have multiple permissions but one permission can only have one user. Each permission can contain either one page tab or one subtab, a page tab can contain one or many subtabs, and subtab only can link to one page tab.

In each event,contain one place and one activity. Package is a discount sale of one activity combined with a page. When users choose a package in their event, it will save as one activity and one place in their event request with a discount price of the package. Place, activity and package have one or more header which each header id link to one gallery picture, or one facilities details or one opportunities information.

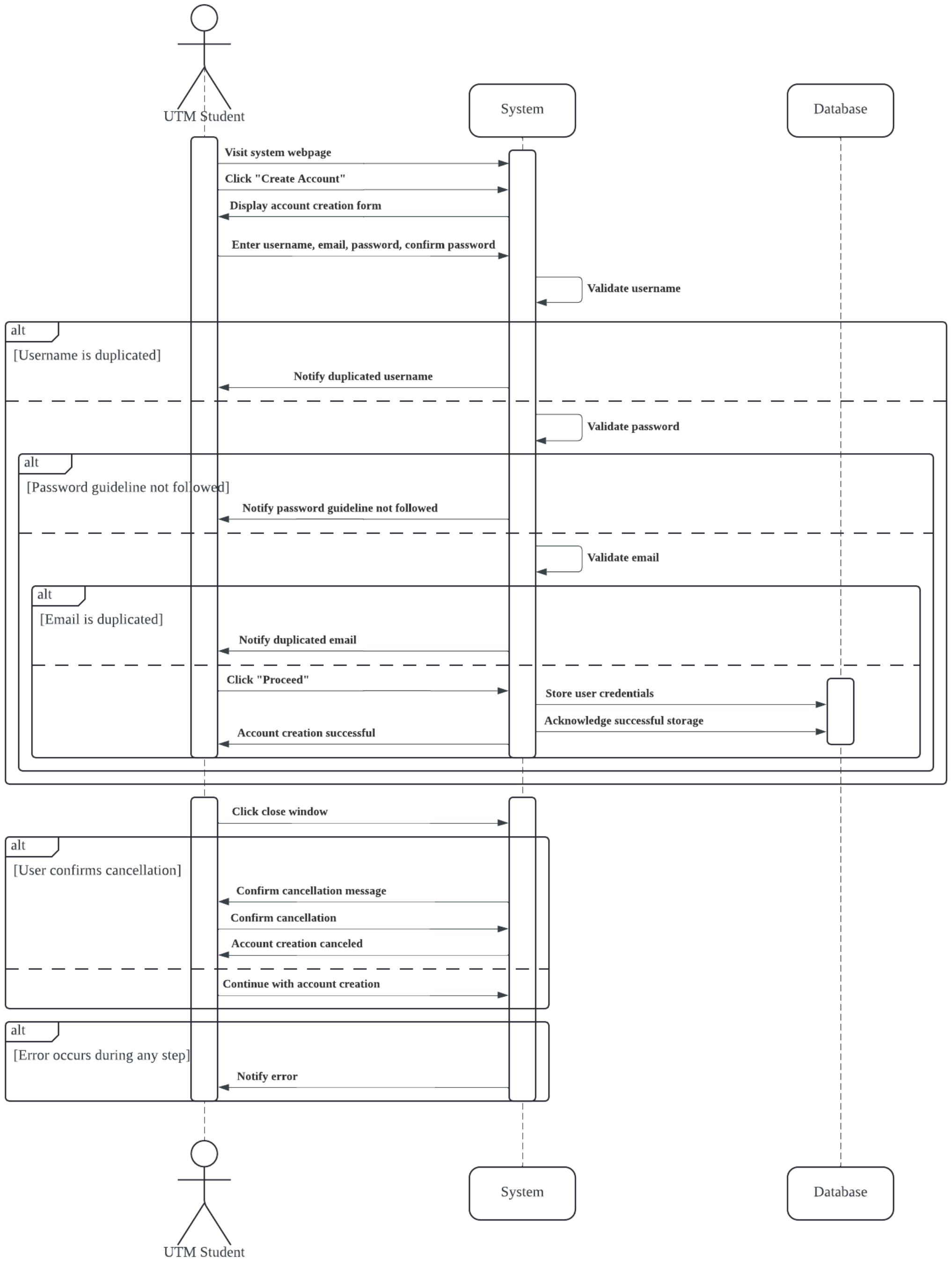
For every object, it will record all the create date, create user, modify date and modify user information for recording the change and who is to blame when there's a problem. Users in the user page are saved as user objects while users in the admin page are saved as admin_user objects. Users that saved as User object cannot login into admin page, while users that saved as Admin_user object also cannot login into user page unless create a new account as user object and use it to login into user page.

This design of the domain model will allow users to have the largest flexibility on maintaining event places, activities and packages; And also provide a secure system for each user and items created in the website.

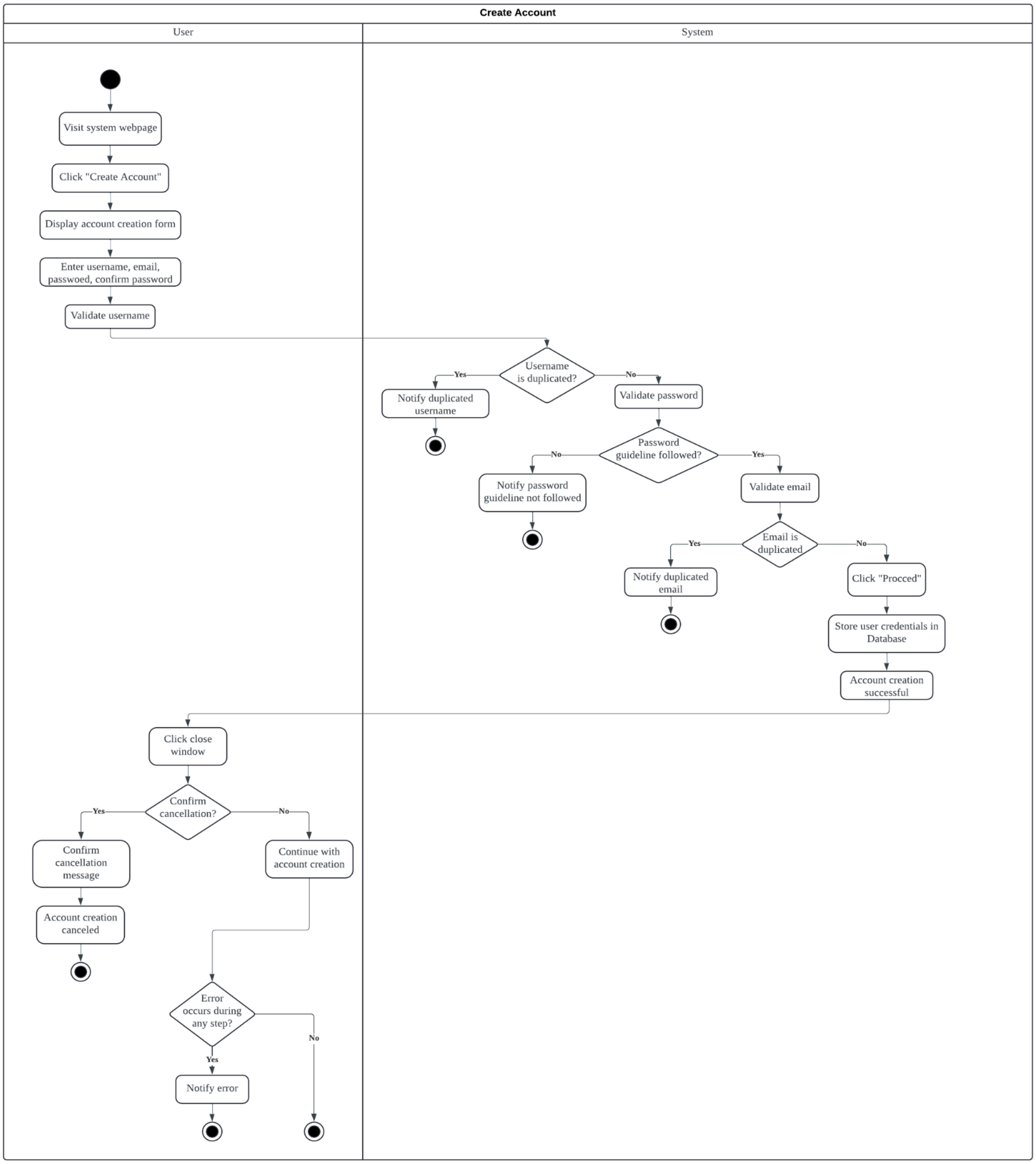
2.2.1 UC001: Use Case Create Account

Use Case ID:	UC001
Use Case Name:	Create Account
Actor:	UTM Students
Brief Description:	This use case describes the process of users creating a new account in order to use the system.
Pre-conditions:	1. User must enter all required information to create an account and use the system
Normal Flow of events:	1. Visit the system webpage. 2. Click on the "Create Account" button. 3. Key in all of the required information such as username, email, password, and confirm password. 4. Users must key in their desired username with the length of greater than 4 characters and it is not being used by other user's the system will notify if it is. 5. Users must key in their password and follow the required guideline, if the guideline it's not being followed the system will notify. 6. Users must key in their email with the correct email format. 7. Once everything has been filled in click on "proceed", in order to complete the create account process.
Post-conditions:	1. The account has been created and active for use in the system. 2. The account credentials have been stored in the database.
Alternative flow n (if any):	1. If user click the close window button for create account page, a message will prompt mentioned “Do you want to cancel this process” 2. If yes then the process will be cancel and will not update on the database
Exception flow (if any);	If a user encounters any error during the account creation process, the system will notify the user with the error they are currently facing. The potential errors they could face; 1. Duplicated username : username is currently being used by other users. 2. Password did not follow the required guideline : user intended password doesn't follow the guideline given.

	3. Duplicated email : the email has been registered in the system.


UTM Student
System
Database
Visit system webpage
Click "Create Account"
Display account creation form
Enter username, email, password, confirm password
Validate username
alt
[Username is duplicated]
Notify duplicated username
Validate password
alt
[Password guideline not followed]
Notify password guideline not followed
Validate email
alt
[Email is duplicated]
Notify duplicated email
Click "Proceed"
Store user credentials
Acknowledge successful storage
Account creation successful
Click close window
alt
[User confirms cancellation]
Confirm cancellation message
Confirm cancellation
Account creation canceled
Continue with account creation
alt
[Error occurs during any step]
Notify error
UTM Student
System
Database

Sequence Diagram for Create Account

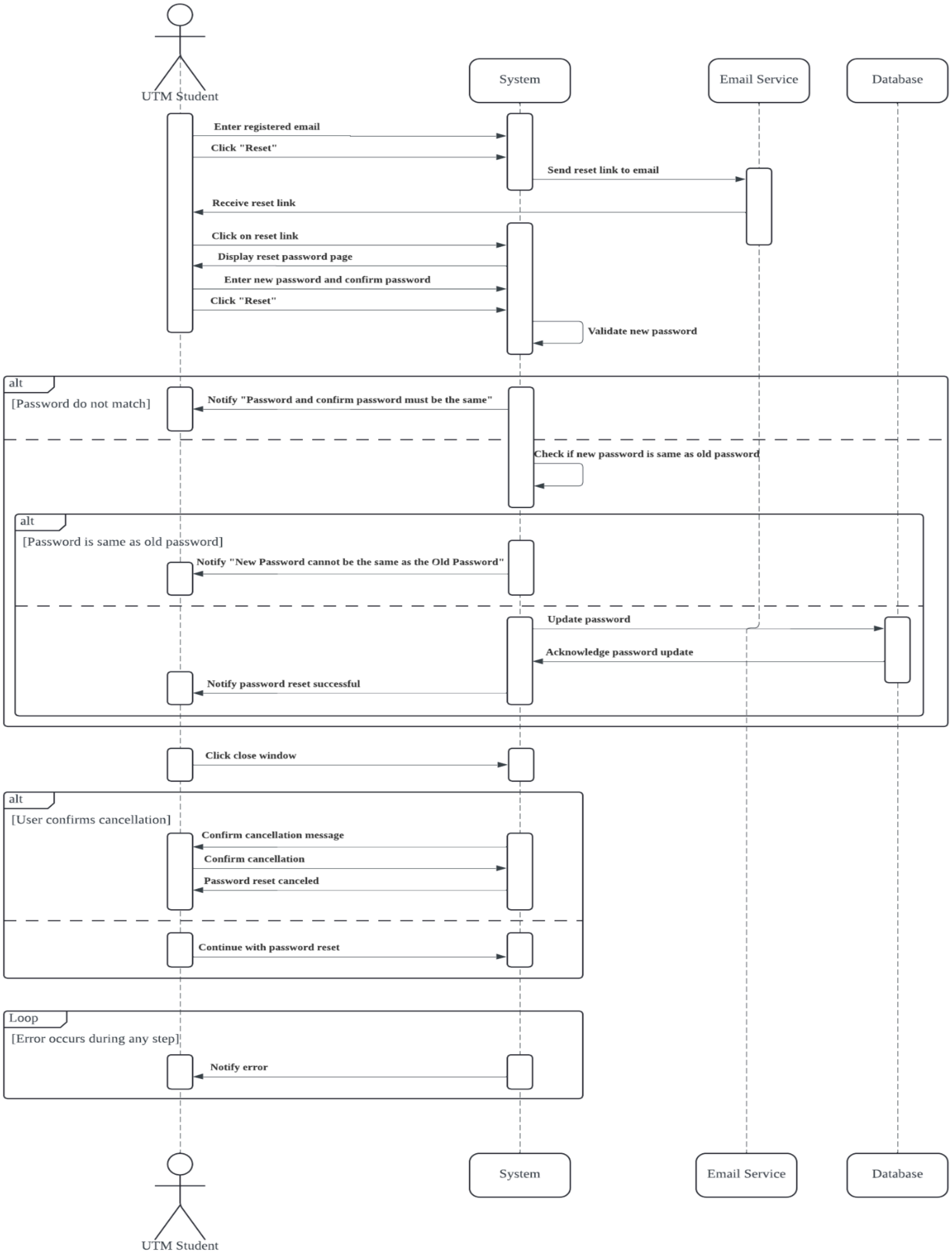

Create Account
User
System
Visit system webpage
Click "Create Account"
Display account creation form
Enter username, email, passwoed, confirm password
Validate username
Username is duplicated?
Yes
No
Notify duplicated username
Validate password
Password guideline followed?
No
Yes
Notify password guideline not followed
Validate email
Email is duplicated
Yes
No
Notify duplicated email
Click "Procced"
Store user credentials in Database
Account creation successful
Click close window
Confirm cancellation?
Yes
No
Confirm cancellation message
Continue with account creation
Account creation canceled
Error occurs during any step?
Yes
No
Notify error

