Vietnam National University Ho Chi Minh City

Ho Chi Minh City of Technology

Computer Science and Engineering Faculty

SOFTWARE ENGINEERING

ASSIGNMENT 4

Professor: Bui Hoai Thang

Teaching Assistant: Bui Cong Tuan

Student: Tran Thi Ngoc Diep – 1827005 – Class B01

Group 4

11/2019

Contents

PART 1: ASSIGNMENT 1

1. Introduction (Group work)

1.1. Purposes

The purpose of this report is specifying a Smart Campus System (SCAMS) in detail including system design, features, methods and operation of the system and how the system interacts with variety range of users such as lecturers, students, security staffs,guesses. The system is expected to make the university smart.

1.2. Definitions

SCAMS: Smart Campus System

1.3. Overview

- Section 1: Group work - Introduction to the SCAMS
- Section 2: Group work - Overall description of the SCAMS
- Section 3: Group work and individual work - Functional requirements
- Section 4: Group work and individual work - Non-Functional requirements
- Section 5: Individual - Use-case detail/scenario for use-case the student is in-charged
- Section 6: Individual - Non-interactive functional requirements
- Section 7: Individual – Conclusion
- Section 8: Group list

2. Overall Description (Group work)

2.1. System Use Case Diagram

The SCAMS system includes/connects to the following components:

- **Room Information and Schedule**: connects Room Management Service

(ROMS) of the university to fetch room data, book and modify room for different purposes.

- **IoT system**: connects to the IoT system of the university to view information, control electrical devices.
- **User Database System**: connects to the User Database System of the university to fetch user information and verify user.
- **Infrastructure Database**: connects to the Infrastructure Database of the university to fetch infrastructure and facility information .
- **AI service**: connects to the external AI service for detecting human.
- **Emailing System**: connects to the external email service to send email automatically to users and other stakeholders.
- **End User Interface System**: includes a system of interfaces for end user on different platforms.
- **Statistics and Reports System**: includes a systems of statistics and reports for different purposes.

End-users of SCAMS system:

- *Lecturers*
- *Security Staffs*
- *Office Staffs*
- *Students*
- *Guests*
- *Board of Rectors*
- *System Administrators*

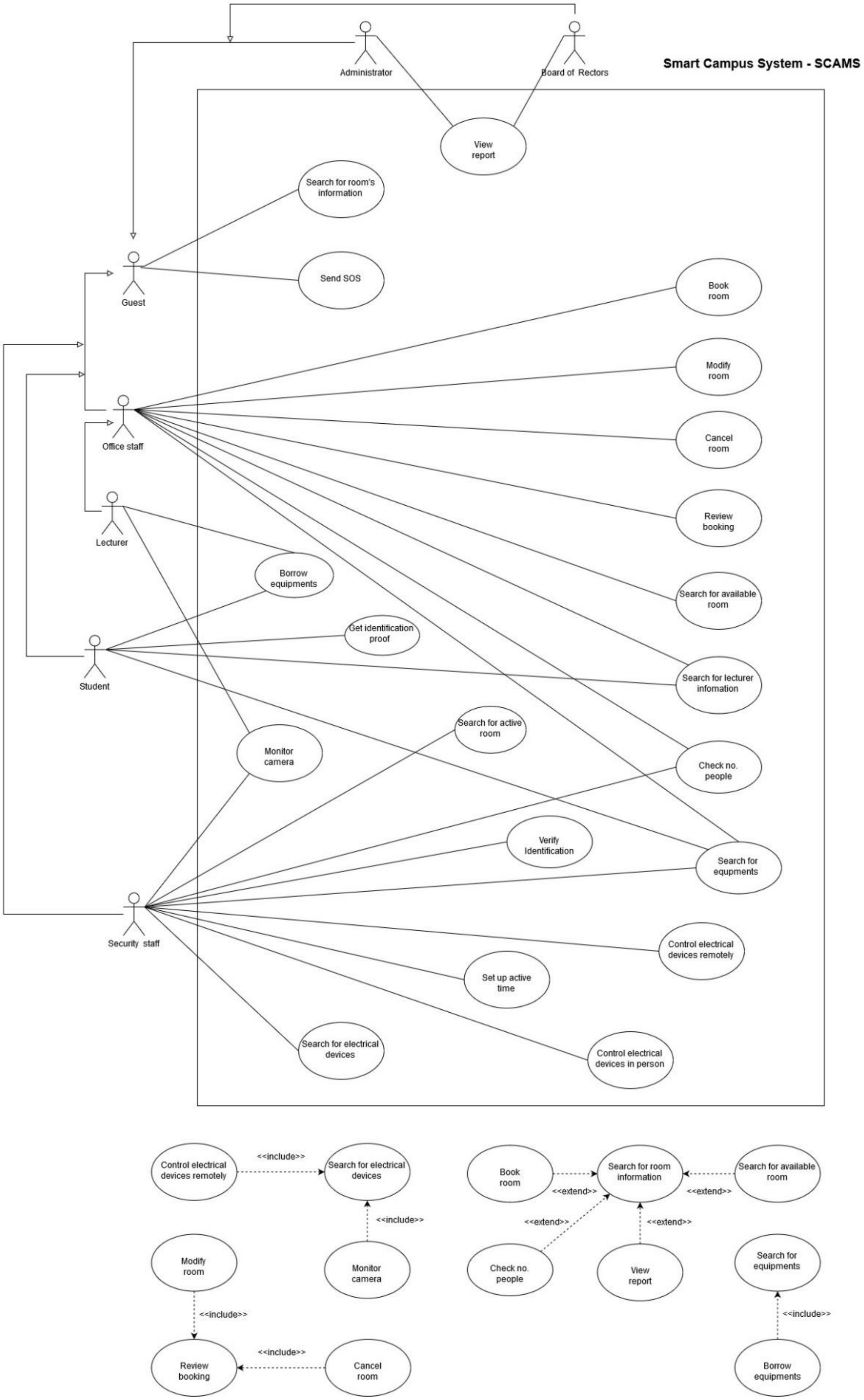


Figure 1.1: System Use-case Diagram For Smart Campus System (SCAMS)

2.2. System Context Diagram

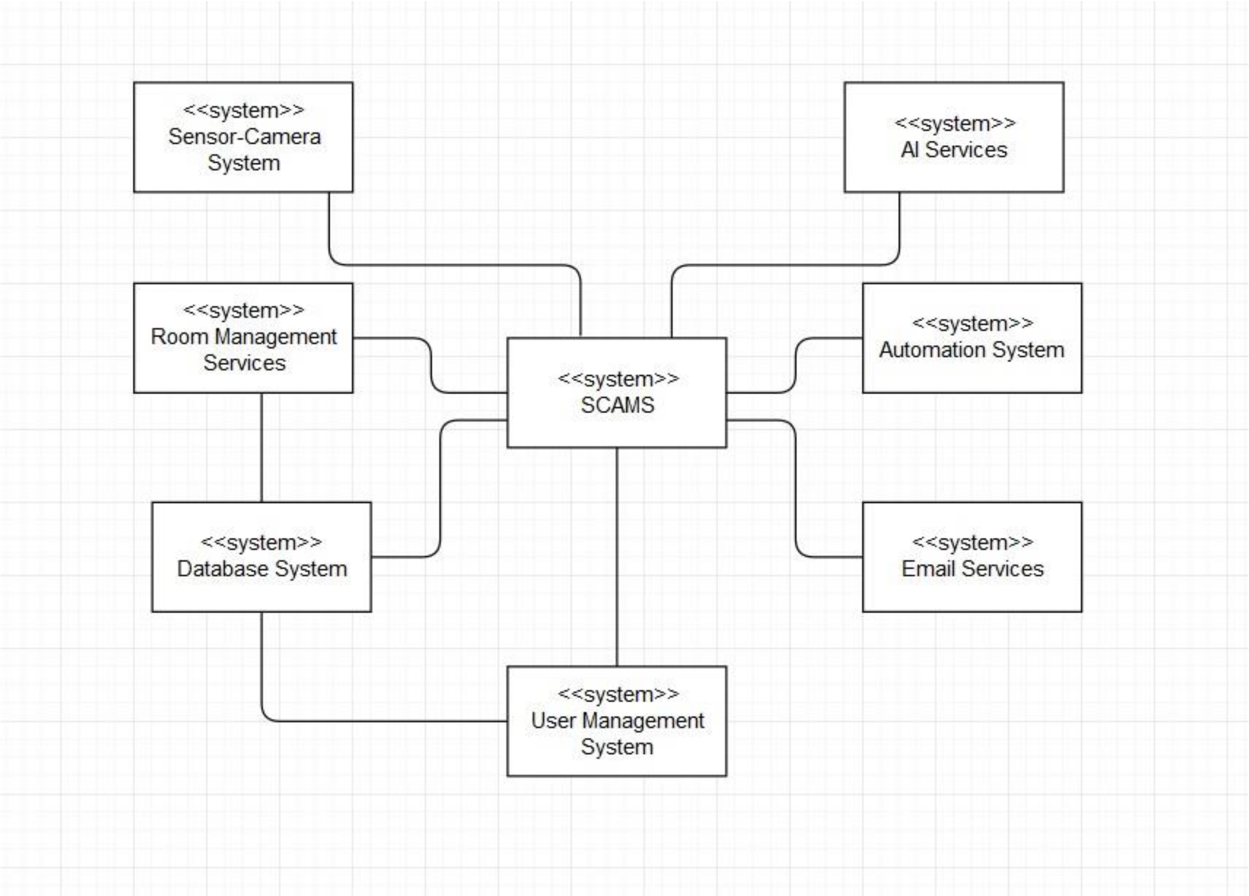


Figure 1.2: Context-model Diagram

3. Functional Requirement Specifications (Group work)

3.1. Room Information and Schedule

3.1.1. Search room for information

All users can search for room information (available or not, capacity, current lecture/event information, direction to the room) using date, building, room number.

3.1.2. Search room for booking

Lecturers and office staff can search room to see its information (available or not, capacity) for booking purpose, using search criteria: date, building, room number.

3.1.3. Book room

Lecturers and office staff can book a room in one or more class periods for different purposes (teaching, event).

3.1.4. Cancel room

Lecturers and office staff can cancel a room they booked if they change their schedule.

3.1.5. Modify room

Lecturers and office staff can modify a room they booked if they change their schedule.

3.1.6. Review booking

Lecturers and office staff can review all of their booked rooms on the application

3.1.7. Confirm by email

Email will be sent to the lecturer/office staff email address when he/she finished booking/modify/canceling room.

3.1.8. Remind by email

User (lecturer & office staff) will receive a reminder of his/her booked room's information & schedule before the lecture.

3.1.9. Campus map

User can open the whole campus map on the application.

3.1.10. Recommend room

The system recommend lecturer the suitable room for them.

3.1.11. Search Lecturer

Lecturers, Students, Office Staff and Security Staff can search for lecturers information.

3.1.12. Borrow Babratory

User can register for borrowing laboratory equipment.

3.2. Electrical Automation System

3.2.1. Turn on automatically

The doors and its electrical (lights, fans) devices will be turned on automatically before the lecture if they are off.

3.2.2. Turn off automatically

Electrical devices will be turned off and doors will be closed after the end of lecture & after the last person left the room if there are no next lecture in the same room.

3.2.3. Alert not functional

An alert will be sent to the security staff if doors the automation system is function properly (not on or of as it has to be).

3.2.4. Alarm automatically

The sound will be alarmed at if there is still some person in the room after the closed hour of the university.

3.3. Security Staff

3.3.1. Search for electrical devices

Security staff can search for electrical devices by room's ID, building's ID.

3.3.2. Search for active room

Security can search for rooms in the campus which is still in active after the closed hour.

3.3.3. Control electrical devices remotely

Security staff can control electrical devices using the application.

3.3.4. Control electrical devices in-person

Security staff use code on the system to access control panel attached on each room, corridor, building to control in-person.

3.3.5. Monitor camera

User (security staff) can monitor rooms, corridor and building using camera.

3.3.6. Enter the campus

User can use the changeable QR code provided by the system to verify identification in order to enter the campus instead of using the ID card

3.4. Sensor and Camera System

3.4.1. Check number of people

User can check for the number of people in the classroom currently.

3.4.2. Automatically control the light

The lights in corridors can be turned on/off automatically based on human detecting system.

3.4.3. Alert detecting human

The system can send an alert to the security staff in case of detecting human.

3.4.4. Detect fire

The system is able to detect fire based on a system of sensor detecting smoke and fire.

3.5. SOS Alert

User can send SOS alert to every user in the campus if he/she is in danger.

3.6. Statistical and Reporting System

3.6.1. View electrical usage

System Administrators and Board of Rectors can view reports on the usage of electrical devices.

3.6.2. View room usage

System Administrators and Board of Rectors can view reports on the usage of electrical devices in a period of time.

3.6.3. Automatically send report

A monthly report will be generated and sent automatically to the System Administrators and the Board of Rectors via email.

3.6.4. User feedback

Any user can provide information and suggestions about the use of electrical devices to the System Administrators using the reporting box.

4. Non-Functional Requirements Specification (Group and Individual work)

4.1. Group work - Non-Functional Requirements Specification

4.1.1. Efficiency requirements

- Response Time:
  - The maximum start up time of the system is 2 seconds
  - The maximum react time of the system for a particular job is 2 seconds.
  - The maximum receiving search result time is 4 seconds.
  - The doors and its electrical (lights, fans) devices will be turned on automatically 15 minutes before the lecture if they are off.
  - Electrical devices will be turned off and doors will be closed immediately after the end of lecture & after the last person left the room if there are no next lecture in the same room.
  - Report for System Administrator and Board of Rectors will be generated before 5pm on the last working day of the month.
  - The maximum interaction for each activity (searching, booking, cancelling, modifying, monitoring, SOS messaging, report retrieving) is 2.
- Space:
  - At least 20 percent of the processor capacity and storage space available to the system shall be unused at normal load seasonal periods.
- Performance:

- At least 80 percent of the request must be successful in 1 minute.

4.1.2. Regulatory

- The system shall not affect other system while using shared resources.

4.1.3. Usability

- User Interface:
  - The interface must be available in English and Vietnamese.
  - The theme color of interface shall be blue (Color Hex #1384d4).
- Learnability:
  - The product shall be able to used by adult members (age 18 to 60) without training.
  - The product shall be self-explanatory for whom encounter it the first time.

4.1.4. Reliability

- Readiness:
  - The system shall be at least 99 percent available everyday, maintenance is not included.
  - Unless the system is non-operational, the system shall send a user an email informing them that the system is unavailable.
- Fault tolerance:
  - The system should ensure operation in case of errors.
- Resilience:
  - The system should recover operation in at most 3 hours in case of incidents.

4.1.5. Confidentiality

- Security:

  - The system should provide multi-layer security for accessing management tools.

- Integrity:
  - Username and password of users should be encrypted.
- Authentication:
  - The system should provide two-factor authentication.
  - The system should authenticate by a sequence of 4 numbers (PIN) besides password.

4.1.6. Survivalbility

- Analytical ability:
  - The system should be able to identify issues itself and provide reports.
- Modifiability:
  - In source code, function calls shall not be nested more than two levels deep.
  - A junior developer who has at least 6 months of experience programming should be able to add a new feature, including code modifications and testing.
- Adaptation ability:
  - The system can operate on several platforms: web and mobile.
  - The system should display on different resolution based on devices.
  - The system must be able to interface with any HTML (HyperText Markup Language) browser.
- Installability:
  - The system should be installed easily and automatically.
  - The installation time must be within 5 minutes.

- When a new version of the main system is released, it shall be possible to upgrade to it from any previous version.

4.2. Individual - Other Non-Functional Requirements

4.2.1. Efficiency requirements

- User (lecturer) will receive a reminder of his/her booked room's information & schedule before 5pm the previous day.
- Email will be sent to the user's email address when he/she finished booking/modify/canceling room within 5 seconds.
- The response time of electrical devices for automatically or manually command is less than 2 seconds.
- The system must be able to accommodate up to 400 simultaneous users during the peak load period.
- The system must process a notification (SOS message, alert, fire detection) in maximum 1 second, and up to 100 notifications in no more than 15 seconds.

4.2.2. Regulatory

- The system must guarantee that external systems access such as ROMS shall not be overlap.

4.2.3. Usability

- The theme color of interface shall be blue (Color Hex #1384d4).
- The product shall be self-explanatory for whom encounter it the first time.

5. Use-case in detailed (Individual work)

5.1. Search Room Information

5.1.1. Context Model

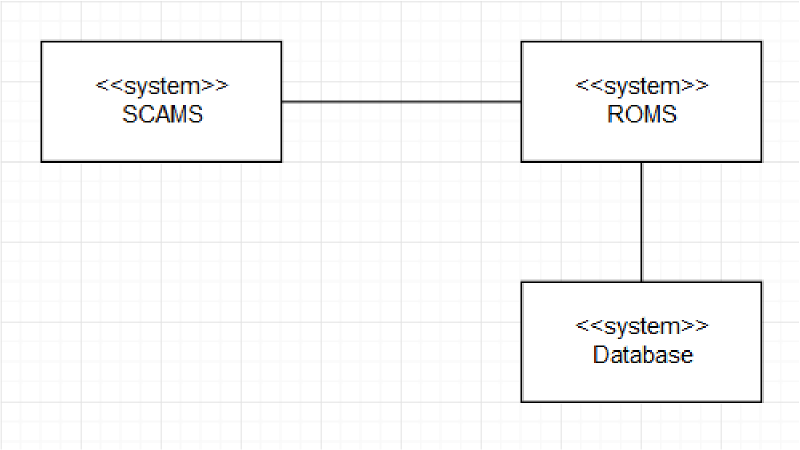


Figure 1.3: Context Model for Search Room Information Use-case

5.1.2. Use-case Diagram

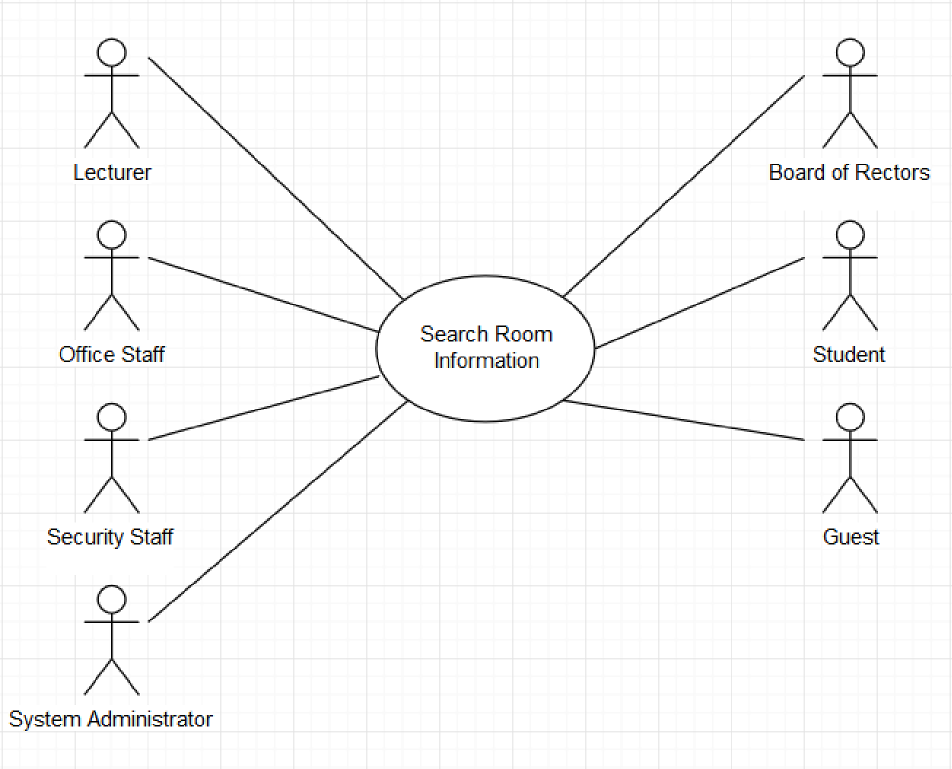


Figure 1.4: Use-case Diagram for Search Room Information

5.1.3. Use-case Scenario Tabular

Use case name	**Search Room Information**
Actor	User (Security Staff, Lecturer, Office Staff, System Administrator, Board of Rectors, Student, Guest)
Description	User can search for room information
Preconditions	User logged in into SCAMS application
Triggers	None
Normal Flow	1. User selects "Search" on SCAMS home page 2. System represents a search panel with multiple filters (date) and (building, room number). Date and building must be filled 3. User selects date, building and room number filters 4. System represents a page with the room's usage timetable and capacity 5. User chooses each lecture/event 6. System represent a pop up box giving the lecture/event's information: lecturer/speaker, subject/event's name 7. User choose to quit the view 8. User choose to quit the search
Exception	*Exception at step 4. Cannot find the room number* 4a. The system represents a pop up box saying that user should use other search criteria (Alternative 3a) *Exception at step 4. System failed to connects with the ROMS system to fetch information.* 4b. A pop-up box will explain the situation: "This service is not available"
Alternative Flows	*Alternative 1: at step 3* 3a. User selects date and building 3b. System represent a list of room in that building and it's usage timetable, capacity 3c. User select on a particular room to see its usage timetable, move to step 4 *Alternative 2: at step 5* 5a. User chooses "Room information" 5b. System represents the room's capacity, facilities and direction to that room