Activity Diagram for Create Account

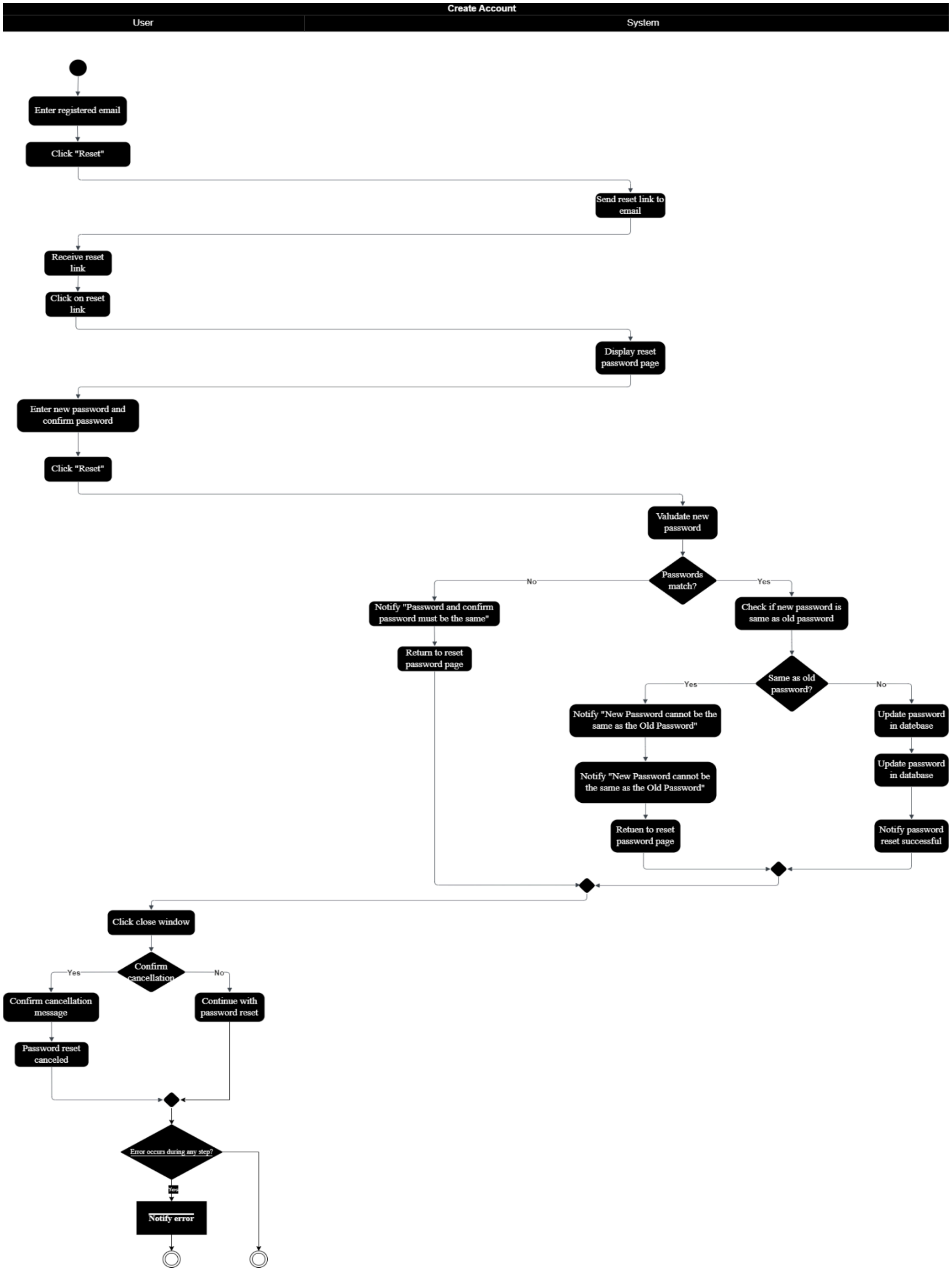
2.2.2 UC002: Use Case Change Password

Use Case ID:	UC002
Use Case Name:	Change Password
Actor:	UTM Students
Brief Description:	This use case describes the process of users changing their password for the system.
Pre-conditions:	1. User key in the email that has been registered with the system. 2. Click the "Reset" button to proceed.
Normal Flow of events:	1. Check the email for the reset link. 2. User will prompt the reset password page. 3. User key in the new password and confirm password. 4. Users click the "Reset" button. 5. System will proceed with checking if the password followed the guideline and not the old password. 6. Upon success the message will be prompt.
Post-conditions:	1. User password has been renewed with the new password. 2. User cant refresh the page to reset the password
Alternative flow n (if any):	1. If user click the close window button for reset password page, a message will prompt mentioned “Do you want to cancel this process” 2. If yes then the process will be cancel and will not update on the database
Post-conditions (if any)	1. Users cannot use the same link to change their password in the future. 2. Users cannot refresh the page to change the password.
Exception flow (if any);	1. if the new password and confirm password are not equal then it will popup an error message stating "Password and confirm password must be the same" and will direct to reset password page again. 2. If the password is the same as the old password, popup

	the error message “New Password cannot be the same as the Old Password.” and return to the reset password page.


UTM Student
System
Email Service
Database
Enter registered email
Click "Reset"
Send reset link to email
Receive reset link
Click on reset link
Display reset password page
Enter new password and confirm password
Click "Reset"
Validate new password
alt
[Password do not match]
Notify "Password and confirm password must be the same"
Check if new password is same as old password
alt
[Password is same as old password]
Notify "New Password cannot be the same as the Old Password"
Update password
Acknowledge password update
Notify password reset successful
Click close window
alt
[User confirms cancellation]
Confirm cancellation message
Confirm cancellation
Password reset canceled
Continue with password reset
Loop
[Error occurs during any step]
Notify error
System
Email Service
Database
UTM Student

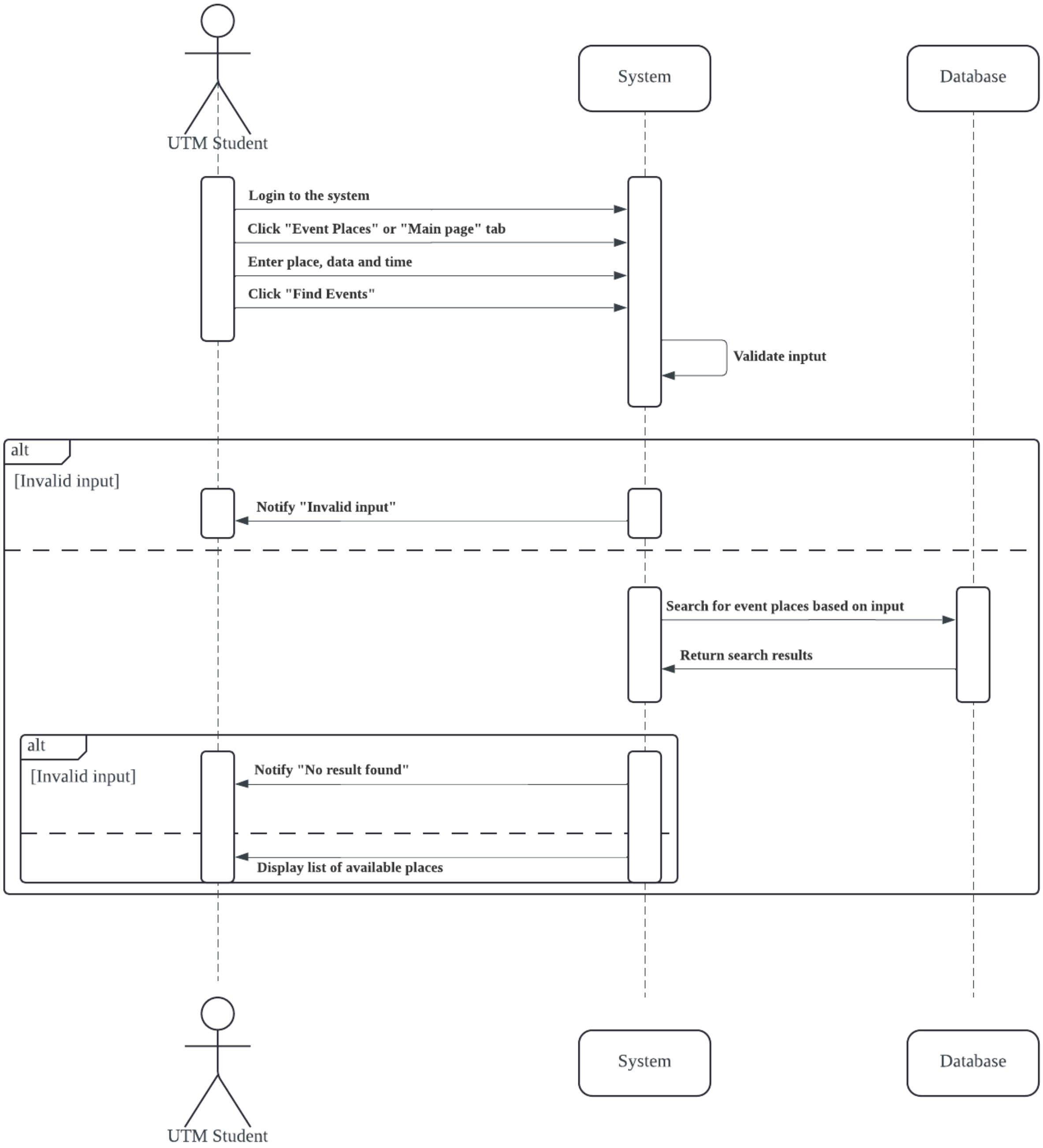
Sequence Diagram for Change Password


Create Account
User
System
Enter registered email
Click "Reset"
Send reset link to email
Receive reset link
Click on reset link
Display reset password page
Enter new password and confirm password
Click "Reset"
Valudate new password
Passwords match?
No
Yes
Notify "Password and confirm password must be the same"
Return to reset password page
Check if new password is same as old password
Same as old password?
Yes
No
Notify "New Password cannot be the same as the Old Password"
Notify "New Password cannot be the same as the Old Password"
Retuen to reset password page
Update password in datebase
Update password in database
Notify password reset successful
Click close window
Confirm cancellation
Yes
No
Confirm cancellation message
Password reset canceled
Continue with password reset
Error occurs during any step?
Yes
Notify error

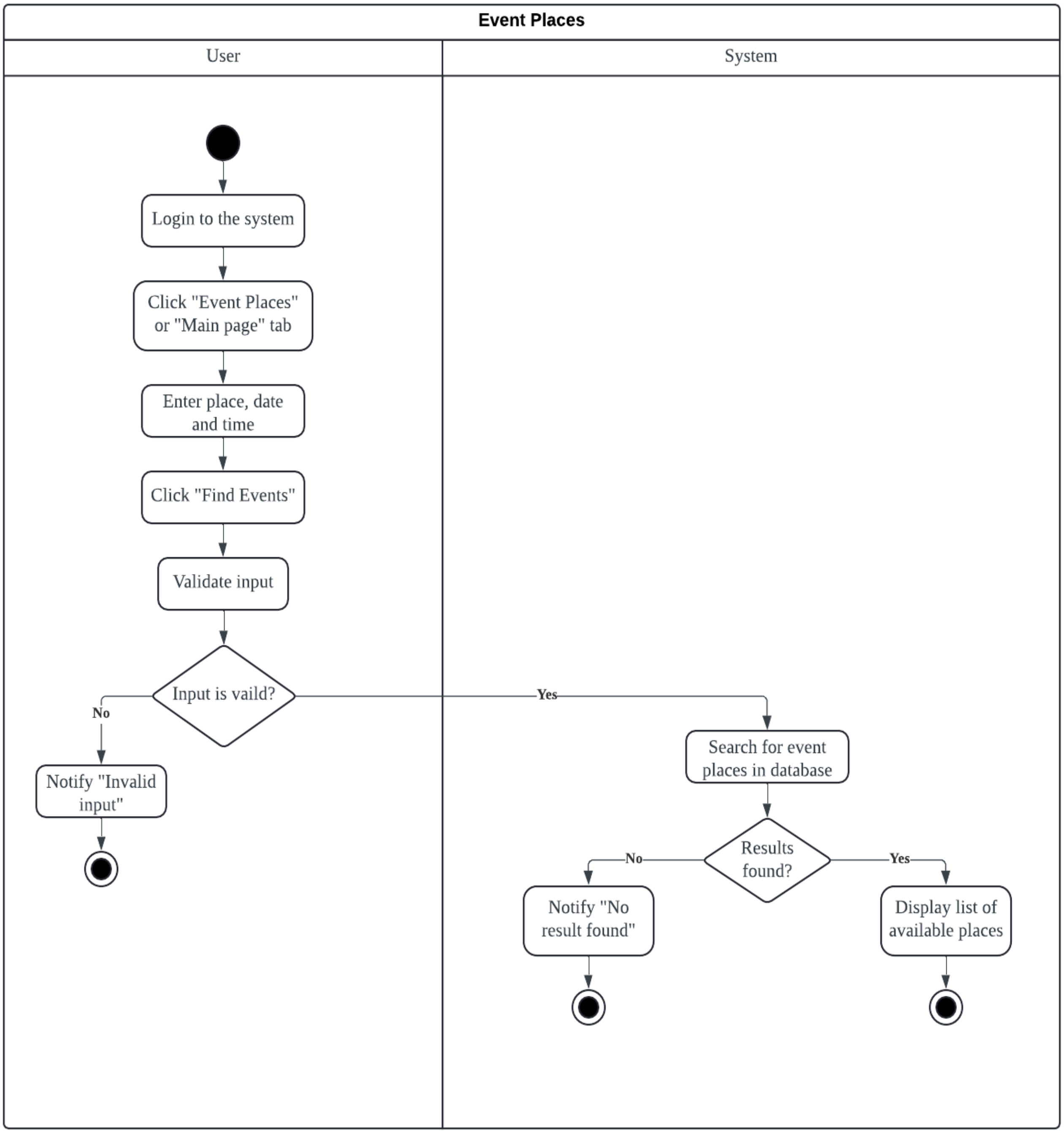
Activity Diagram for Change Password

2.2.3 UC003: Use Case Search for Event Places

Use Case ID:	UC003
Use Case Name:	Search Event Places
Actor:	UTM Students
Brief Description:	This use case describes the process for students to search any available Event Places either with name, data range or time range.
Pre-conditions:	Users need to key in their desired place, date and time for the event.
Normal Flow of events:	1. User login to the system. 2. Users click on the "Event Places" or “Main page” tab. 3. Enter the place to held the event. 4. Enter the date and time. 5. Once done click on "Find Events" . 6. It will display the list available based on user search input.
Post-conditions:	1. List of available places will be displayed based on user input.
Alternative flow n (if any):	1. If the user does not know the exact place name; a. Users can enter a partial name or leave the place field blank to search all available places
Exception flow (if any);	1. If a user enters invalid input data such as wrong date format, the system will popup an error message stating "Invalid input". 2. If there is no match based on the user input the system will prompt "No result found".



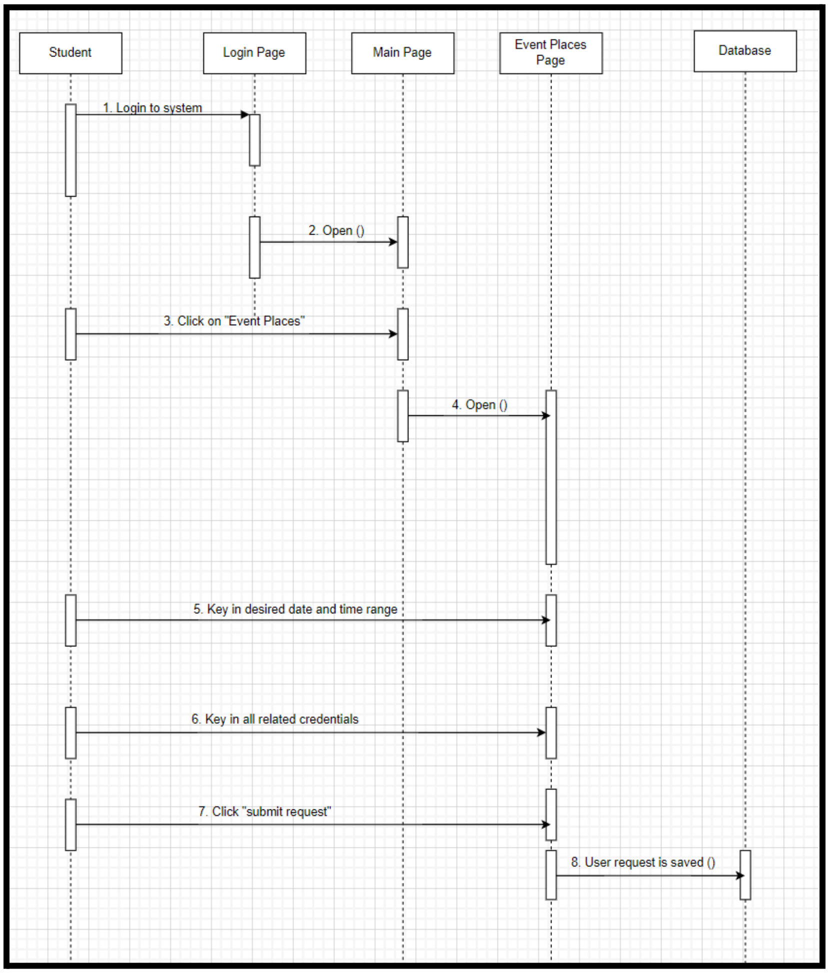
Sequence Diagram for Search Event Places