5.1.4. Non-functional Requirements

Category	Requirement	Unit	Maximum value
Efficiency	Receive search result	second	4
	Number of interactions	interactions	4
Usability	User shall learn to use easily	minute	5
	Available 24/7		24/7
Adaptability	The system shall operate on multiple platforms		

5.2. Search Room for Booking

5.2.1. Context Model

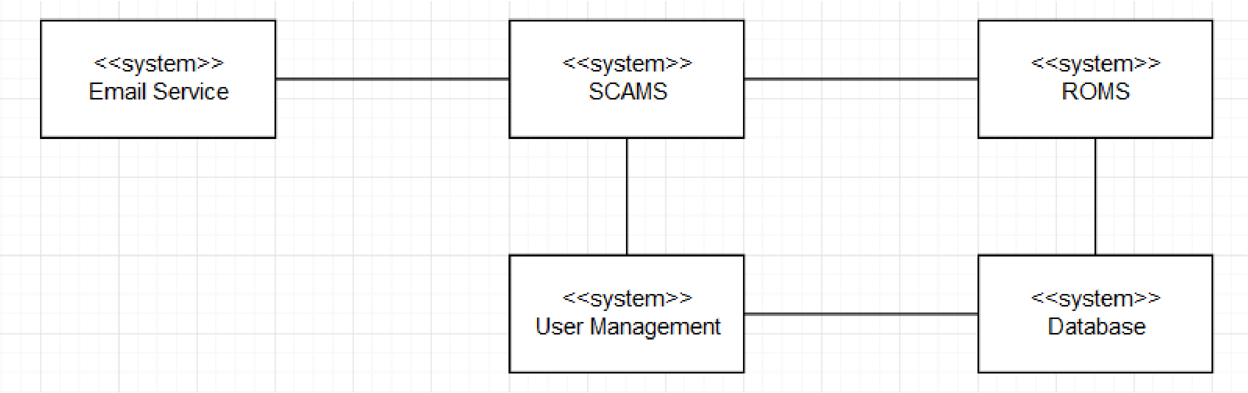


Figure 1.5: Context Model for Search Room for Booking Use-case

5.2.2. Use-case Diagram

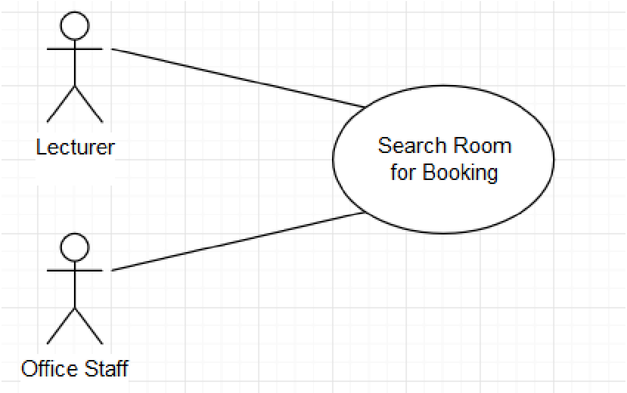


Figure 1.6: Use-case Diagram for Search Room for Booking Use-case

5.2.3. Use-case Scenario Tabular

Use case name	**Search Room for Booking**
Actor	User (Lecturer, Office Staff)
Description	User can search for available room and book
Preconditions	User logged in into SCAMS application
Triggers	None
Normal Flow	1. User selects "Search" on SCAMS home page 2. System represents a search panel with multiple filters (date) and (building, room number). Date and building must be filled 3. User selects date, building and room number filters 4. System represents a page with the room's usage timetable and capacity 5. User chooses "Book" button on the room 6. System represent a booking page where there's a list of available class period in that day of that room 7. User choose the time he/she willing to book Then User fill in the class/event information. - If user is a lecturer, a dropdown list of his/her active classes will be shown and he/she can choose which class it is for this room. He/she can also book room for other purpose, which will be explained in "Other" box. - If user is an office staff, he/she will describe the event in "Event" box and fill in the "Speaker" information. 8. User select "Confirm" 9. System shows a pop-up announces that the room is booked successfully, the user is now back to the search result screen 10. An email is sent immediately to the user's email address confirm that the booking is successful 11. User choose to quit the search
Exception	*Exception at step 4. Cannot find the room number* 4a. The system represents a pop up box saying that user should use other search criteria (Alternative 3a) *Exception at step 4. System failed to connects with the ROMS system to fetch information.* 4b. A pop-up box will explain the situation: "This service is not available" *Exception at step 4. The room full at that time*

	Alternative 2
Alternative Flows	*Alternative 1: at step 3* 3a. User selects date and building 3b. System represent a list of room in that building and their usage timetable, capacity 3c. User select on a particular room to see its usage timetable, move to step 4 *Alternative 2: at step 4* 4a. User quit the search 4b. User search using another date or another room

5.2.4. Non-functional Requirements

Category	Requirement	Unit	Maximum value
Efficiency	Receive search result	second	4
	Number of interactions	interactions	6
	Email shall be sent immediately after the booking is confirmed	second	1
Usability	User shall learn to use easily	minute	10
	Available 24/7		24/7
Adaptability	The system shall operate on multiple platforms		

5.3. Cancel Booking

This use-case is a continuous of use-case "Review booking", where user (lecturer, office staff) can review all of his/her booking. Then if user selects on a particular booking, he/she will see the modify and cancel booking options. **This use-case does not focus on the Review Booking use-case, but only on the cancel room booking process.**

5.3.1. Context Model

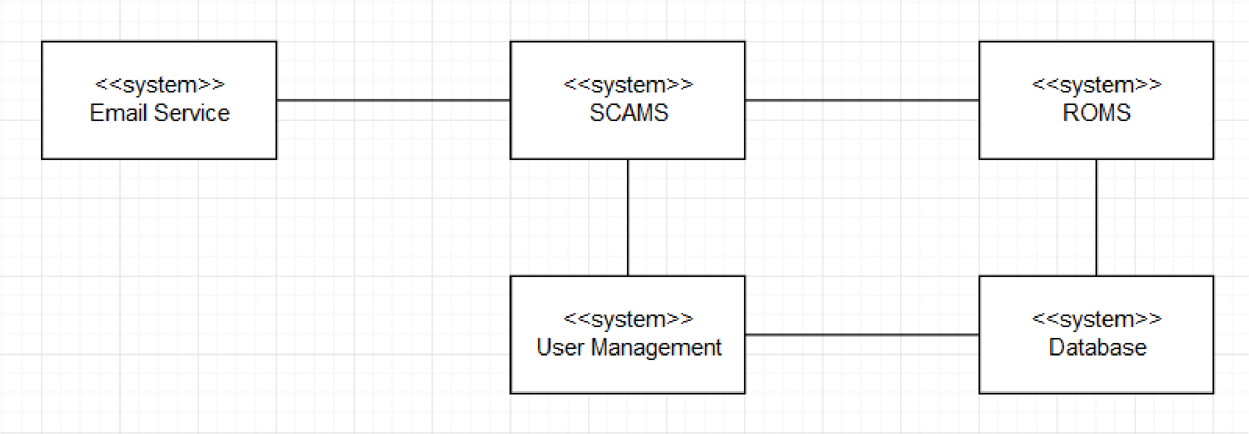


Figure 1.7: Context Model for Cancel Room Use-case

5.3.2. Use-case Diagram

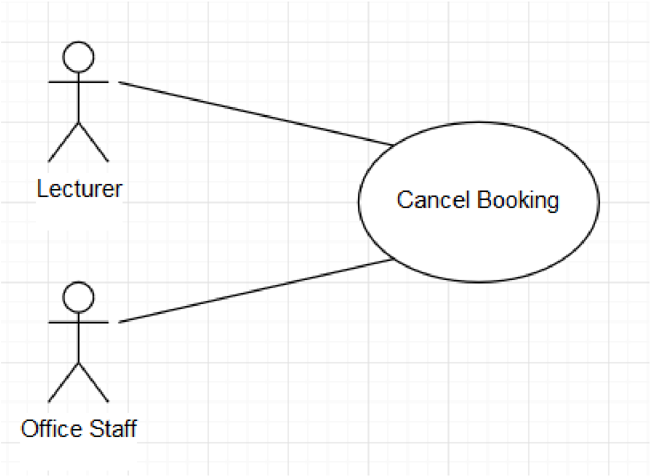


Figure 1.8: Use-case Diagram for Cancel Room Use-case

5.3.3. Use-case Scenario Tabular

Use case name	**Cancel Room**
Actor	User (Lecturer, Office Staff)
Description	User can cancel his/her booked room
Preconditions	User logged in into SCAMS application
Triggers	None
Normal Flow	1. User selects "Manage Booking" on menu from home screen 2. System represents a list of his/her booked room 3. User selects filter (Class/Event or Date) to filter and find the booking he/she is looking for 4. System represents the list of room(s) booking on search result 5. User selects on the room booking want to cancel 6. System represents the booking information and options "Modify" and "Cancel" 7. User chooses "Cancel" 8. System shows a pop-up box announces that the cancel process is done successfully. 9. An email is sent immediately to the user's email address 10. User choose to quit the manage booking section
Exception	*Exception at step 2. System failed to connects with ROMS* 2a. A pop-up box will explain the situation: "This service is not available at the moment"
Alternative Flows	