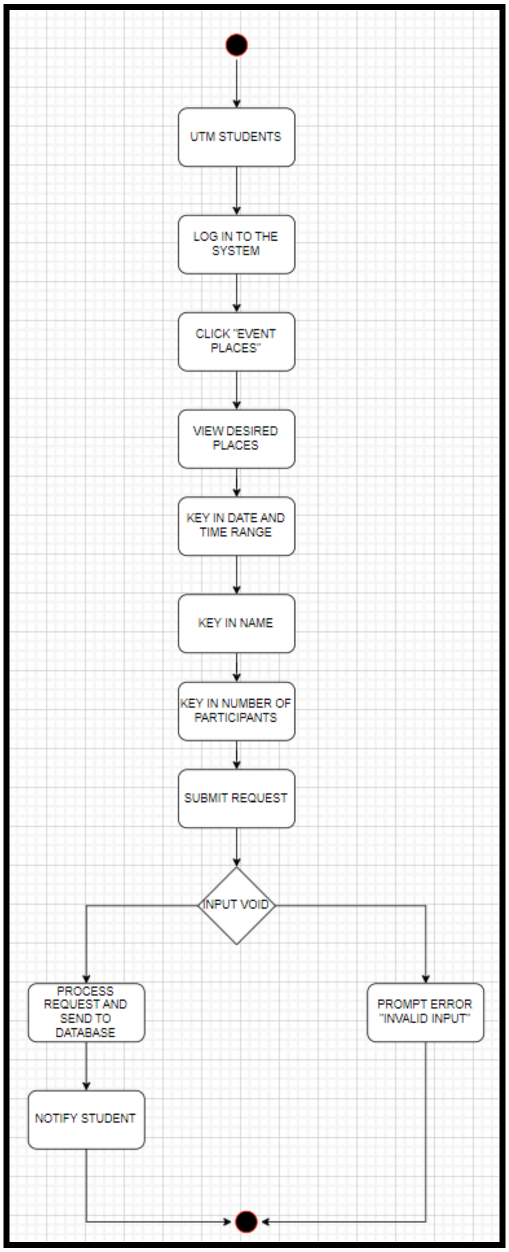
Activity Diagram for Search Event Places

2.2.4 UC004: Use Case Request Booking

Use Case ID:	UC004
Use Case Name:	Request Booking
Actor:	UTM Students
Brief Description:	This use case describes the process for students to request booking an Event place for a range of time or date.
Pre-conditions:	1. Users must log into the system to proceed with booking requests. 2. Users can search for event places based on a time and date range or choose a place first and enter the date and time range later.
Normal Flow of events:	1. Users log in to the system. 2. Users view the desired place to book by Search on the "Event Places" tab. 3. Key in the desired date and time range to held the event. 4. Key in name of requester. 5. Key in the desired date and time range. 6. Key in number of participants involved. 7. Once done click the button "Submit Request".
Post-conditions:	1. The user request has been sent to the PIC and database.
Exception flow (if any);	1. If a user inputs invalid data for example incorrect format for date or time, the system will prompt an error stating "Incorrect format" or "Invalid Input". 2. If the booking request conflicts with the existing booking, the system will suggest other available dates and times.


Student
Login Page
Main Page
Event Places Page
Database
1. Login to system
2. Open ()
3. Click on "Event Places"
4. Open ()
5. Key in desired date and time range
6. Key in all related credentials
7. Click "submit request"
8. User request is saved ()

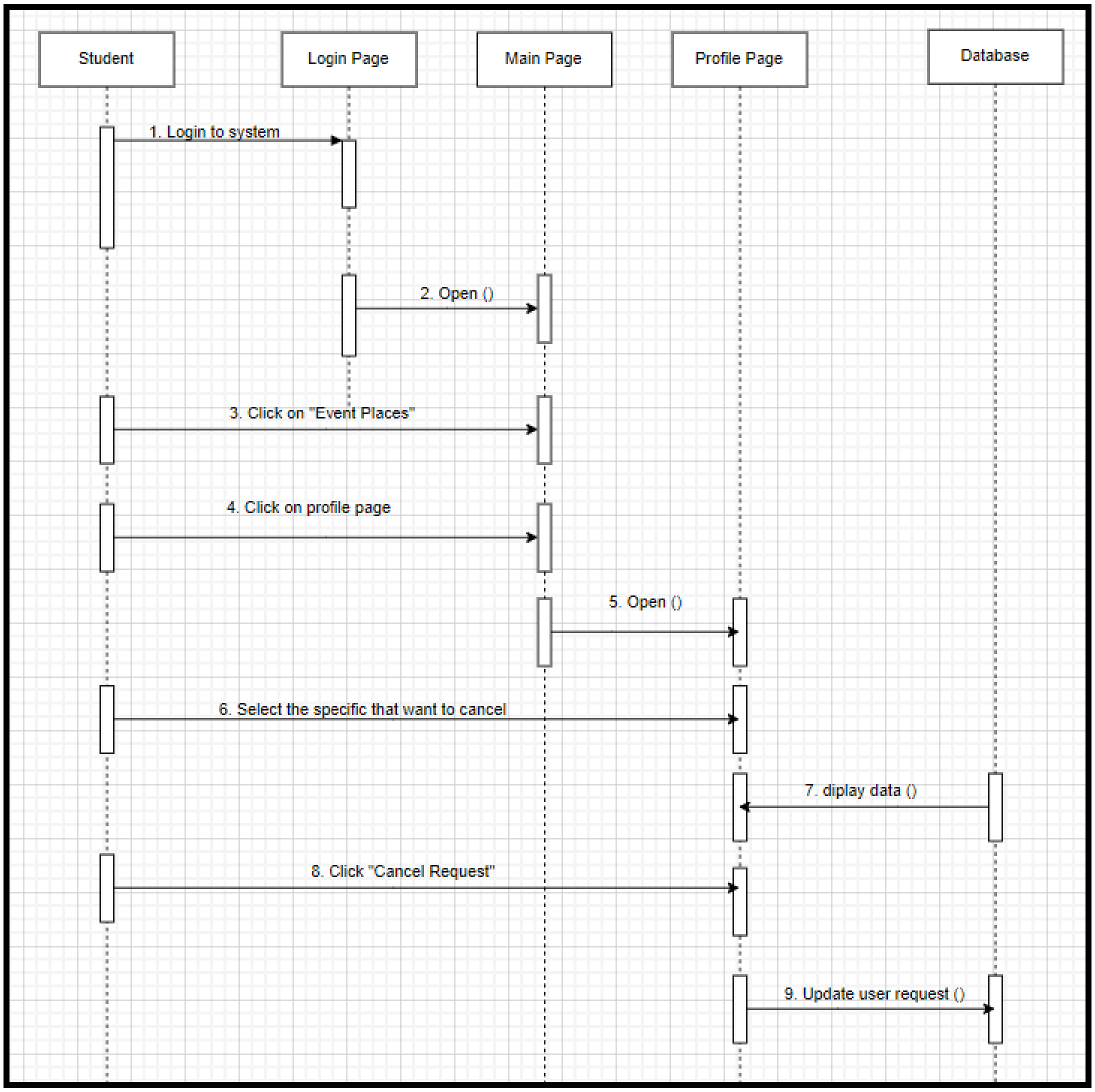
Sequence diagram for request booking



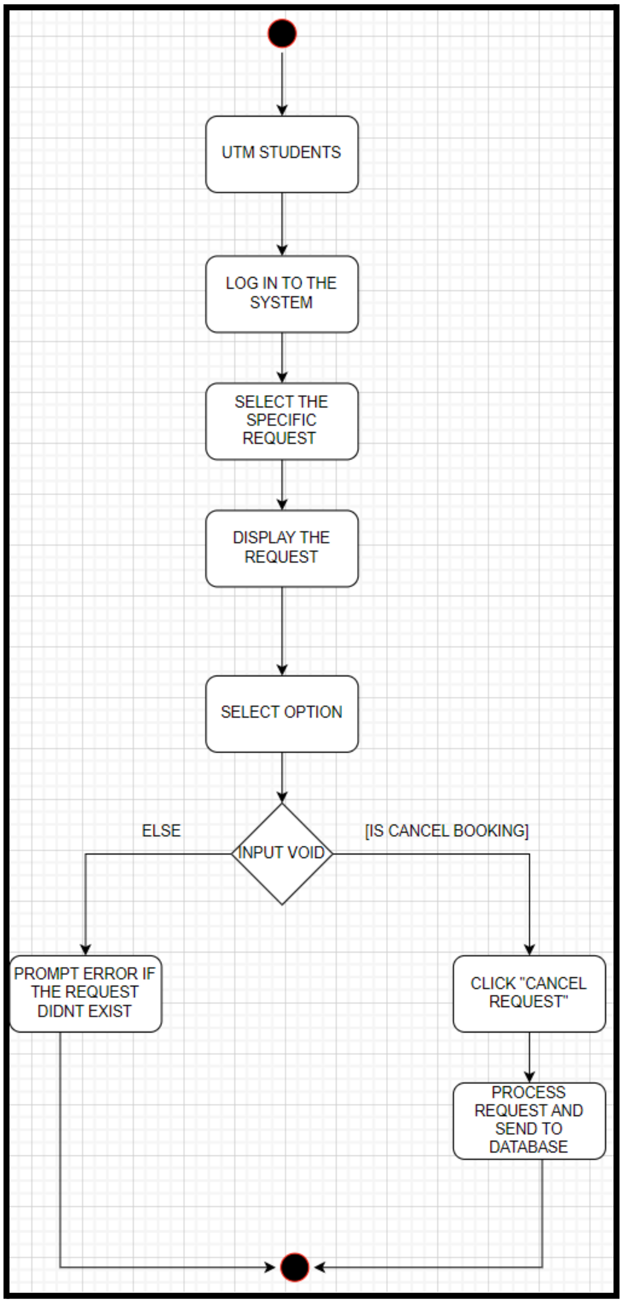
Activity Diagram for Request Booking

2.2.5 UC005: Use Case Cancel Booking

Use Case ID:	UC005
Use Case Name:	Cancel Booking
Actor:	UTM Students
Brief Description:	This use case describes the process for students who want to cancel the date for their event, allowing them to cancel their event and choose another place.
Pre-conditions:	1. Users need to be signed in to the system. 2. Users need to have an existing booking request.
Normal Flow of events:	1. Users log in to the system. 2. The system displays the list of requests that have been made by the user. 3. The user selects the specific request that they want to cancel. 4. The system details of the request and option to cancel booking. 5. User clicks "Cancel Request". 6. System will prompt "Request has been canceled".
Post-conditions:	1. The request has been canceled and updated on the database. 2. The system sends confirmation via email.
Alternative flow n (if any):	1. If the user wants to reschedule instead of canceling, the system provides an option to reschedule on step number 3.
Exception flow (if any);	1. If the user selects an invalid or non-existent request the system displays an error message.



Sequence diagram for cancel booking

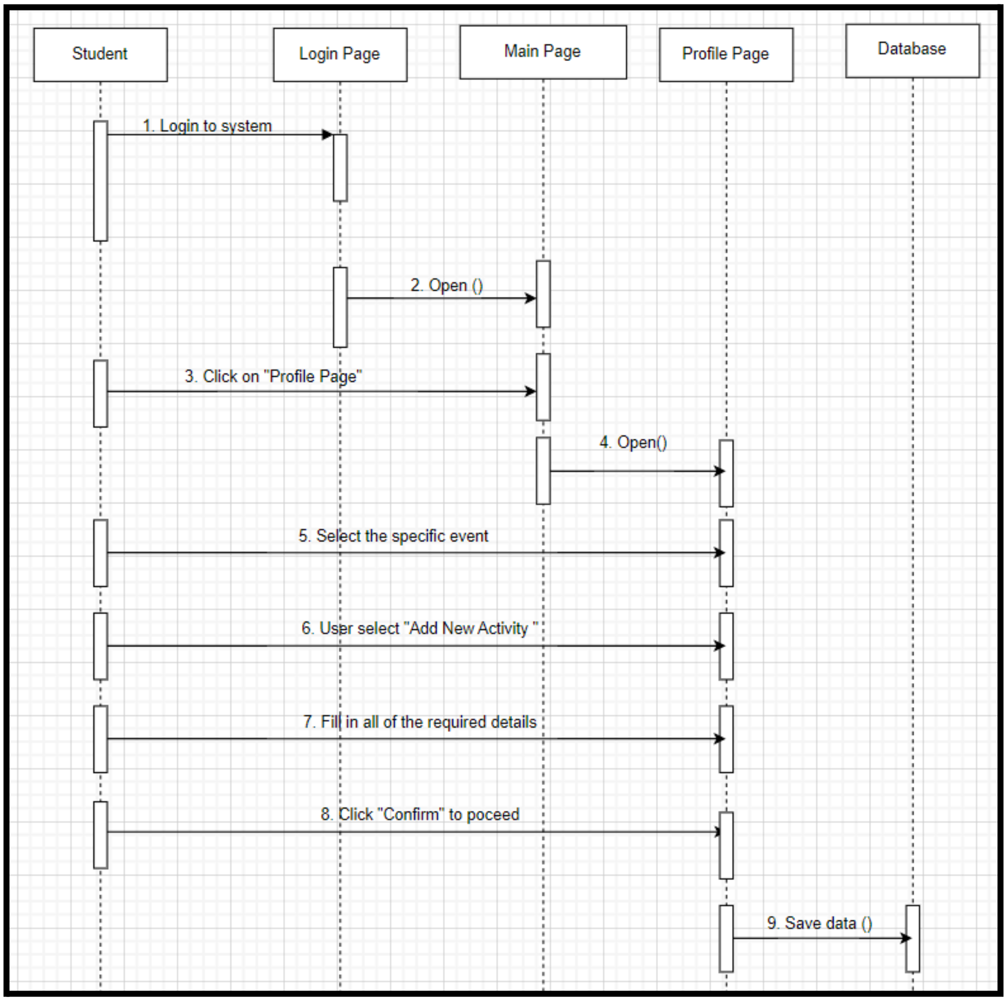


Activity Diagram for Cancel Booking

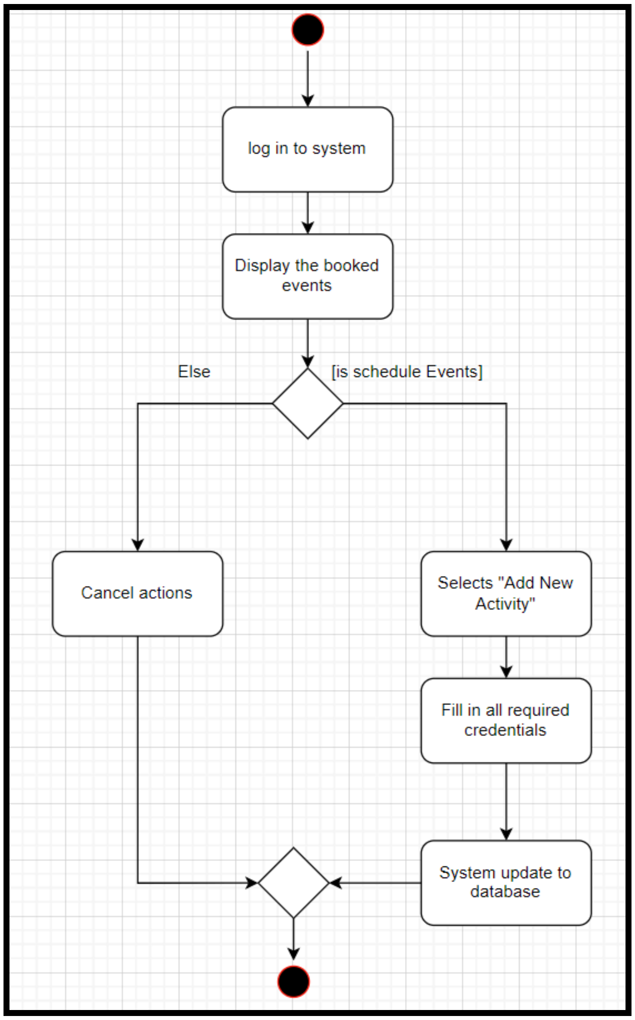
2.2.6 UC006: Use Case Schedule Event Activities

Use Case ID:	UC006
Use Case Name:	Schedule Event Activities
Actor:	UTM Students
Brief Description:	This use case outlines the process for a user to schedule activities for an event at a booked event place within the event management system. Activities could include presentations, workshops, entertainment, etc.
Pre-conditions:	The user needs to have a confirmed booking for the event place.
Normal Flow of events:	1. The user logs into the system. 2. The system displays a list of booked event places associated with the user's account. 3. The user selects the specific event for which they want to schedule activities. 4. The system presents options for scheduling event activities, such as adding new activities or editing existing ones. 5. The user selects the "Add New Activity" option. 6. The system prompts the user to enter details for the new activity, including the activity name, date, time, duration, description, and any other relevant information. 7. The user fills in the required details for the new activity and confirms the scheduling. 8. The system updates the event schedule with the newly added activity.
Post-conditions:	1. The new activity is successfully scheduled for the specified date and time within the event. 2. The event schedule is updated to reflect the newly added activity.
Alternative flow n (if any):	1. If the user wants to edit or delete an existing activity instead of adding a new one, they select the corresponding option in step 4. 2. The system prompts the user to select the activity they wish to edit or delete. 3. The user makes the necessary changes or confirms the deletion. 4. The system updates the event schedule accordingly.

Post-conditions (if any)	1. The selected activity is successfully edited or deleted from the event schedule. 2. The event schedule is updated to reflect the changes made by the user.
Exception flow (if any);	1. If there is a system error or the scheduling process cannot be completed due to technical issues, the system displays an error message. 2. The user is prompted to try again later or contact technical support for assistance. 3. The event schedule remains unchanged until the issue is resolved.


Student
Login Page
Main Page
Profile Page
Database
1. Login to system
2. Open ()
3. Click on "Profile Page"
4. Open()
5. Select the specific event
6. User select "Add New Activity "
7. Fill in all of the required details
8. Click "Confirm" to poceed
9. Save data ()

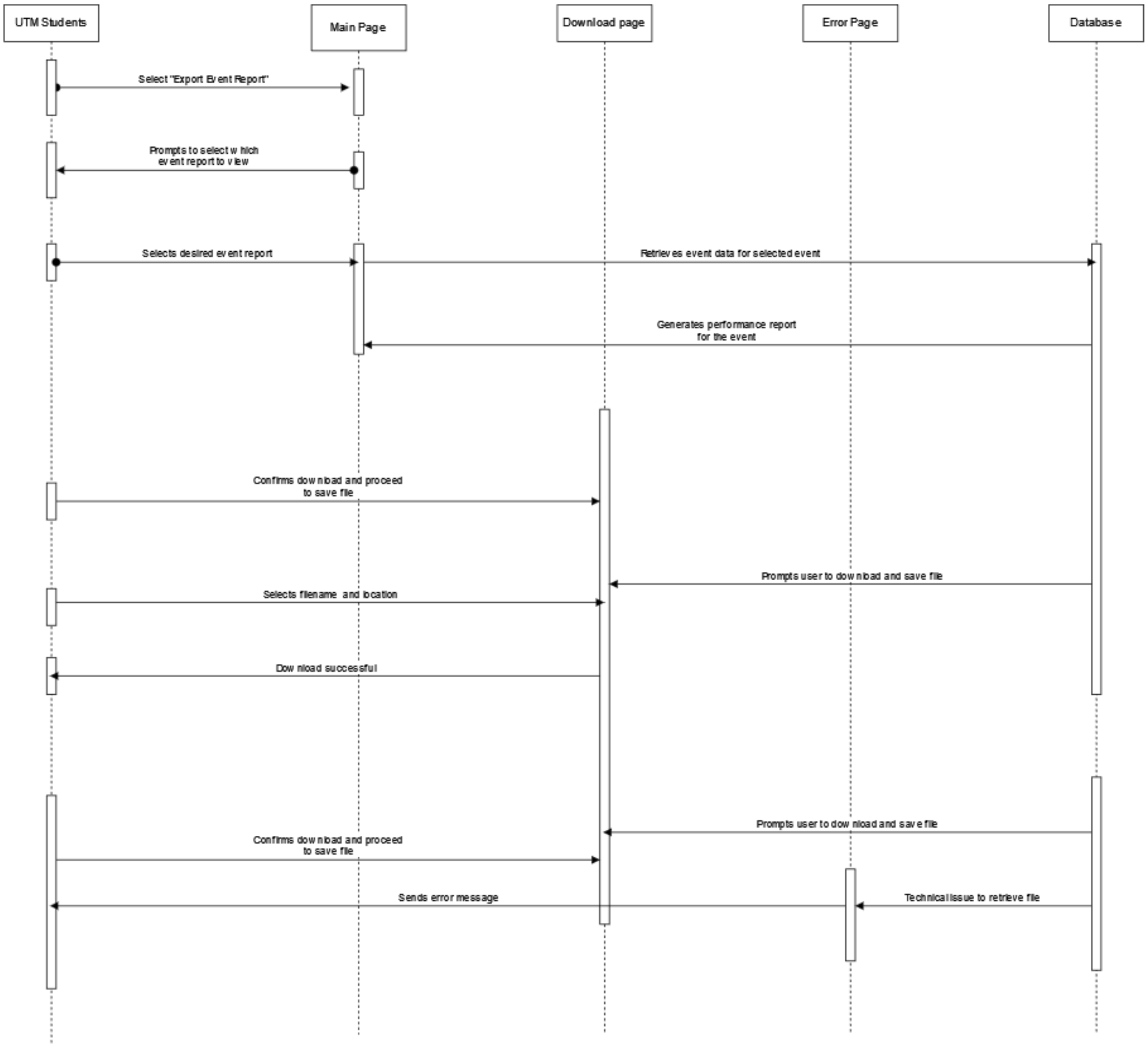
Sequence Diagram for Schedule Events Activities


log in to system
Display the booked events
Else
[is schedule Events]
Cancel actions
Selects "Add New Activity"
Fill in all required credentials
System update to database

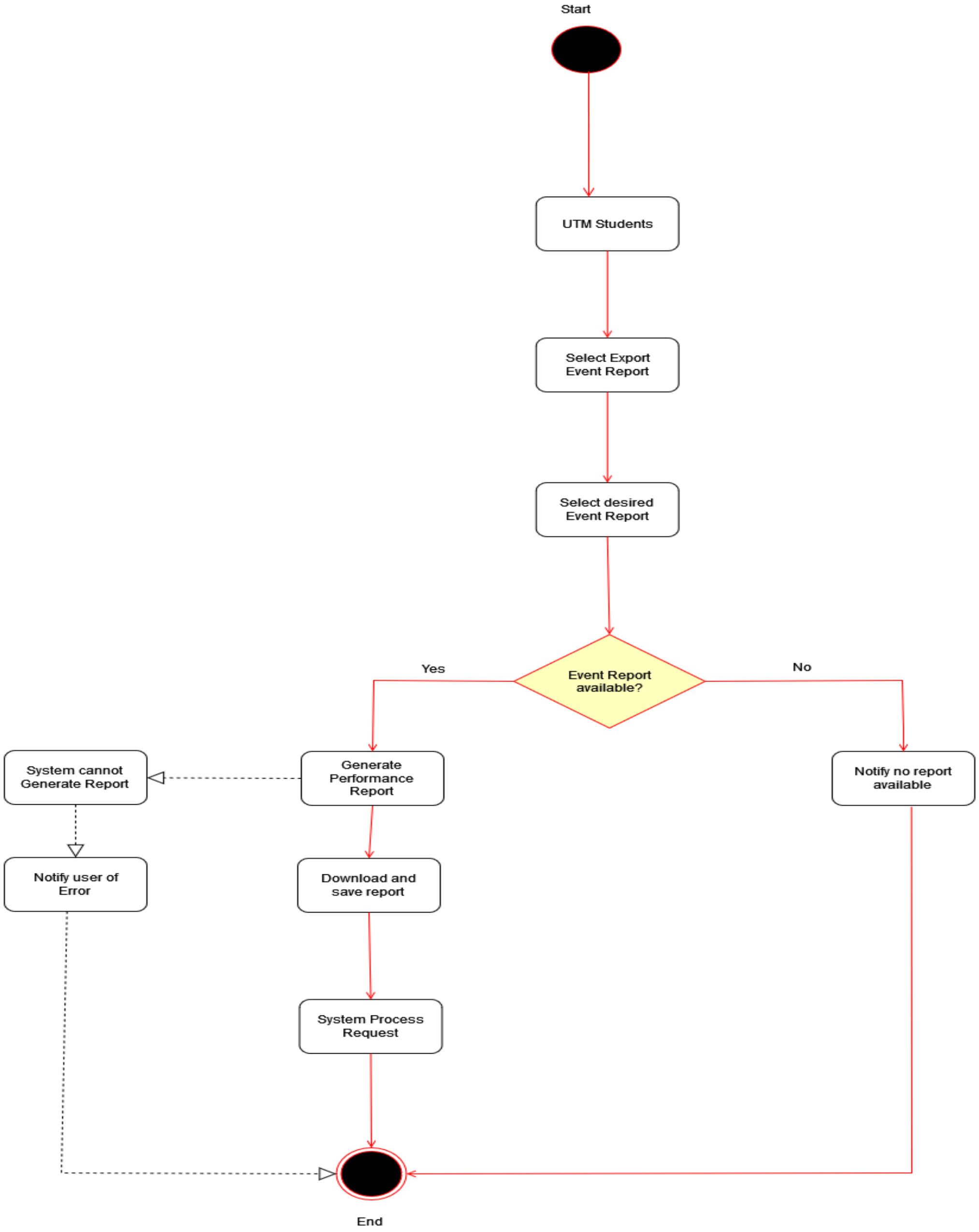
Activity Diagram for Schedule Events Activities

2.2.7 UC007: Use Case Export Event Report

Use Case ID:	UC007
Use Case Name:	Export Event Report
Actor:	UTM Students
Brief Description:	The use case outlines the process by which UTM students are able to view the Export Event Report after they have completed their Event Performance.
Pre-conditions:	1. The user must complete an event to view the event report.
Normal Flow of events:	1. The user selects the ‘Export Event Report’. 2. The system prompts to select which event report to view. 3. The user selects their desired event report. 4. The system generates a performance report for the desired event of the user. 5. The user can download the report and choose to save the file in their device. 6. The system processes the request and lets the user successfully download the report.
Post-conditions:	1. The user cancels to view the report and returns back to the system dashboard. 2. The system processes the cancellation request of the user.
Alternative flow n (if any):	1. If there is no event by the user then the system will notify the user that no report is available.
Post-conditions (if any)	1. If the system fails because of a technical issue or server issue then the system will notify the user that “Server error. Please Try again later”.


UTM Students
Main Page
Download page
Error Page
Database
Select "Export Event Report"
Prompts to select which event report to view
Selects desired event report
Retrieves event data for selected event
Generates performance report for the event
Confirms download and proceed to save file
Prompts user to download and save file
Selects filename and location
Download successful
Prompts user to download and save file
Confirms download and proceed to save file
Sends error message
Technical issue to retrieve file

Sequence Diagram for Export Event Report


Start
UTM Students
Select Export Event Report
Select desired Event Report
Event Report available?
Yes
No
Generate Performance Report
System cannot Generate Report
Notify user of Error
Download and save report
System Process Request
Notify no report available
End

Activity Diagram for Export Event Report

2.2.8 AC001: Use Case Create Account

Use Case ID:	AC001
Use Case Name:	Create Account
Actor:	System Administrator
Brief Description:	The use case describes creating a new account for admin users in the admin page with only one who has the permission to access.
Pre-conditions:	1. The user must log in into the admin page using an admin account that has the permission to access the create account page.
Normal Flow of events:	1. Admin opens the create account page. 2. Admin fill in a new account username, email ,password and confirm password then click ‘Create’ button to submit the data into the database. 3. System saves the data into the database. 4. Popup a success message at the front end.
Post-conditions:	1. The created account is active and can be used in the admin page(ONLY). 2. The account appears in the database. 3. Admin is able to use the account to log in into the system using the correct username and password. 4. Admin can also change their password using the email registered by getting the authorized confirmation thru email. 5. The input field is cleared.
Exception flow (if any);	If there is an error on creating an account, the system should display an error message with a clear description of each error and stop the process to prevent data being saved into the database. 1. Duplicated username: “The username already exists! Try another username or add number behind the username” 2. Username start with number: “The username should start with any character between A-Z” 3. Username character is less or equal to 4: “The username should be longer than 4 characters.” 4. Duplicated email: “The email is already registered!” 5. Password and Confirm Password difference: “The confirm password should be the same as the password.”

	6. Random error : “There is an error upon passing data. Please try again.”

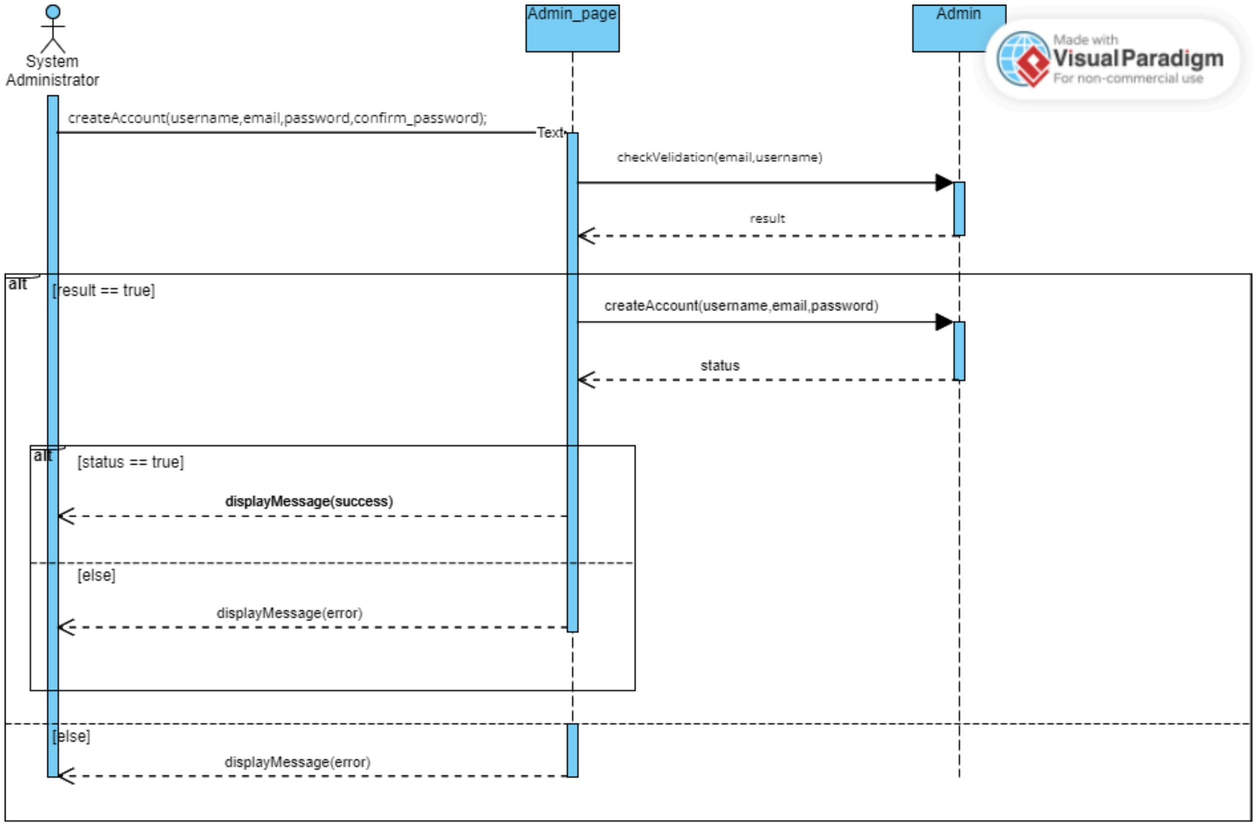

System Administrator
Admin_page
Admin
Made with Visual Paradigm For non-commercial use
createAccount(username,email,password,confirm_password);
Text
checkVelidation(email,username)
result
alt
[result == true]
createAccount(username,email,password)
status
alt
[status == true]
displayMessage(success)
[else]
displayMessage(error)
[else]
displayMessage(error)

Figure 2.2.8.1 Shows the sequence diagram for user case AC0001 Create Account.