5.3.4. Non-functional Requirements

Category	**Requirement**	**Unit**	**Maximum value**
Efficiency	Receive search result	second	4
	Number of interactions	interactions	4
	Email shall be sent immediately after the booking is canceled	second	1
Usability	User shall learn to use easily	minute	10
	Available 24/7		24/7

Adaptability	The system shall operate on multiple platforms		

5.4. Modify Booking

This use-case is a continuous of use-case "Review booking", where user (lecturer, office staff) can review all of his/her booking. Then if user selects on a particular booking, he/she will see the modify and cancel booking options. **This use-case does not focus on the Review Booking use-case, but only on the modify room booking process.**

5.4.1. Context Model

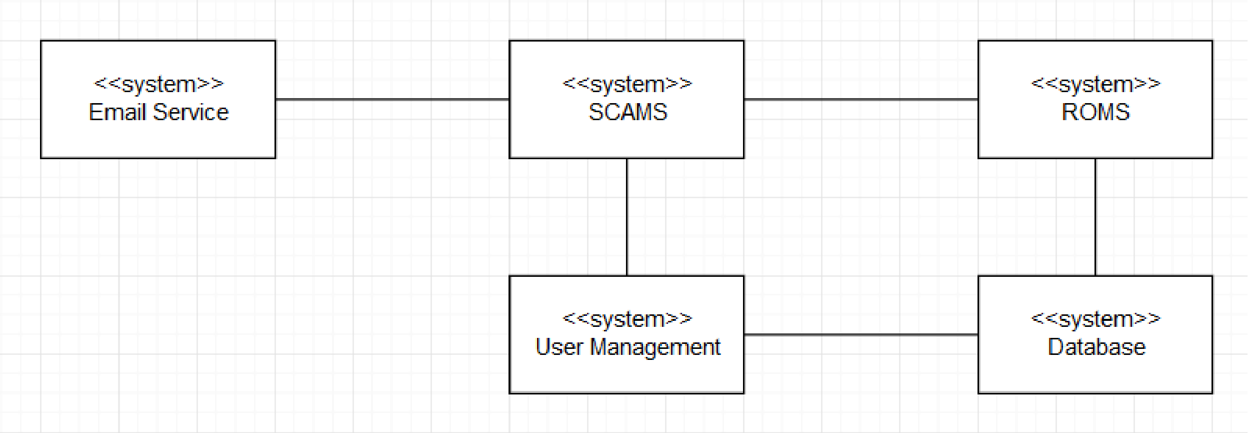


Figure 1.9: Context Model for Modify Room Use-case

5.4.2. Use-case Diagram

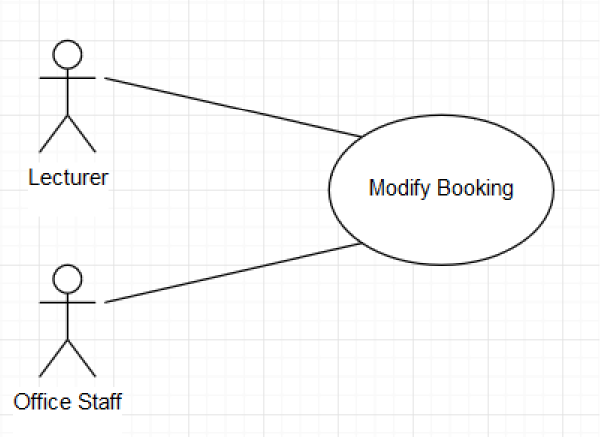


Figure 1.10: Use-case Diagram for Modify Room Use-case

5.4.3. Use-case Scenario Tabular

Use case name	**Modify Room**
Actor	User (Lecturer, Office Staff)
Description	User can modify his/her booked room
Preconditions	User logged in into SCAMS application
Triggers	None
Normal Flow	1. User selects "Manage Booking" on menu from home screen 2. System represents a list of his/her booked room 3. User selects filter (Class/Event or Date) to filter and find the booking he/she is looking for 4. System represents the list of room(s) booking on search result 5. User selects on the room booking want to modify 6. System represents the booking information and options "Modify" and "Cancel" 7. User chooses "Modify" 8. System represents a search panel with multiple filters (date) and (building, room number). Date and building must be filled 9. User selects date, building and room number filters 10. System represents a page with the room's usage timetable and capacity 11. User chooses "Select" button on the room 12. System represent a pop up list of available class period in that day of that room 13. User choose the time he/she willing to book 14. User select "Confirm" 15. System shows a pop-up announces that the booking is modified successfully, the user is now back to the manage booking screen 16. An email is sent immediately to the user's email address confirm that the booking modify process is successful 17. User choose to quit the manage booking section
Exception	*Exception at step 2. System failed to connects with the ROMS system to fetch information.* 2b. A pop-up box will explain the situation: "This service is not available" *Exception at step 12. The room full at that time* Alternative 2

Alternative Flows	*Alternative 1: at step 10* 10a. User selects date and building 10b. System represent a list of room in that building and their usage timetable, capacity 10c. User select on a particular room to see its usage timetable, move to step 12 *Alternative 2: at step 12* 12a. User chooses another date/room and continue the modify process *Alternative 3: at step 12* *12b. User choose to quit the searching new date/room and cancel the modify process, continue in step 17*

5.4.4. Non-functional Requirements

Category	Requirement	Unit	Maximum value
Efficiency	Receive search result	second	4
	Number of interactions	interactions	6
	Email shall be sent immediately after the booking is modified	second	1
Usability	User shall learn to use easily	minute	10
	Available 24/7		24/7
Adaptability	The system shall operate on multiple platforms		

6. Other non-interactive functional requirement (individual)

- An email will be sent to all student of that class the lecturer book/modify/cancel room automatically
- A message will be automatically sent to System Administrator if there is any error in connecting with other systems

PART 2: ASSIGNMENT 2

1. Sequence Diagram

1.1. Search Room Information

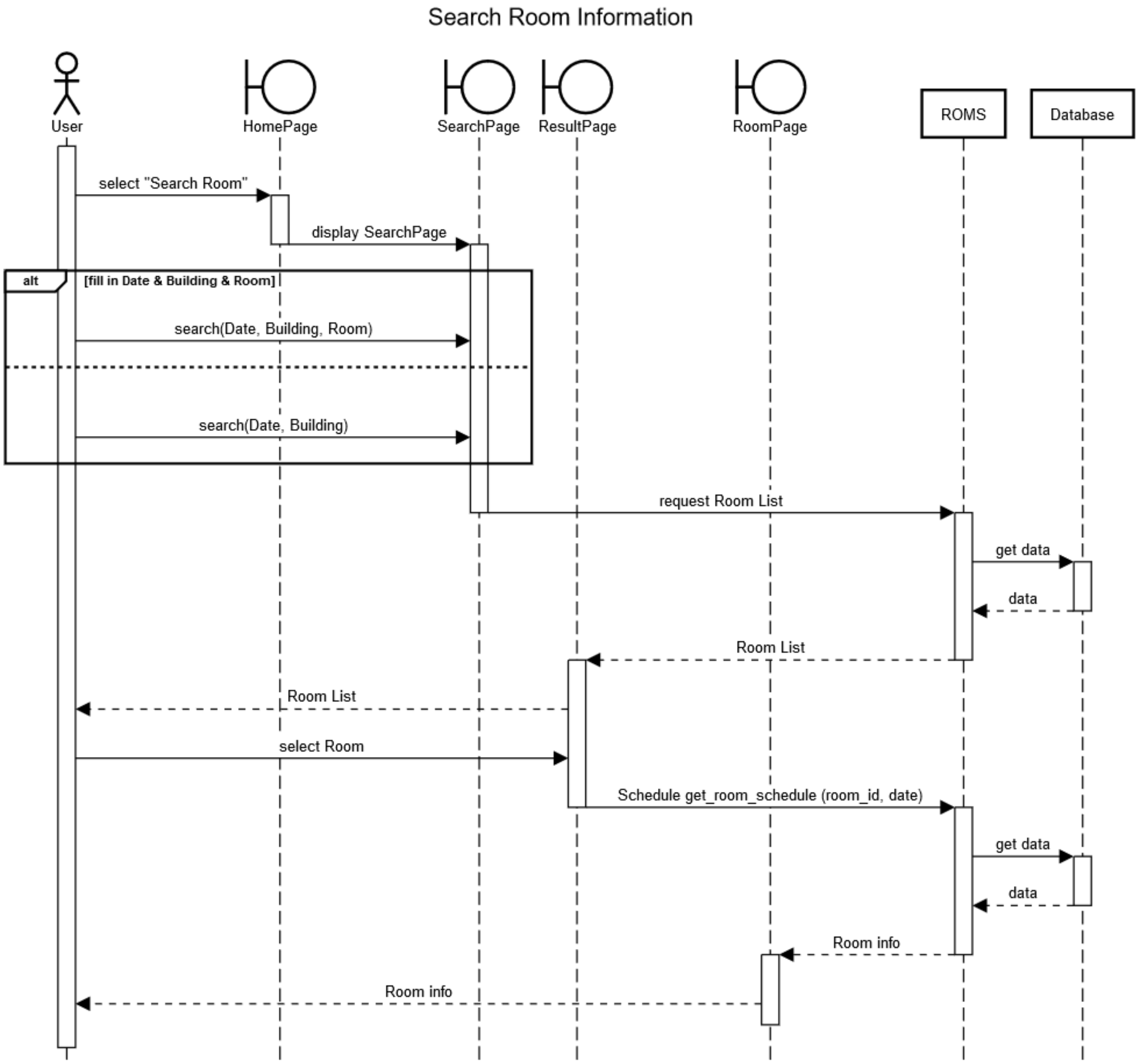


Figure 2.1: Sequence Diagram for Search Room Information

1.2. Search Room for Booking

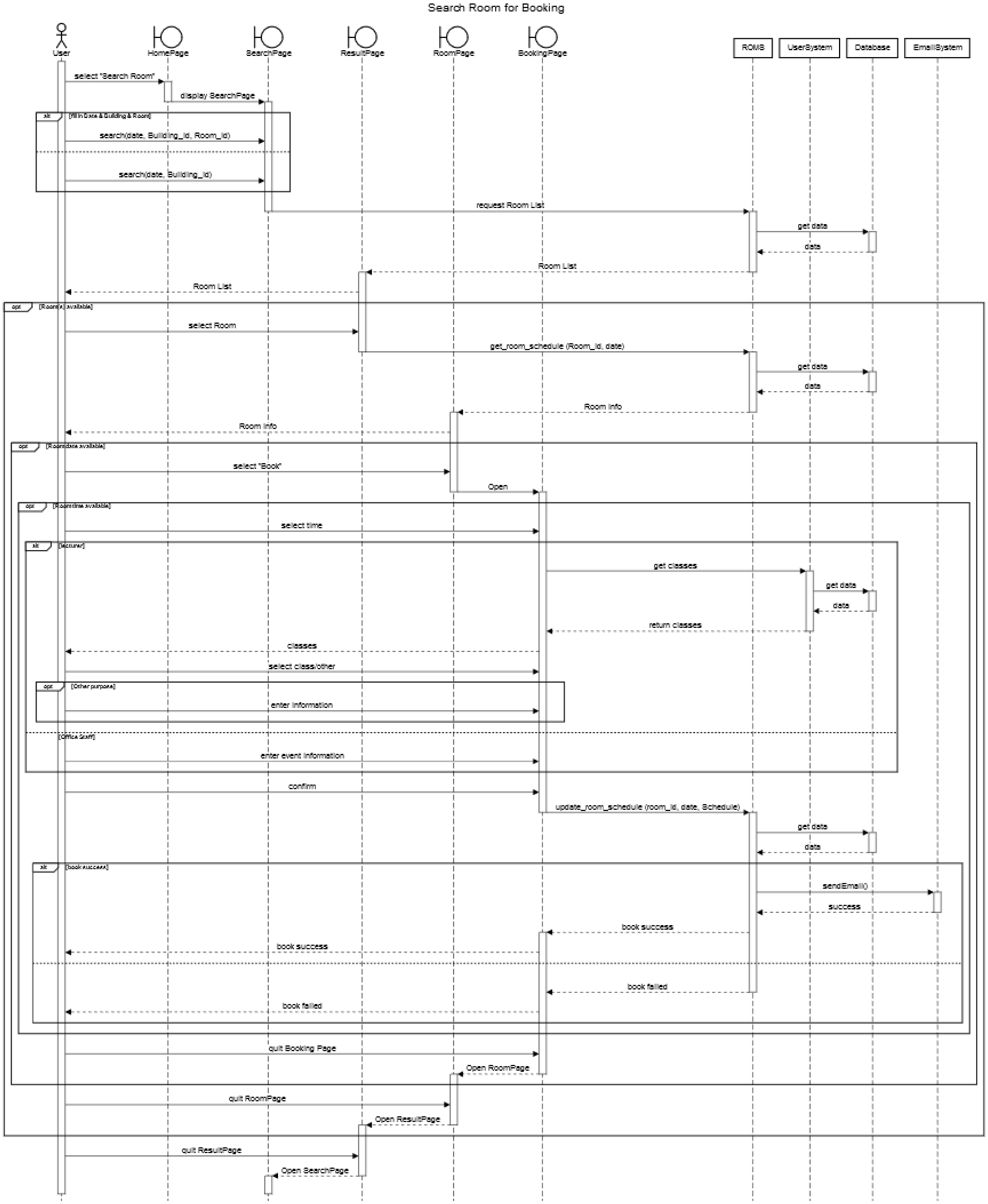


Figure 2.2: Sequence Diagram for Search Room for Booking

1.3. Cancel Booking

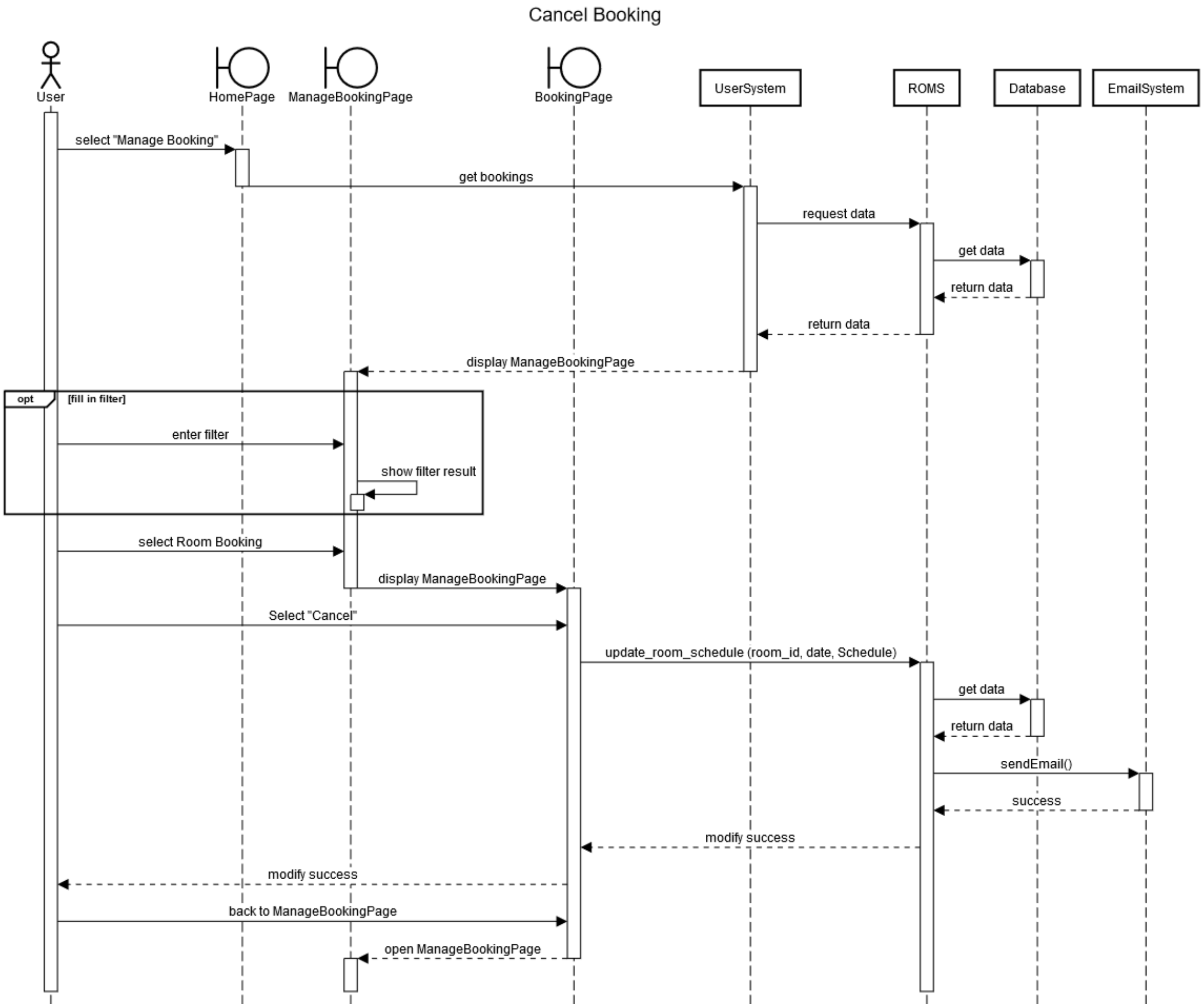


Figure 2.3: Sequence Diagram for Cancel Booking

1.4. Modify Booking

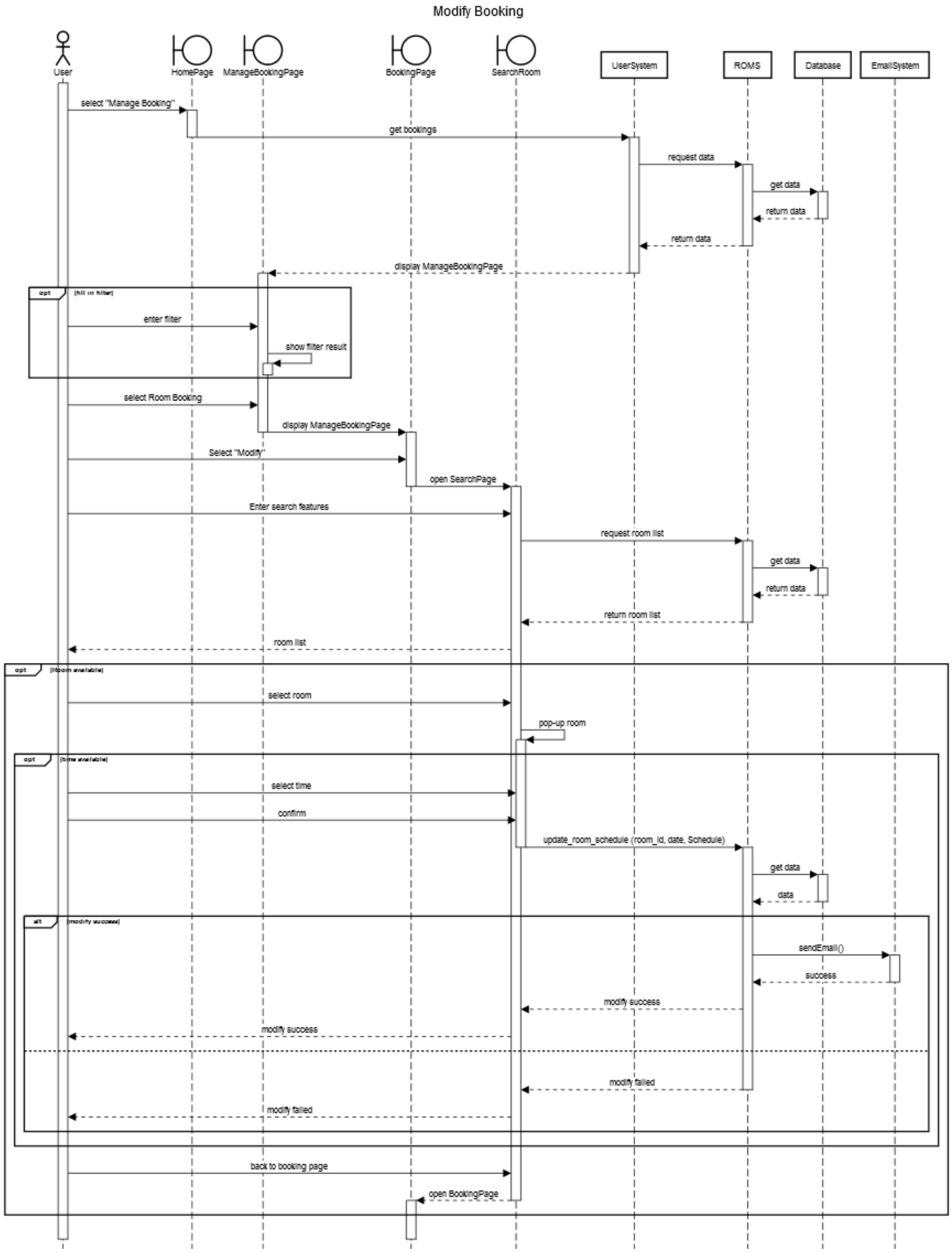


Figure 2.4: Sequence Diagram for Modify Booking

2. State Diagram for a Class Room

2.1. State-Chart Diagram

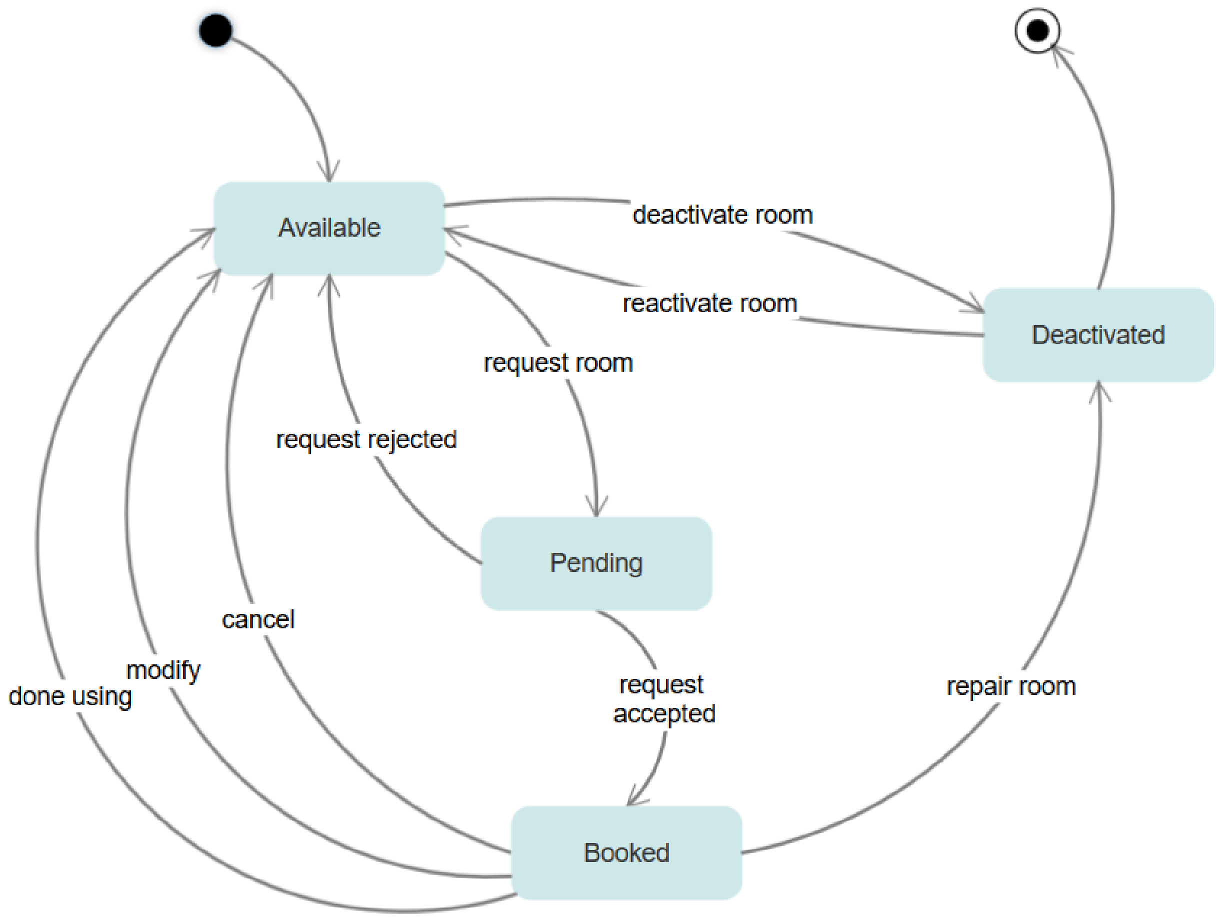


Figure 2.5: State Diagram for a Class Room

2.2. States and stimuli for the Class Room

State	Description
Available	The room is waiting to be booked
Pending	The room is hold for 10 minutes approximated during the booking process to wait for data from Database. Other will see it as booked.
Booked	The room is booked. Other will see it as booked.
Deactivated	Room is no longer used or being repair.

Stimulus	Description
Request Room	User has pressed the "Confirm" button on booking page
Request Rejected	ROMS has returned a failed message.
Request Accepted	ROMS has returned a success message.