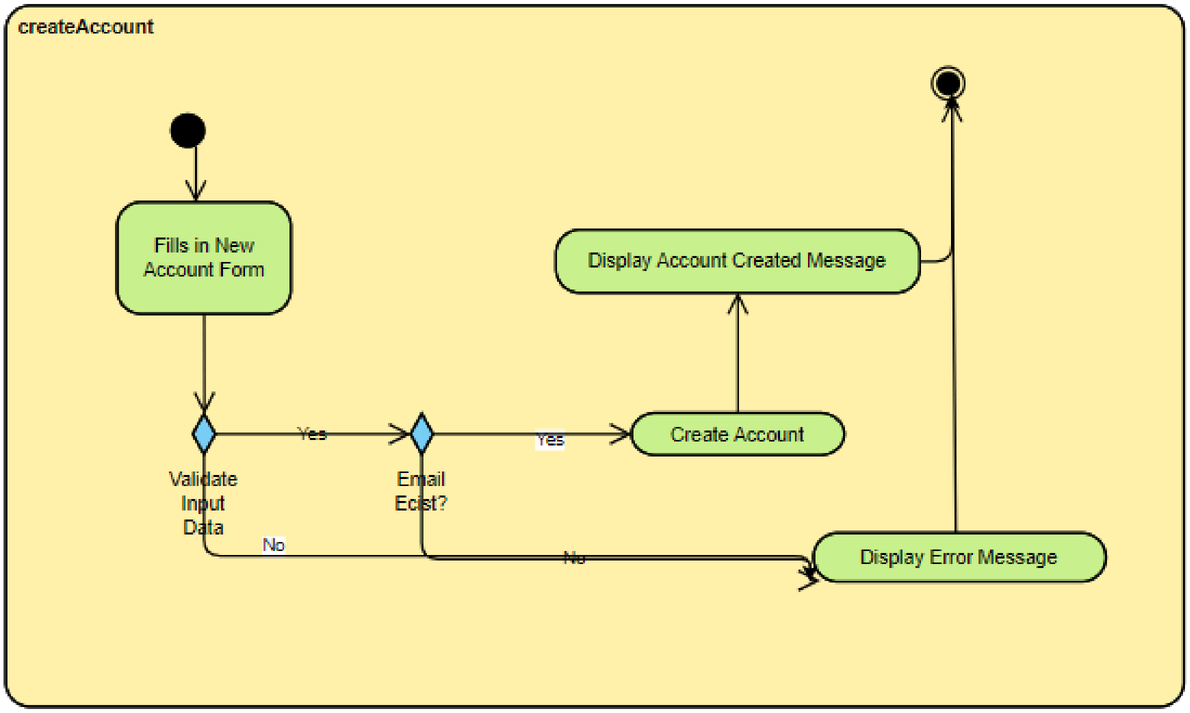


Figure 2.2.8.1 Shows the activity diagram for user case AC0001 Create Account.

2.2.9 AC002: Use Case Change Password

Use Case ID:	AC002
Use Case Name:	Change Password
Actor:	System Administrator / KTR Administrator
Brief Description:	This use case describes the process of admin users changing their password with a clear work flow of the system.
Pre-conditions:	1. User haven’t log in into their account
Normal Flow of events:	1. Users filled in the email that registered with the account inside the reset password box. 2. User click ‘Reset’ button 3. User click the reset link in their email 4. User key in the new password and confirm password 5. User click ‘Reset’ button 6. System update the password data into database 7. Shows pop-up success message to user at front end and end the reset password mode.
Post-conditions:	1. User password has been renewed to the new password. 2. Users cannot use the link in the email again to change their password. 3. Users cannot refresh the page to change password again.
Alternative flow n (if any):	1. If the user clicks the close button for reset password mode, a popup message "Do you want to cancel the reset password section?” 2. If yes, the reset password window is closed, and the link for reset password is discarded.
Post-conditions (if any)	1. User password does not have any changes. 2. Users cannot use the link in the email again to change their password. 3. Users cannot refresh the page to change password again
Exception flow (if any);	1. If the new password and confirm password are not equal, the popup error message ‘Password and Confirm Password must be the same to each other.’ and return to the reset password page. 2. If the password has blank space (space bar), popup the error message "Password cannot have space bar, please use ‘_’ or ‘-’ to replace the space bar or remove the space bar.” and return to the reset password page

	3. If the password is the same as the old password, popup the error message “New Password cannot be the same as the Old Password.” and return to the reset password page. 4. Random error will popup “Error upon passing data.” and return to the reset password page.

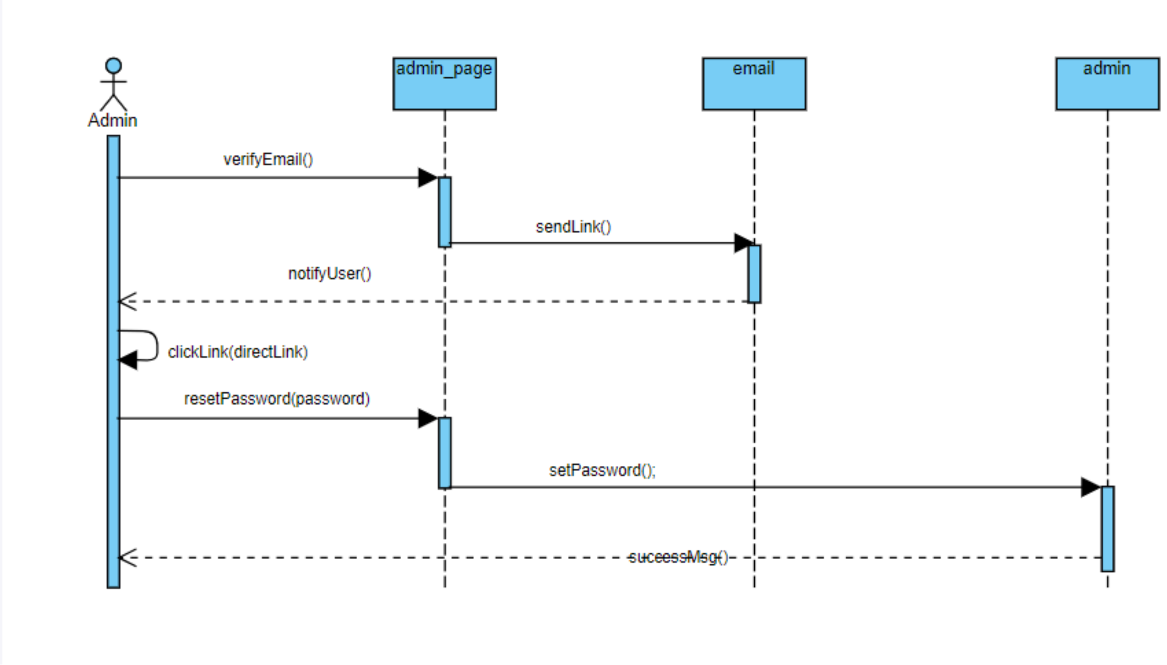


Figure 2.2.9.1 Shows the sequence diagram of AC002 Reset Password.

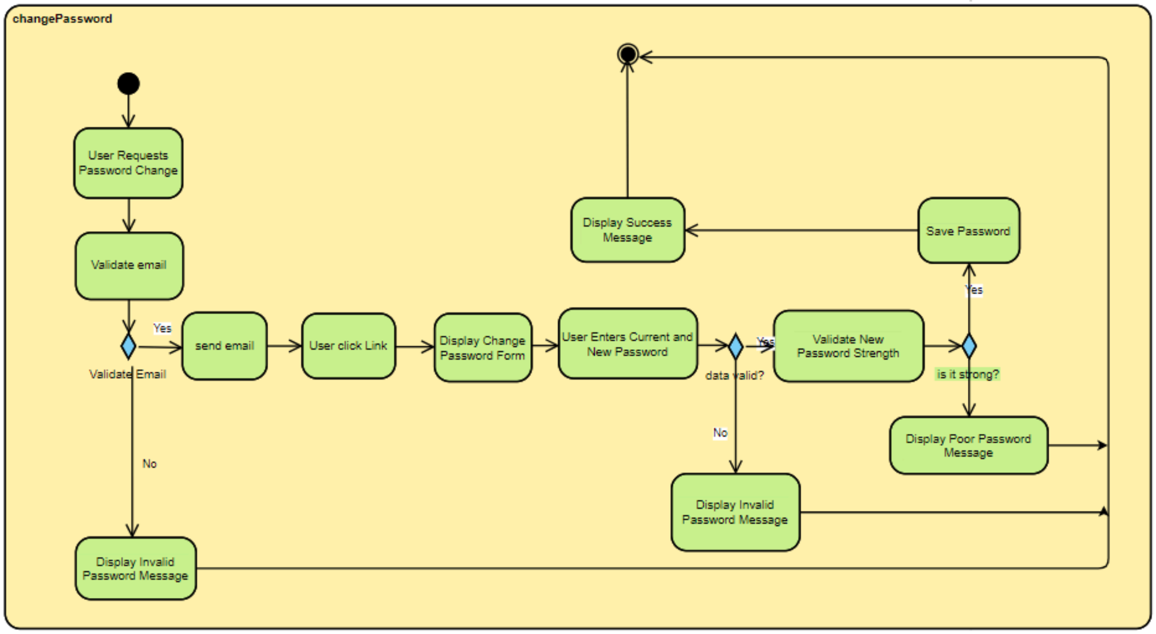

changePassword
User Requests Password Change
Validate email
Yes
Validate Email
No
send email
User click Link
Display Change Password Form
User Enters Current and New Password
data valid?
Yes
No
Validate New Password Strength
is it strong?
Yes
Save Password
Display Success Message
Display Poor Password Message
Display Invalid Password Message
Display Invalid Password Message

Figure 2.2.9.2 Shows the activity diagram of AC002 Reset Password.

2.2.10 AC003: Use Case Change Permission

Use Case ID:	AC003
Use Case Name:	Change Permission
Actor:	System Administrator
Brief Description:	In this use case, it will show the system flow and user flow of the process on changing user permission in the admin page.
Pre-conditions:	1. User login in an account that has permission to open the change permission page. (Usually system administrator or KTR manager)
Normal Flow of events:	1. User click in the Change Permission tab. 2. The logged in user checks the checkbox of ‘Read’,’Write’,’Update’ and ‘Delete’ for one of the users in the user permission list, User A. Then click the ‘Save’ button. 3. System update the user permission status for User A in the database. 4. A success message pops up on the front page, and reset the user permission list.
Post-conditions:	1. User A will now gain permission for ‘Read’, ‘Write’,’Update’, and ‘Delete’ for the selected page. 2. Permission status of User A in the permission list should change to checked in the Page list in Change Permission Page. 3. User A will be able to see the permission gained page shows in his/her side menu. 4. User A will be able to create new data for the permission gained page. 5. User A will be able to update existing data for the permission gained page. 6. User A will be able to delete existing data for the permission gained page.
Alternative flow n (if any):	1. System Administrator uncheck the permission box of ‘Read’ for User A, and click the ‘Save’ button.
Post-conditions (if any)	1. User A will not be able to see the page appear in his/her side menu. 2. If User A uses hard code to make the page appear or direct the page, User A will not be able to load the page list or any information of the page.

Alternative flow n (if any):	1. System Administrator checks the permission box of 'Read' for User A but uncheck the permission box of 'Write', 'Update' and 'Delete' of User A for the page, and clicks the 'Save' button.
Post-conditions (if any)	1. User A will be able to see the page in their side menu. 2. User A will be able to load the information inside the gained permission page. 3. The create button is hidden for User A, User A not able to Create new data for the page. 4. The update button is hidden for User A, User A not able to Update existing data for the page. 5. The delete button is hidden for User A, User A not able to Delete existing data for the page.
Exception flow (if any);	1. Random error shows a popup message : 'Error upon passing data.' and remain on the page.

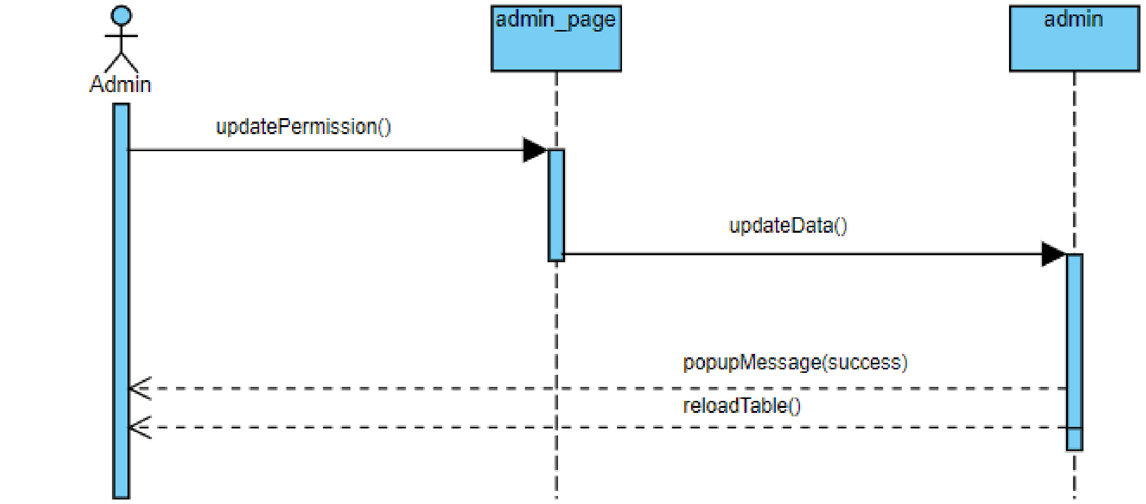


Figure 2.2.10.1 Shows the sequence diagram for AC003 Change Permission

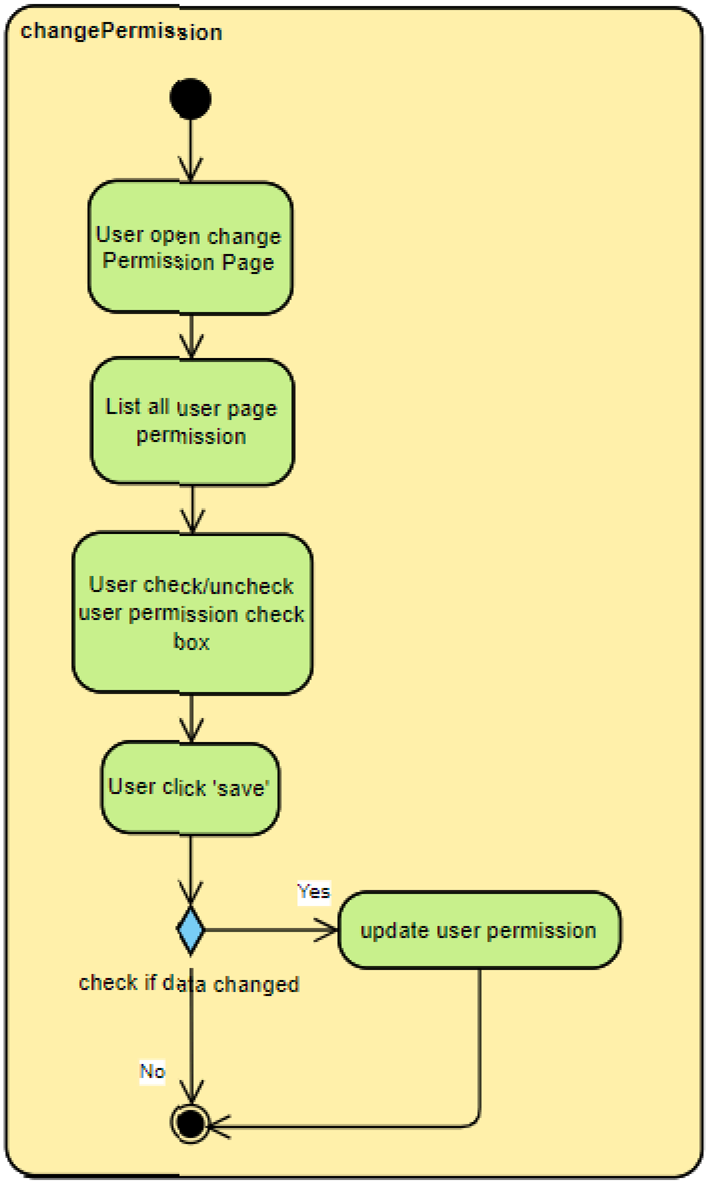

changePermission
User open change Permission Page
List all user page permission
User check/uncheck user permission check box
User click 'save'
Yes
update user permission
check if data changed
No

Figure 2.2.10.2 Shows the activity diagram for AC003 Change Permission

2.2.11 AC004: Use Case Login

Use Case ID:	AC004
Use Case Name:	User Login
Actor:	System Administrator / KTR Administrator
Brief Description:	This User Case describes the system flow and user flow for the process of login in the Admin page.
Pre-conditions:	1. User visiting the login page. 2. User not logged in yet. 3. Users already have their own admin account created by the system administrator.
Normal Flow of events:	1. User key-in username. 2. User key-in password. 3. Users click the 'Login' button. 4. System check if the username exists in the database. If yes, continue to the next step. 5. System check if the password is the same as the username saved password in the database. If yes, continue to the next step. 6. Direct user to the main page of the admin page.
Post-conditions:	1. Users can access the main page of the admin page. 2. Top-right corner where the profile picture shows the user picture, and username.
Alternative flow n (if any):	1. If the user clicks the 'Reset Password' link, direct the user to reset password page.
Exception flow (if any);	1. If the username does not exist in the database popup error message : 'Username does not exist.' 2. If the password does not match with the password saved inside the database, popup error message: 'Password not match.' 3. Random error popup error message : 'Error upon passing data.'