Cancel	User has pressed "Cancel" on the booking manage section
Modify	User has finished modifying booking by pressing "Confirm" on the booking manage section and other room/ other time is booked.
Done Using	User has done using the room and it becomes available again.
Deactivate Room	Administrator has pressed on "Deactivate Room" on the admin page
Reactivate Room	Administrator has pressed on "Reactivate Room" on the admin page
Repair Room	Administrator has pressed on "Deactivate Room" on the admin page

3. Activity Diagram

3.1. Activity Diagram for Search Room processing

There are **several external systems** that the SCAMS connects to, in order to clear out **the process of data** between these systems when a user needs to search for rooms, the following activity diagram is drawn.

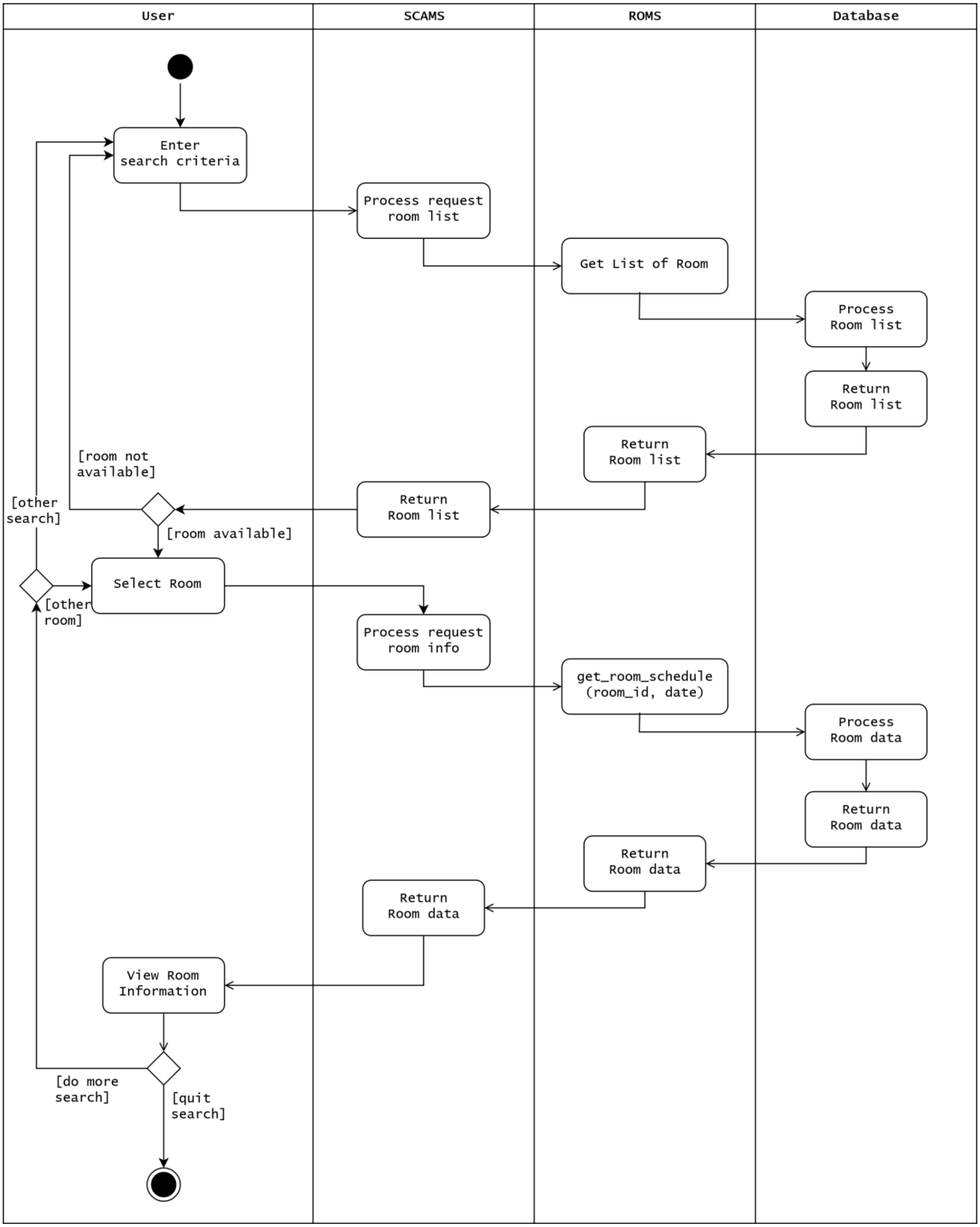


Figure 2.6: Activity Diagram for a Search Room Information Processing

3.2. Activity Diagram for Enter Information to book room processing

In the process of booking a room, **after user pressed the "Book" button** and **before he/she presses the "Confirm" button**, there are several steps the user must follow. The below activity diagram will clear out that process.