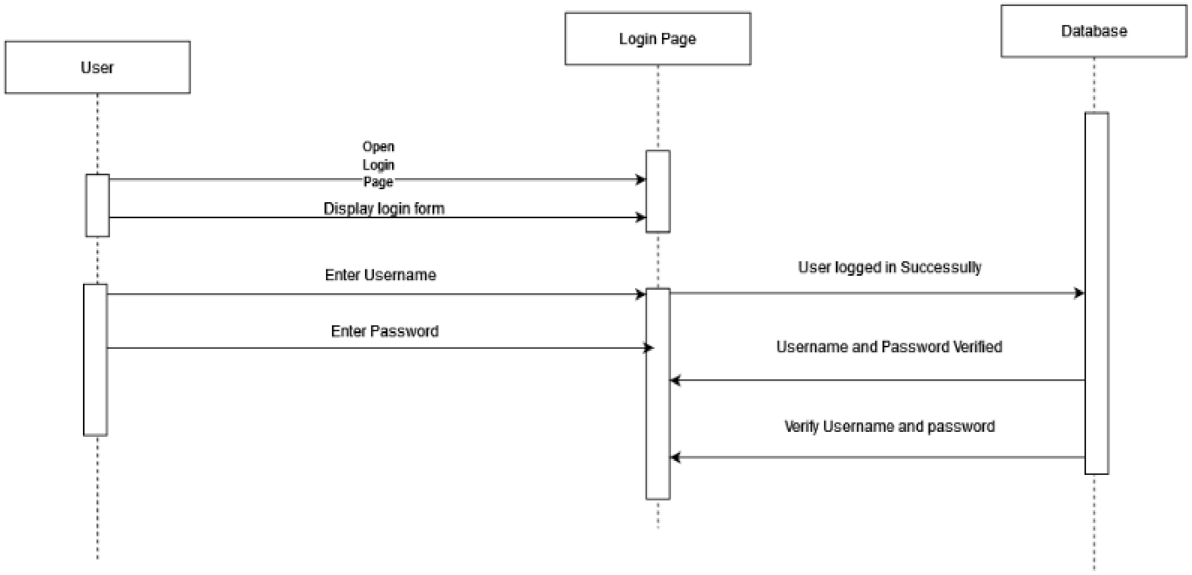

User
Login Page
Database
Open Login Page
Display login form
Enter Username
Enter Password
User logged in Successully
Username and Password Verified
Verify Username and password

Figure 2.2.11.1 Shows the sequence diagram for AC004 use case login

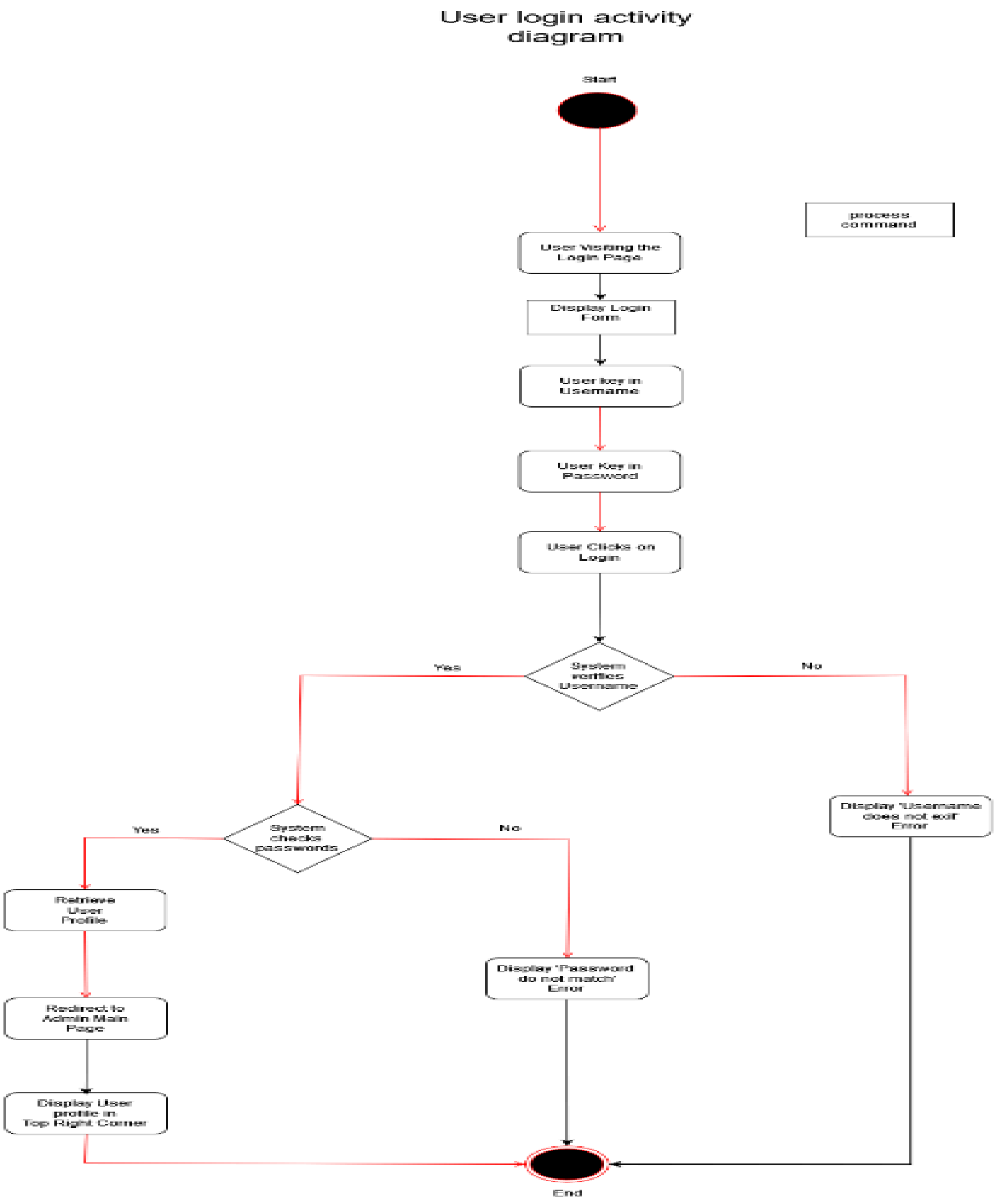

User login activity diagram
Start
process command
User Visiting the Login Page
Display Login Form
User key in Username
User Key in Password
User Clicks on Login
System verifies Username
Yes
No
Display 'Username does not exit' Error
System checks passwords
Yes
No
Retrieve User Profile
Redirect to Admin Main Page
Display User profile in Top Right Corner
Display 'Password do not match' Error
End

Figure 2.2.11.2 Shows the activity diagram for AC004 use case login

2.2.12 AC005: Use Case Manage Event Places

Use Case ID:	AC005
Use Case Name:	Manage Event Places
Actor:	System Administrator/ KTR Administrator
Brief Description:	This use case describes the working flow of the user on managing event places and how the system interacts with the user in the admin page.
Pre-conditions:	1. User login as an admin user with the permission to access the Manage Event Places page. 2. User accessed the page.
Normal Flow of events:	1. System shows the ‘Add New’ if the user has the permission to ‘Insert’. 2. System shows the list of Event Places that already exist with the details of the Event Places as below: a. Place Name b. Place Status c. Place Last Modified Date d. Search bar of Place Name e. Search bar of Place Status f. Filter button for place name, place status and last modified date. g. Edit button for each event place if the user has the permission to update. h. Delete button for each event place if the user has the permission to delete. 3. Users click the ‘Insert’ button. 4. System shows a blank form that allows the user to key-in the data for the new event place. 5. After the user clicks to submit the form, the system sends the data to the back-end. 6. System check if there is duplication on event place name, id. If not, the system proceeds to save the data in the database. Else, the system stops the process, and shows an error popup message at the front-end. 7. After saving the data, the system shows the success message at the front-end.
Post-conditions:	1. New added place shows at the list of the event places. 2. New added place exist in the database, 3. New added place if the status is active, shows in the user

	page.
Alternative flow n (if any):	1. Users click the ‘Edit’ button. 2. System shows the form with previous data inside the input box. 3. Users modify the data. 4. Users click the ‘Save’ Button. 5. System sends the data to the back-end. 6. System check if the modified data is valid or not. If it is valid save the data to the database and show a success message on the popup.Else shows error message on the popup.
Post-conditions (if any)	1. Edited place shows the updated data in the list of event places. 2. Data in database updated. 3. Edited place details updated in user page.
Alternative flow n (if any):	1. Users click the ‘Delete’ button. 2. System shows ‘ Confirm to Delete?’ popup with ‘Yes’ and ‘No’ button. 3. If Users click Yes, the system will delete the data in the backend and show a success message at the front-end. 4. If Users click No, the system will do nothing and close the Confirm to Delete popup message.
Post-conditions (if any)	1. Deleted place removed from the List of Place. 2. Deleted place removed from database. 3. Deleted place not showing in the user page.
Exception flow (if any);	1. Shows error message “Error on passing data” for random error. 2. Shows error message “Data already exists!” for data duplicate error.

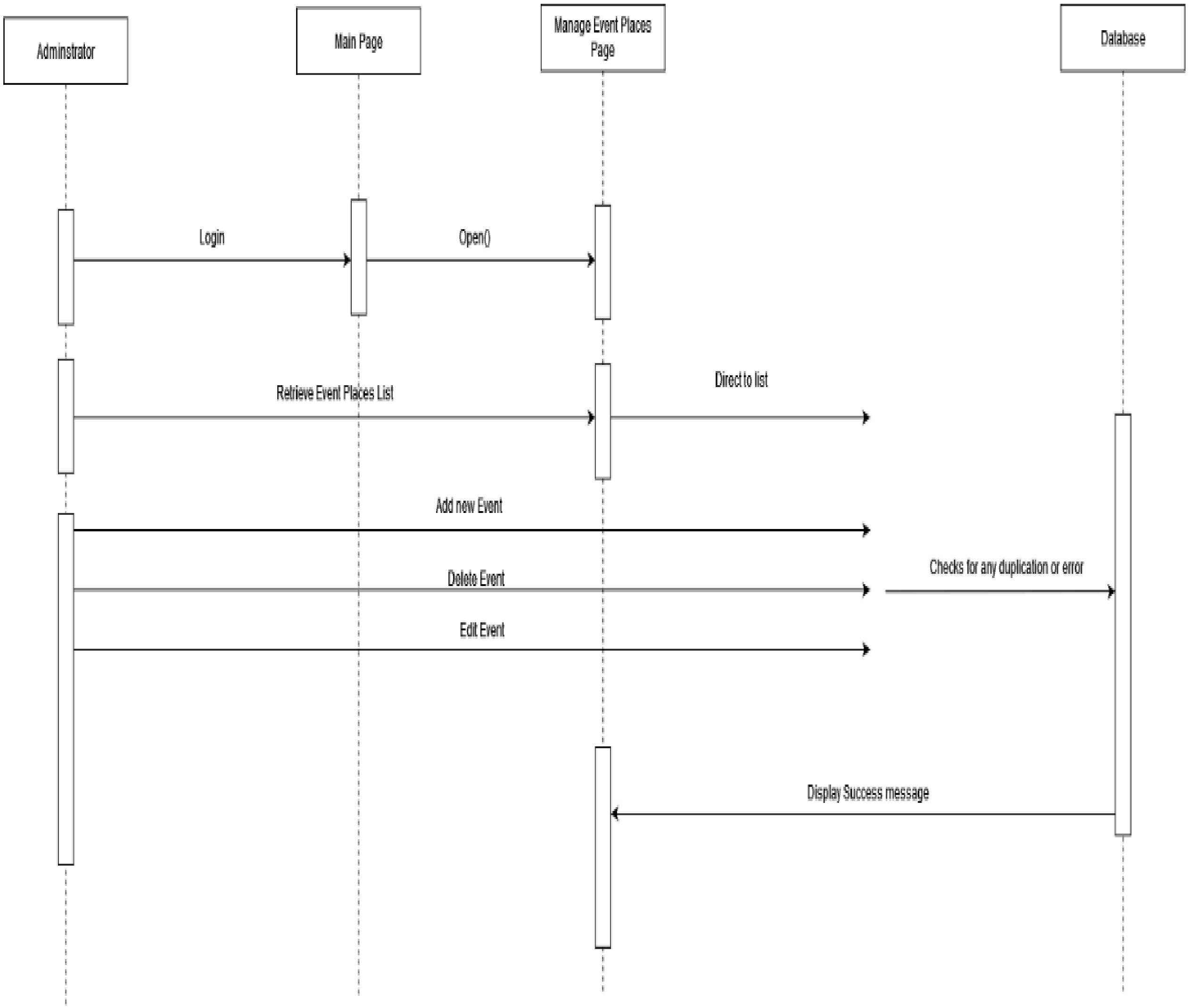


Figure 2.2.12.1 Shows the sequence diagram for AC005 Manage Event Places

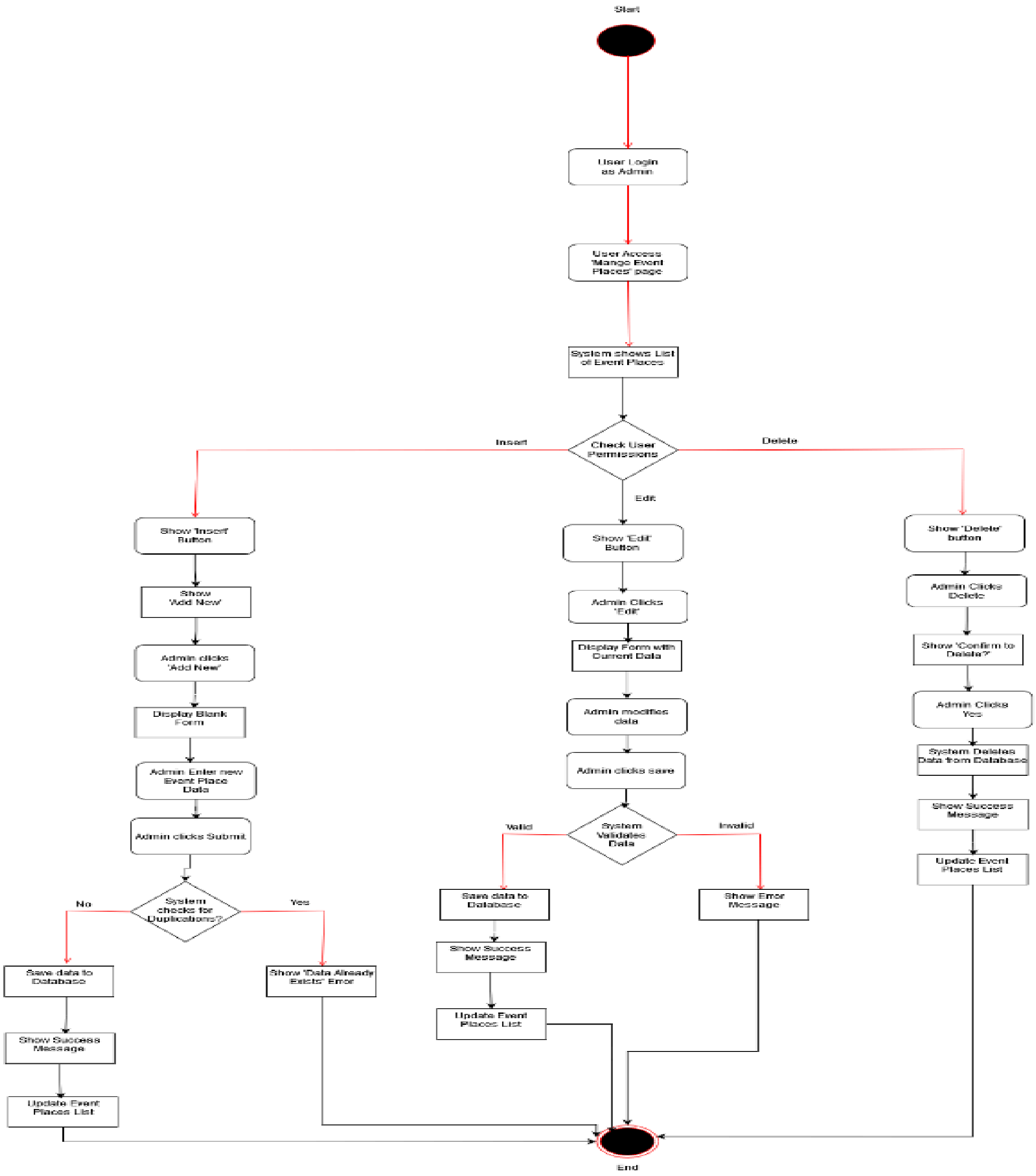
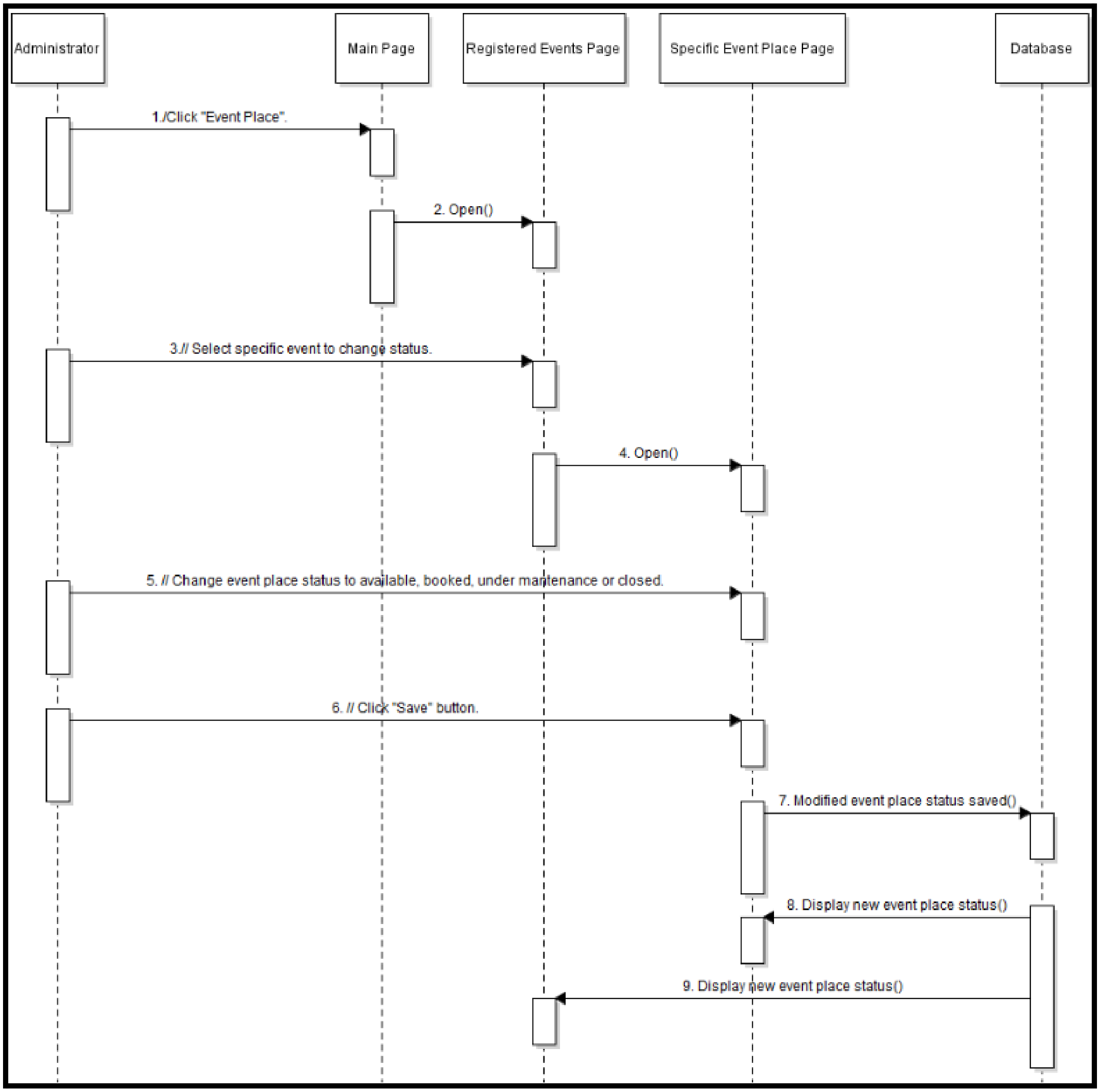


Figure 2.2.12.2 Shows the activity diagram for AC005 Manage Event Places

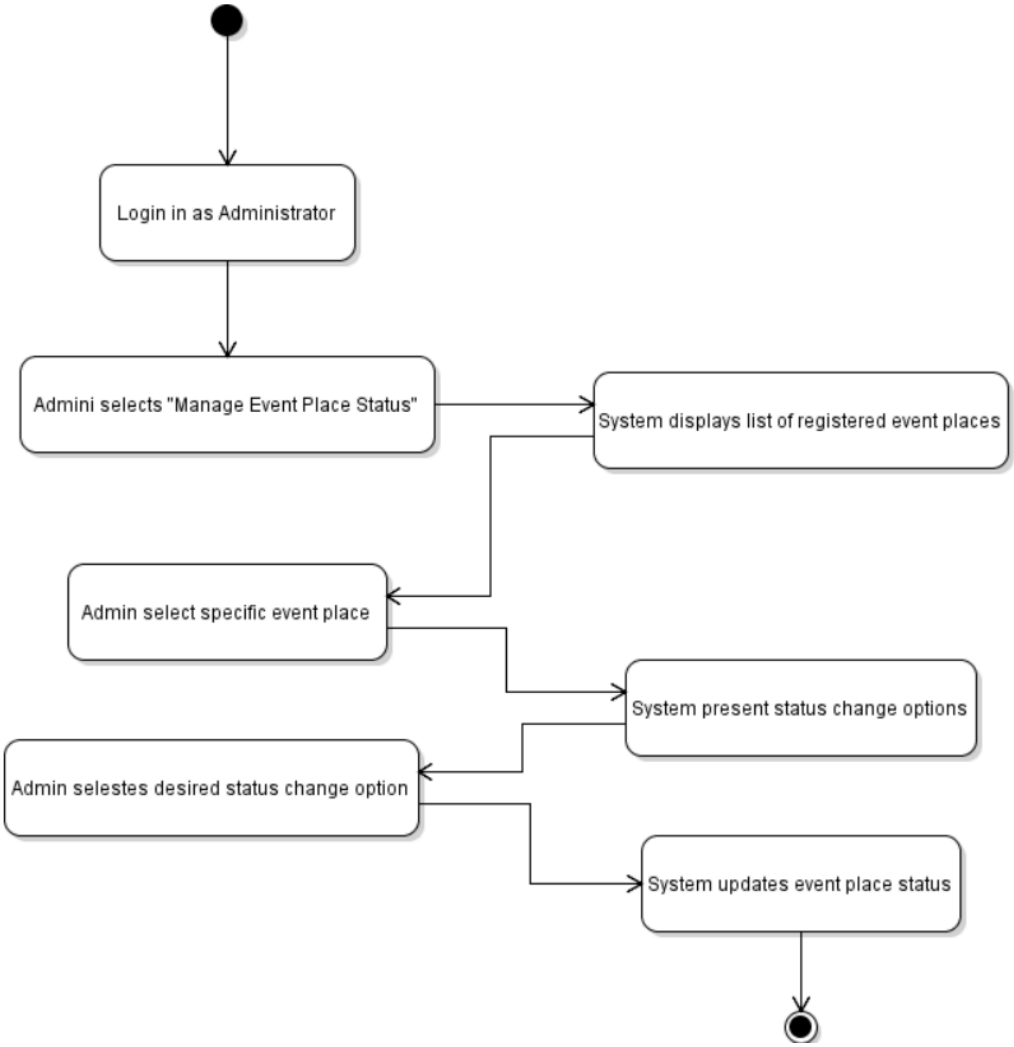
2.2.13 AC006: Use Case Change Event Place Status

Use Case ID:	AC006
Use Case Name:	Change Event Place Status
Actor:	KTR Administrator / Admin
Brief Description:	This use case describes the process of changing the status of an event place within the Kolej Tun Razak Event Management System. The status change could include marking an event place as available, booked, under maintenance, or closed.
Pre-conditions:	1. The administrator must be logged into the event place management system. 2. The event place must be registered or exist in the system.
Normal Flow of events:	1. The administrator selects the "Manage Event Places Status" option from the system main page. 2. The system displays a list of registered event places. 3. The administrator selects the specific event place for which they want to change the status. 4. The system presents options to change the status of the selected event place. 5. The administrator selects the desired status change option; available, booked, under maintenance or closed. 6. The system updates the status of the event place accordingly. 7. The system confirms the successful status change and displays the updated event place details.
Post-conditions:	1. The status of the selected event place is successfully changed in the system. 2. The updated event place status is reflected in the system interface.
Exception flow (if any);	1. If there is a system error or the status change cannot be completed due to technical issues, the system displays an error message. 2. The administrator is prompted to try again later or contact technical support for assistance. 3. The event place status remains unchanged until the issue

	is resolved.


Administrator
Main Page
Registered Events Page
Specific Event Place Page
Database
1./Click "Event Place".
2. Open()
3.// Select specific event to change status.
4. Open()
5. // Change event place status to available, booked, under maintenance or closed.
6. // Click "Save" button.
7. Modified event place status saved()
8. Display new event place status()
9. Display new event place status()

Sequence Diagram for Change Event Place Status

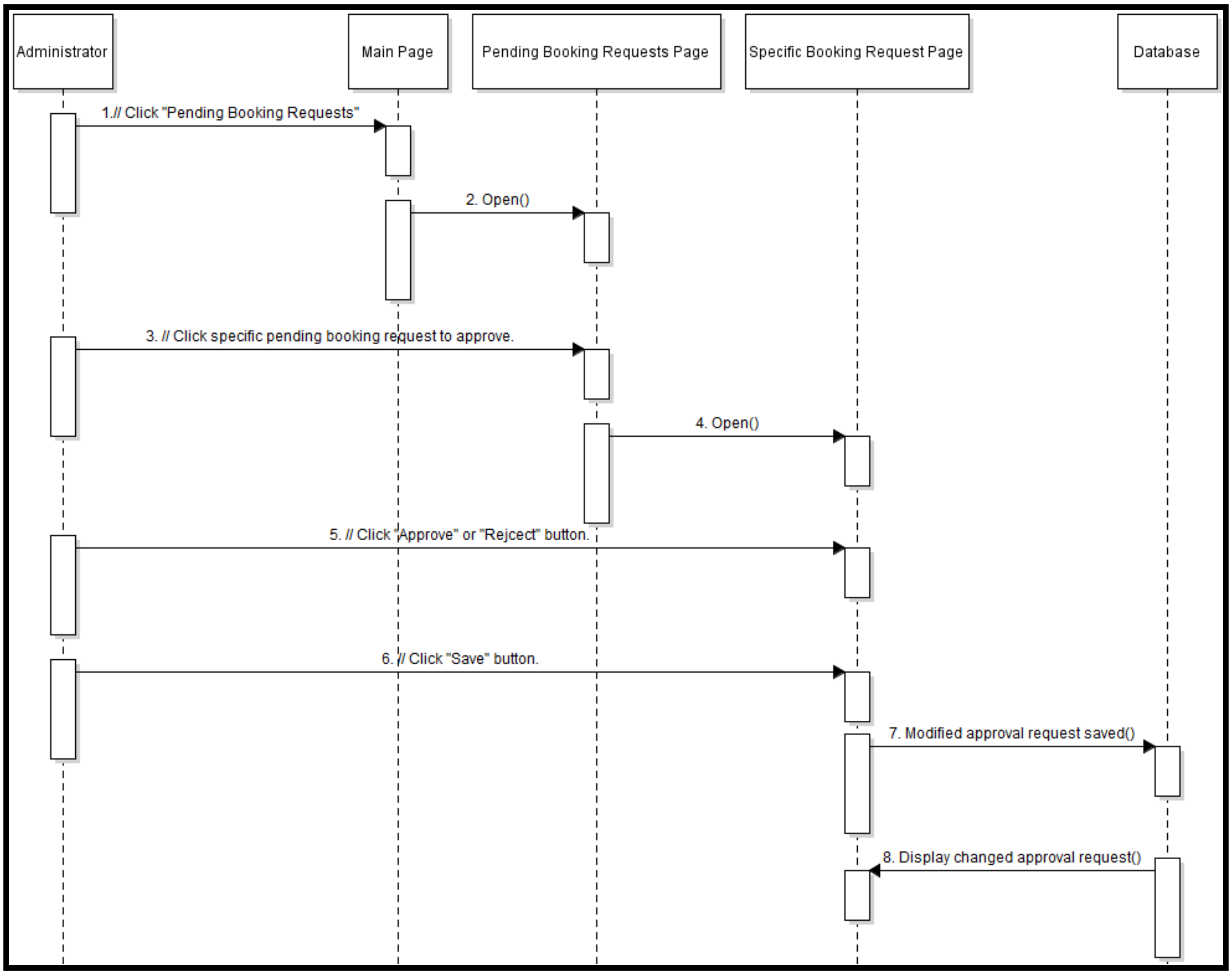

Login in as Administrator
Admini selects "Manage Event Place Status"
System displays list of registered event places
Admin select specific event place
System present status change options
Admin selestes desired status change option
System updates event place status