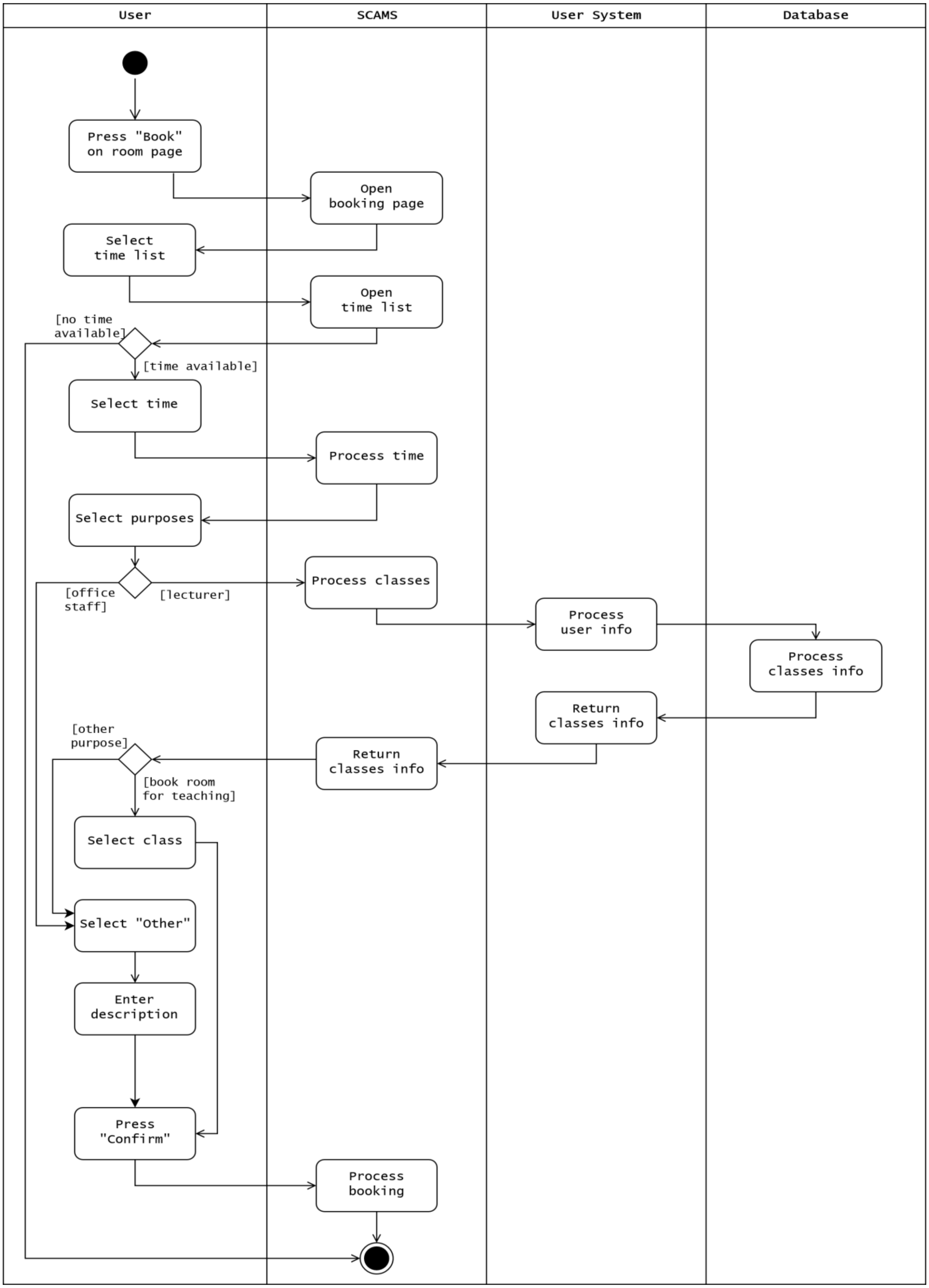


Figure 2.7: Activity Diagram for Enter Information to book room processing

3.3. Activity Diagram for Modify Booking processing

The SCAMS allows user to modify their booking, however, before changing to a new time or date, its availability must be checked. This activity diagram will focus on the modify process from the moment **user has pressed the “Modify” button** on manage booking section until he/she **select the “Confirm” button**.

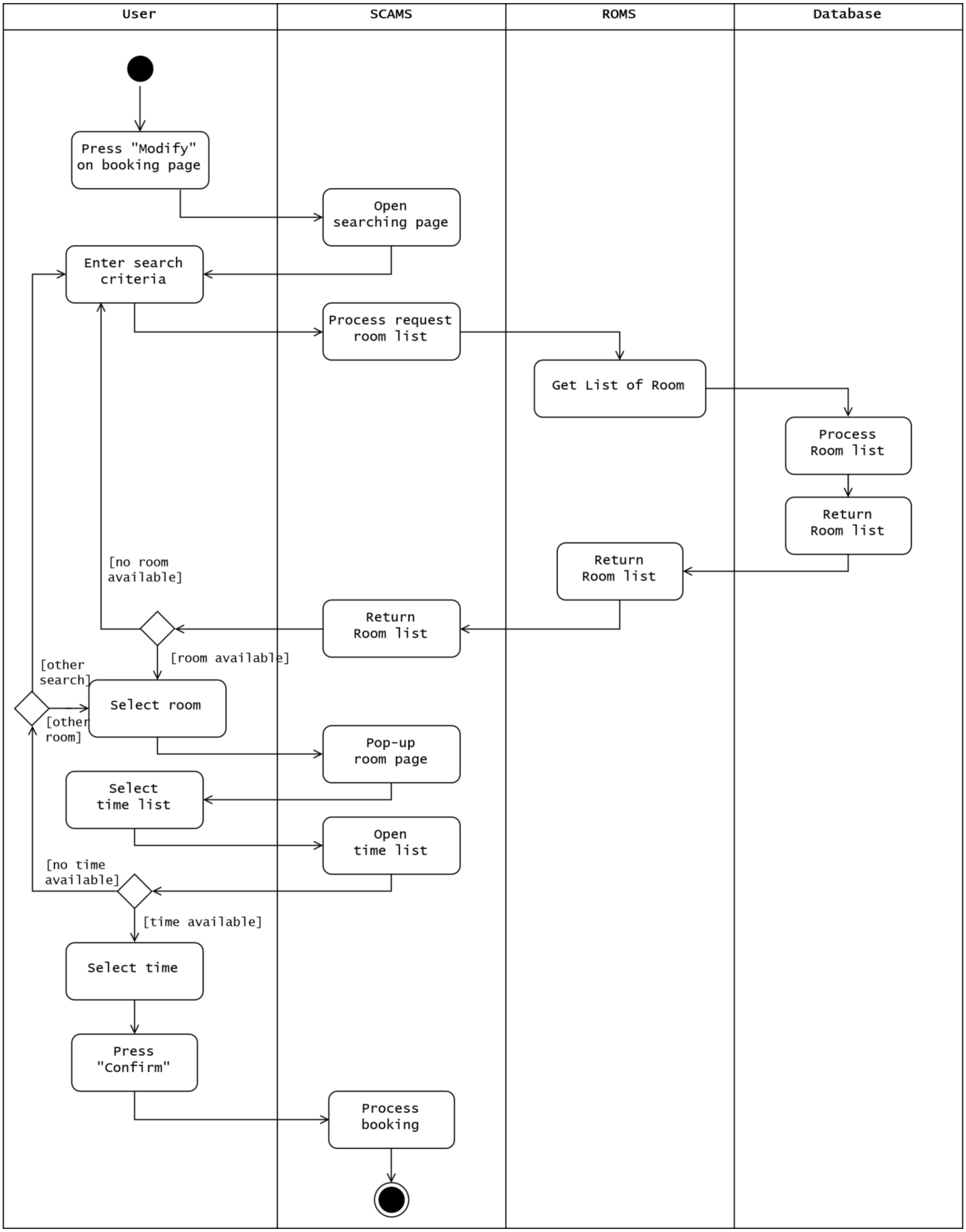


Figure 2.8: Activity Diagram for Modify booking processing

PART 3: ASSIGNMENT 3

1. System Deployment View

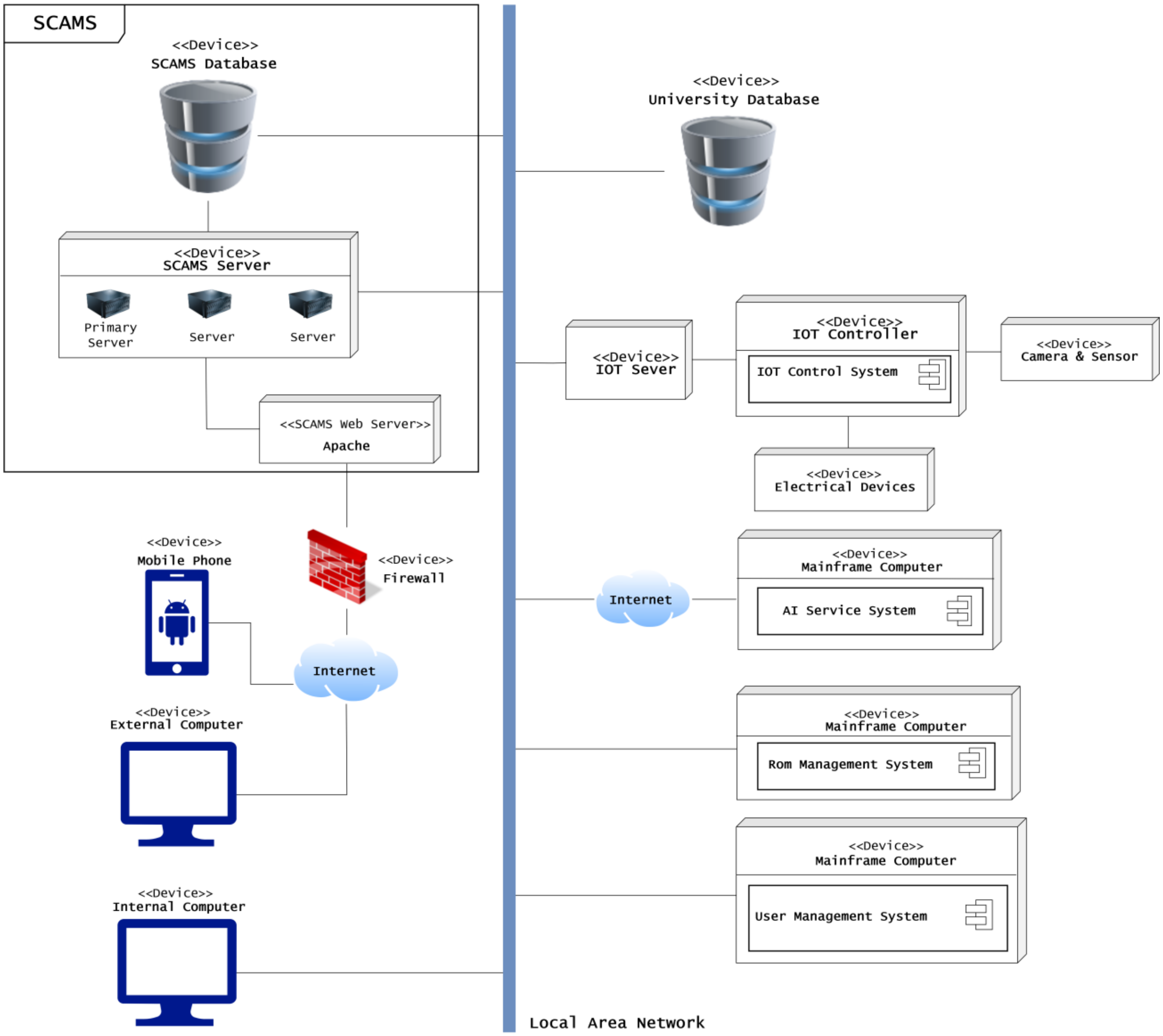


Figure 3.1: SCAMS Deployment View

The above figure shows deployment view of our system. Elements inside the system and other elements that the system connects to (University Database, IOT Server, ROMS, User Management System and Internal Computer) are connecting together through LAN. User connects through the internet have to pass the Firewall to access the application. The IOT Server connects to the IOT controller and IOT devices through specific IOT connection protocol (will be defined later on). Connection with AI Service is through the internet.