Activity Diagram for Change Event Place Status

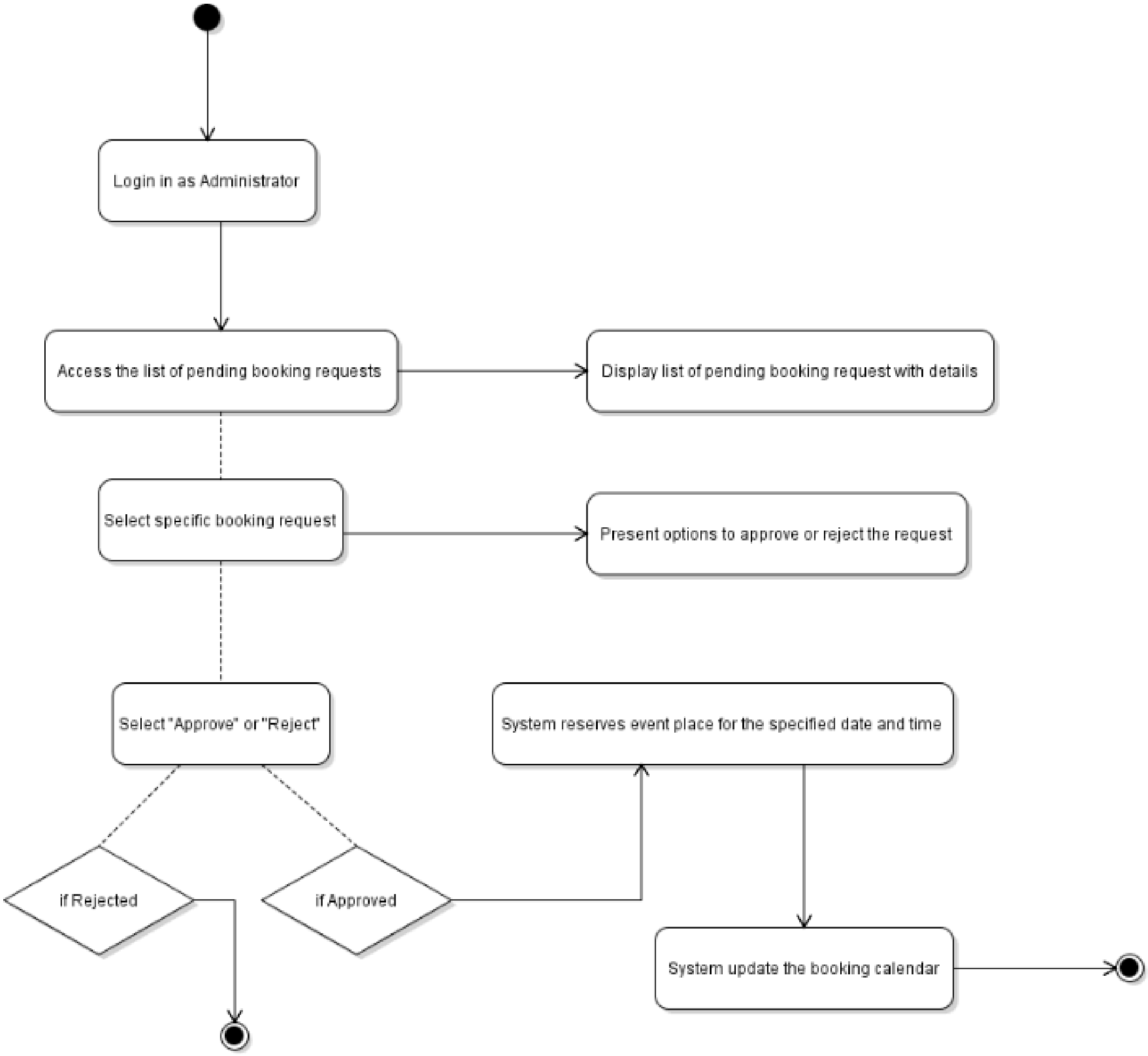
2.2.14 AC007: Use Case Approve Request

Use Case ID:	AC007
Use Case Name:	Approve Request
Actor:	KTR Administrator / Admin
Brief Description:	This use case outlines the process of the administrator approving a booking request for an event place within the Kolej Tun Razak Event Management System. The approval allows the event place to be officially reserved for the specified date and time.
Pre-conditions:	1. The administrator must be logged into the event place management system. 2. A booking request must have been submitted by a user and is pending approval.
Normal Flow of events:	1. The administrator accesses the list of pending booking requests from the system dashboard. 2. The system displays a list of pending booking requests, including details such as event place, date, time, and other information. 3. The administrator selects the specific booking request they wish to approve. 4. The system presents options to approve or reject the selected booking request. 5. The administrator selects the "Approve" or “Reject” option. 6. The system updates the status of the booking request to "Approved" or “Reject” 7. The system reserves the event place for the specified date and time. 8. The system updates the booking calendar to reflect the approval booking.
Post-conditions:	1. The booking request is successfully approved, and the event place is reserved for the specified date and time. 2. The booking calendar is updated to reflect the approved booking.
Exception flow (if any);	1. If there is a system error or the approval process cannot be completed due to technical issues, the system displays an error message. 2. The administrator is prompted to try again later or contact technical support for assistance.

	3. The booking request status remains unchanged until the issue is resolved.


Administrator
Main Page
Pending Booking Requests Page
Specific Booking Request Page
Database
1.// Click "Pending Booking Requests"
2. Open()
3. // Click specific pending booking request to approve.
4. Open()
5. // Click "Approve" or "Rejcect" button.
6. // Click "Save" button.
7. Modified approval request saved()
8. Display changed approval request()

Sequence Diagram for Approve Request

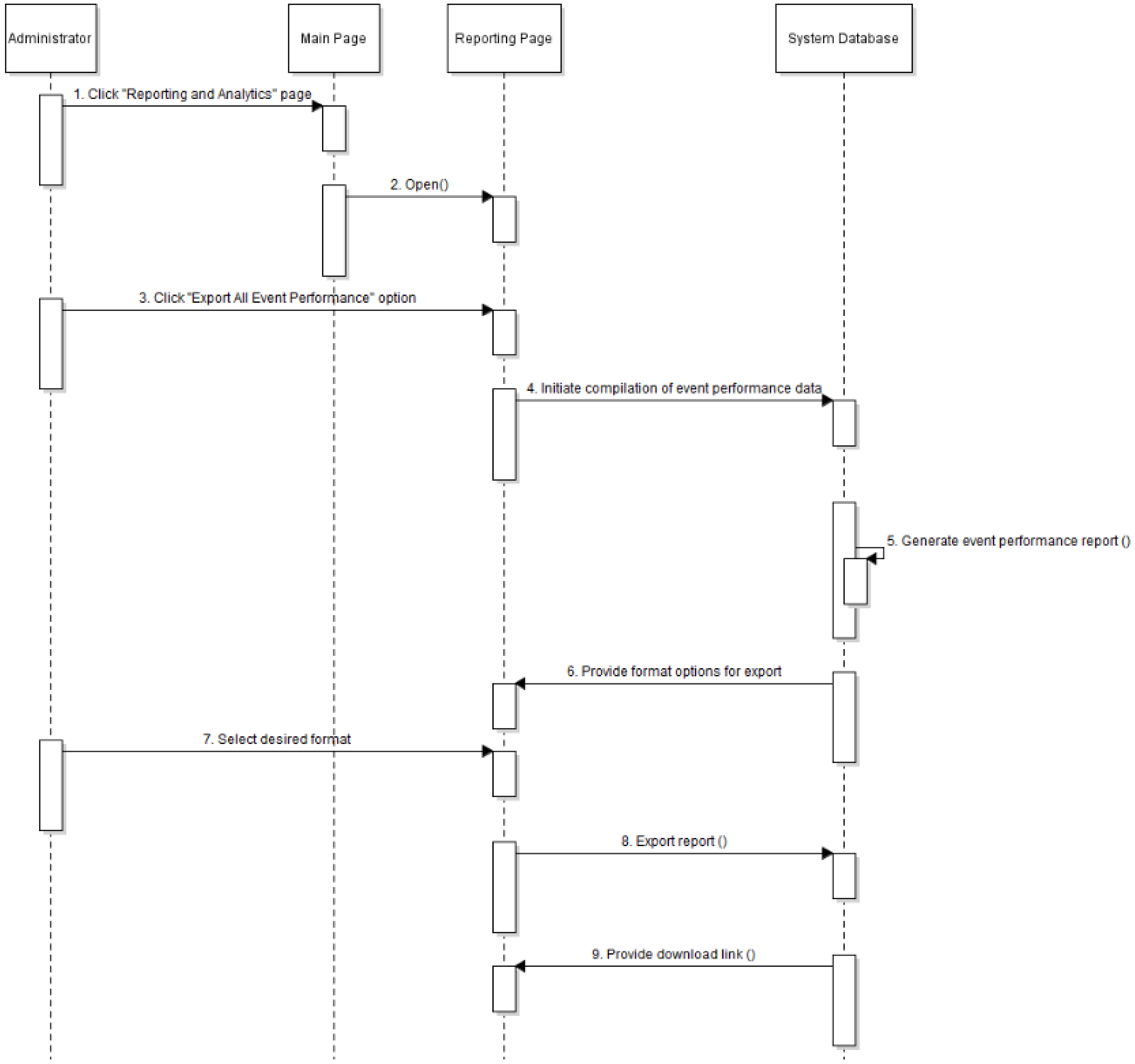

Login in as Administrator
Access the list of pending booking requests
Display list of pending booking request with details
Select specific booking request
Present options to approve or reject the request
Select "Approve" or "Reject"
System reserves event place for the specified date and time
if Rejected
if Approved
System update the booking calendar

Activity Diagram for Approve Request

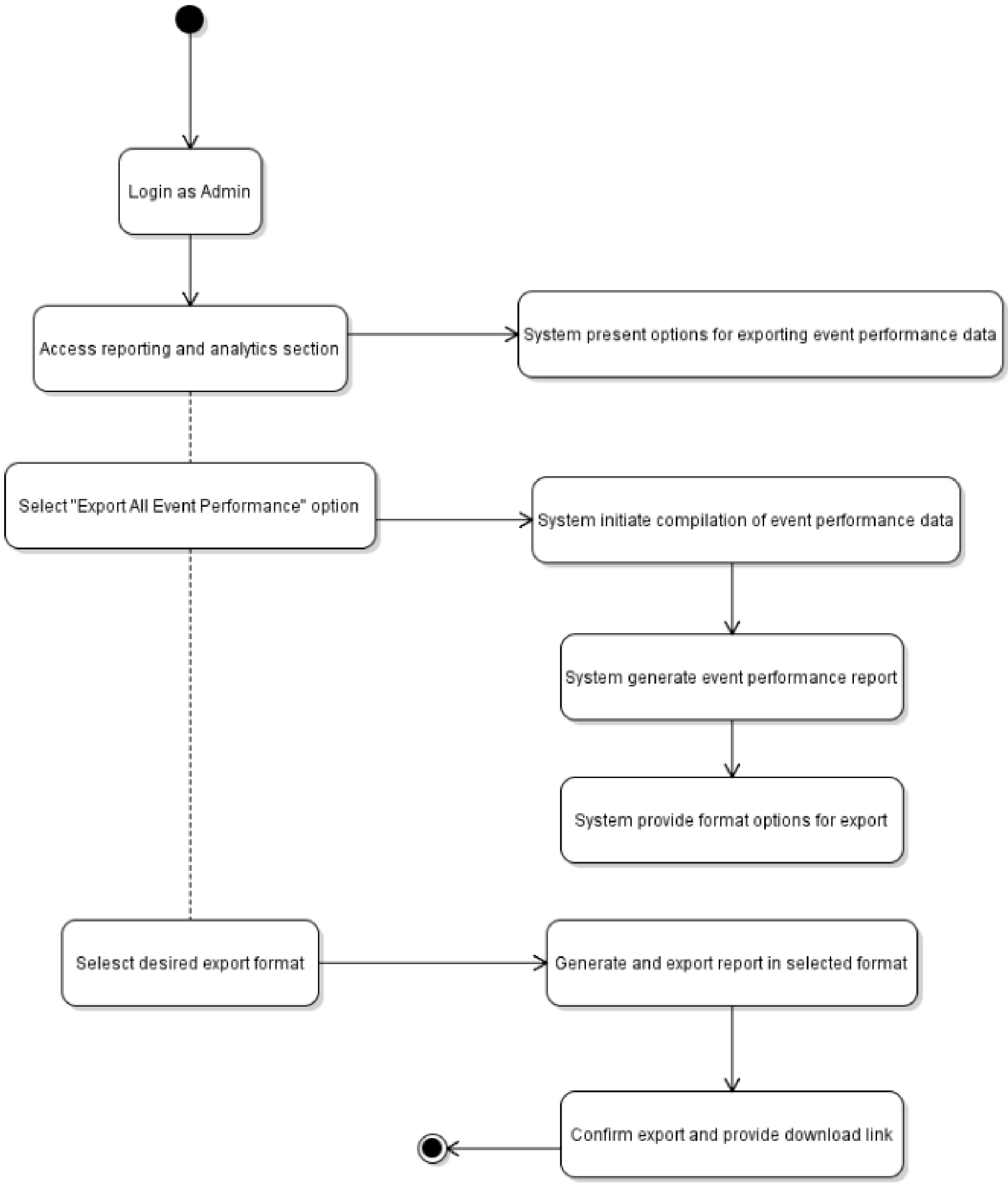
2.2.15 AC008: Use Case Export All Event Performance

Use Case ID:	AC008
Use Case Name:	Export All Event Performance
Actor:	KTR Administrator / Admin
Brief Description:	This use case describes the process of exporting all event performance data from the Kolej Tun Razak event management system. The exported data provides insights into the performance of various events held at the event places managed by the system.
Pre-conditions:	1. The administrator must be logged into the event place management system. 2. The system must contain event performance data for the events held at the managed event places.
Normal Flow of events:	1. The administrator accesses the reporting or analytics section of the event place management system. 2. The system presents options for generating and exporting event performance data. 3. The administrator selects the "Export All Event Performance" option. 4. The system initiates the process of compiling all event performance data. 5. The system generates a report containing comprehensive event performance. 6. The system provides options for the format of the exported file like CSV or Excel. 7. The administrator selects the desired format for the exported file. 8. The system generates and exports the event performance report in the selected format. 9. The system confirms the successful export of the report and provides a download link.
Post-conditions:	1. The event performance data is successfully exported into a file format specified by the administrator. 2. The administrator receives the exported report containing comprehensive event information.
Exception flow (if any);	1. If there is a system error or the export process cannot be completed due to technical issues, the system displays an error message.

	2. The administrator is prompted to try again later or contact technical support for assistance. 3. The event performance data remains accessible within the system for future export attempts.


Administrator
Main Page
Reporting Page
System Database
1. Click "Reporting and Analytics" page
2. Open()
3. Click "Export All Event Performance" option
4. Initiate compilation of event performance data
5. Generate event performance report ()
6. Provide format options for export
7. Select desired format
8. Export report ()
9. Provide download link ()

Sequence Diagram for Export All Event Performance


Login as Admin
Access reporting and analytics section
System present options for exporting event performance data
Select "Export All Event Performance" option
System initiate compilation of event performance data
System generate event performance report
System provide format options for export
Selesct desired export format
Generate and export report in selected format
Confirm export and provide download link

Activity diagram for Export All Event Performance

2.3 Software System Attributes, Performance And Other Requirements

The Event Management System for Kolej Tun Razak needs to follow specific software system characteristics in order to guarantee its efficiency and user-friendliness.

- User-friendliness: The system should be intuitive and easy to navigate for UTM students and KTR administrators alike. In order to improve user experience, it is important to incorporate clear interfaces, logical workflows, and informative feedback messages.

- Consistency : is crucial for the system to operate reliably, without any errors or interruptions occurring. It needs to be strong enough to manage large amounts of user interactions and data processing without experiencing system crashes.

- Maintainability: It is crucial for the system that it can be improved by creating modular and well-documented code for easier maintenance and future updates. Modifications to the system need to be made quickly and efficiently in order to minimize prolonged periods of downtime.

- Portability : Device Portability is very important for accessibility to users, so the system should work with different devices and web browsers. It is important for it to adjust effortlessly to various screen sizes and resolutions, ensuring a uniform user experience on all platforms.

- Compatibility : Harmonization with other pertinent systems or applications utilized at Kolej Tun Razak is necessary. To promote data exchange and interoperability, it is important to ensure that systems are compatible with current databases, authentication methods, and communication protocols.

The Event Management System needs to meet certain performance requirements:

- Response Time : System should react quickly to user inquiries, offering instant feedback and updates. Optimizing response times for critical actions like booking requests and administrative tasks is crucial for improving user satisfaction.

- Throughput: The system needs to efficiently manage numerous simultaneous user interactions, maintaining performance. It needs to ensure that it sustains high levels of throughput, even at busiest times, to avoid any delays or congestion.

- Capacity: The system must have the ability to expand in order to meet the needs of a growing number of users and higher amounts of event-related data. It needs to efficiently manage numerous user registrations, event bookings, and administrative tasks without running out of resources.

- Availability: The system must have high availability, with minimal downtime or disruptions in service. To maintain uninterrupted operation and accessibility for users, it is essential to have redundant servers, load balancing mechanisms, and proactive monitoring in place.

Other Requirements :

Additional non-functional requirements for the system comprise of:

- Security measures must be strict to safeguard user data, sensitive information, and system resources. To avoid unauthorized access or data breaches, it is important to put in place secure authentication mechanisms, data encryption protocols, and access control policies.

- Safety is important in the digital system to protect users and physical facilities. Precautions to avoid accidents, emergencies, or interruptions during events need to be integrated into the system's design.

- Legal and Regulatory: The system must adhere to applicable laws, regulations, and industry standards regarding data privacy, event management, and online transactions. Adhering to legal obligations maintains the system's credibility and reduces the chance of facing legal consequences or fines.

- Environmental: Even if the system's effect on the environment is not direct, it is important to consider factors like energy efficiency, resource usage, and carbon emissions in both the development and operation of the system. Reducing environmental impact is in line with sustainability goals and corporate social responsibility initiatives.

2.4 Design Constraints

Environmental: The system should be able to function well in different environmental conditions on college grounds, such as changes in temperature and humidity.

Hardware Constraints: The system needs to work with a basic hardware setup, guaranteeing good performance on devices with requirements like at least 4GB RAM and dual-core processors.

Security Constraints: The system needs to incorporate strong security measures to safeguard user data and system integrity, such as encrypting sensitive information, using multi-factor authentication, and conducting routine security audits.

Compatibility Constraints: The system should work with current web browsers like Google Chrome, Mozilla Firefox, and Safari to maintain a uniform user experience on various platforms.

Performance Constraints: The system needs to be enhanced to support up to 500 simultaneous users with a response time of under 3 seconds for important actions like booking events and approving requests.