2. Development View

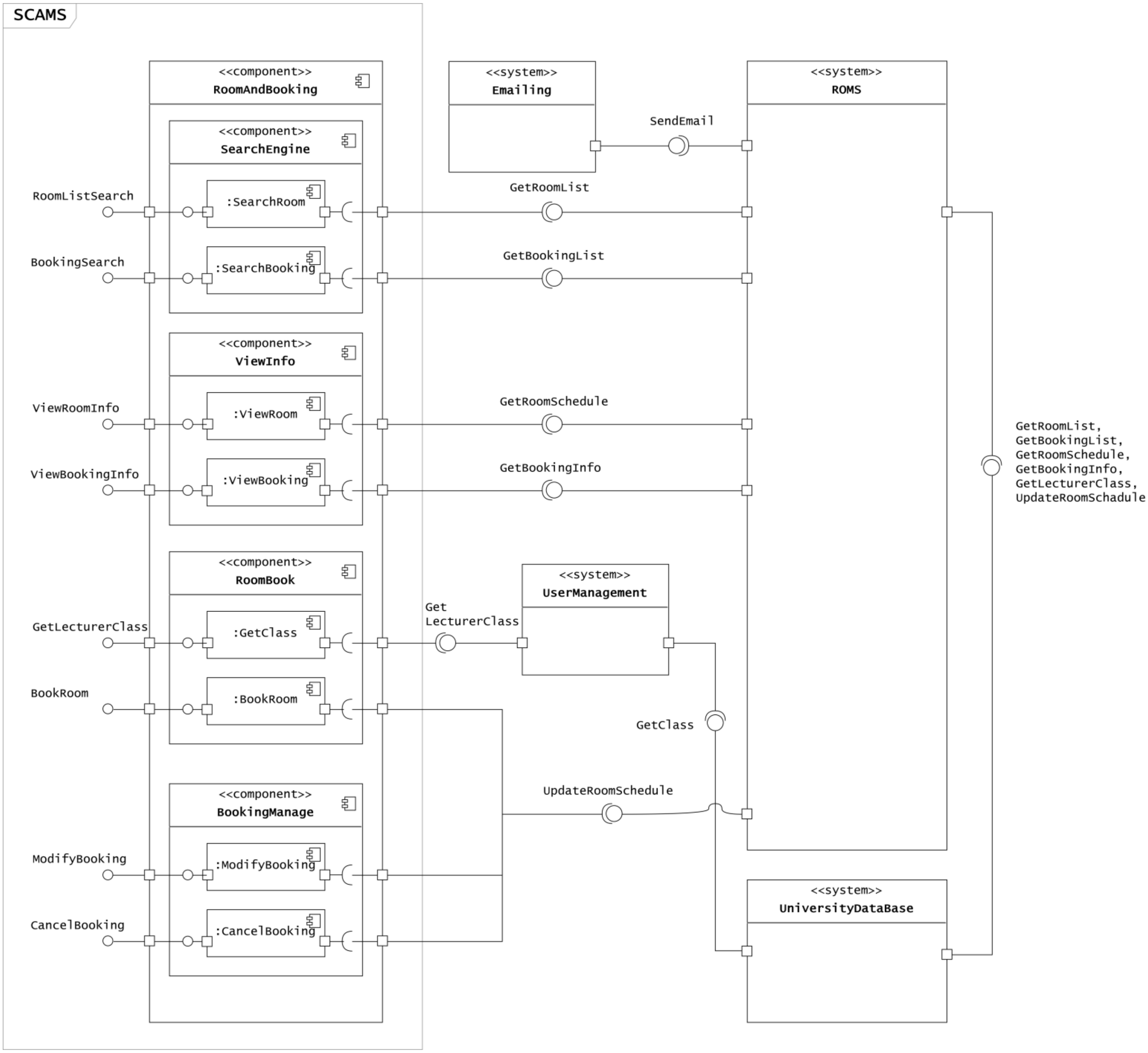


Figure 3.2: Component Diagram for Search Room, Book & Cancel & Modify Room Functions

The above component diagram shows the Development View of the following functions:

- Search Room Information / Booking.
- Book Room
- Cancel Room
- Modify Room

The above diagram only focus on the SCAMS system, it does not specify detailed components of other systems that the SCAMS connects to (ROMS, User Management, Emailing, University Database). Only interfaces that they require and provide are specified.

PART 4: ASSIGNMENT 4

1. Group work

1.1. Software Structure – MVC Paradigm

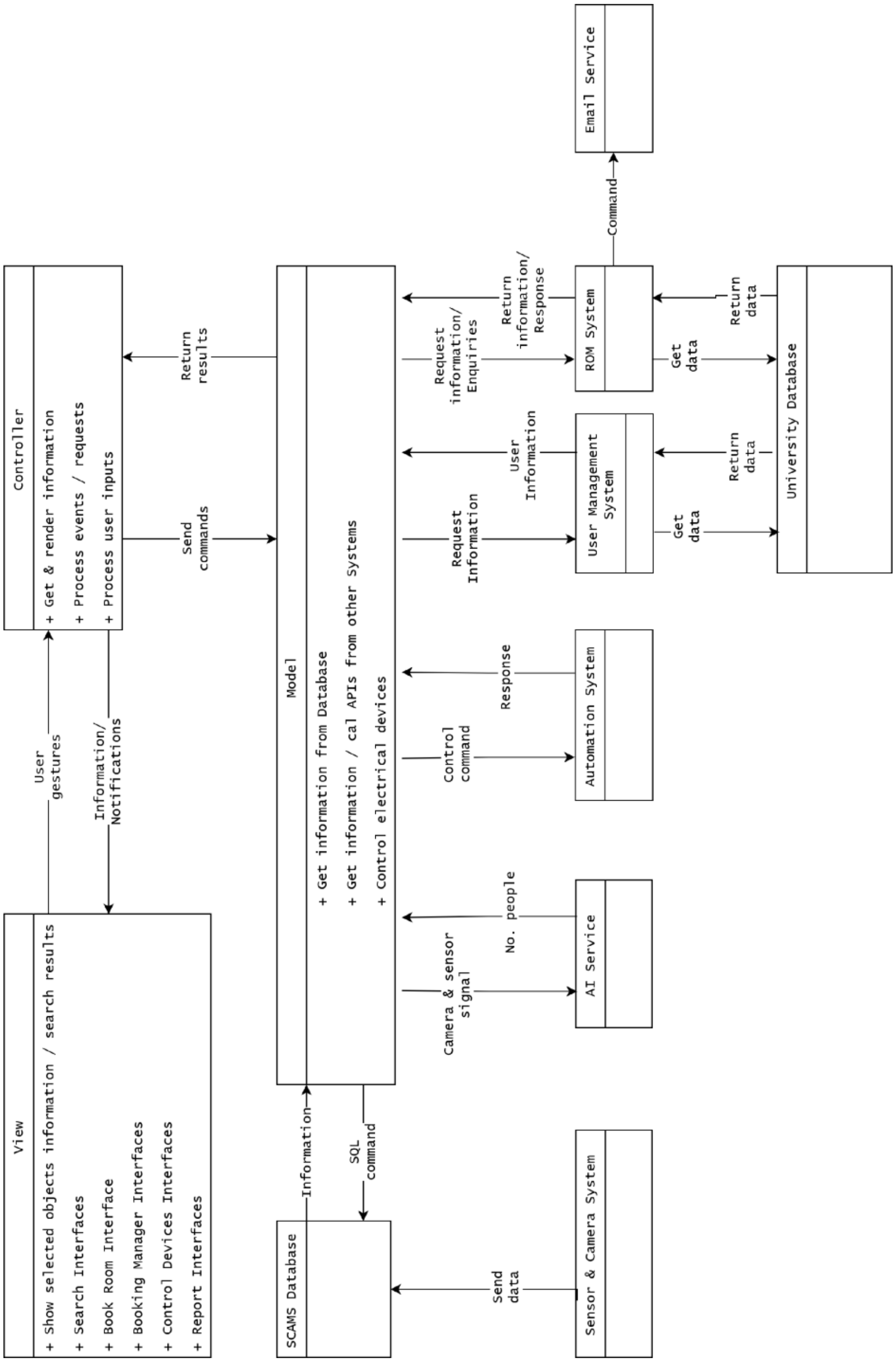


Figure 4.1: SCAMS MVC Paradigm

Our MVC view is quite different from the original MVC view, where Model updates View directly. Here Model gets all information and send it to Controller to render the view.

From this viewpoint:

(1) User inputs some actions/ information (clicking button, completing form, etc.) from the View.

(2) Controller will determine what to do with User's gesture/information, and send it off to the right Model.

(3) Models are Classes in our system, it will process the information from Controller, connects to Databases, call APIs from external Systems and return results to the Controller.

(4) After receiving the result, Controller will send back information to the User with appropriate View.

1.2. System Class Diagram

The below diagram is class diagram for our system. This is only the Domain Classes with some primary methods.

Each member in our team will implement these methods as closure for the part they are in charge of into Implement Level Class Diagrams.

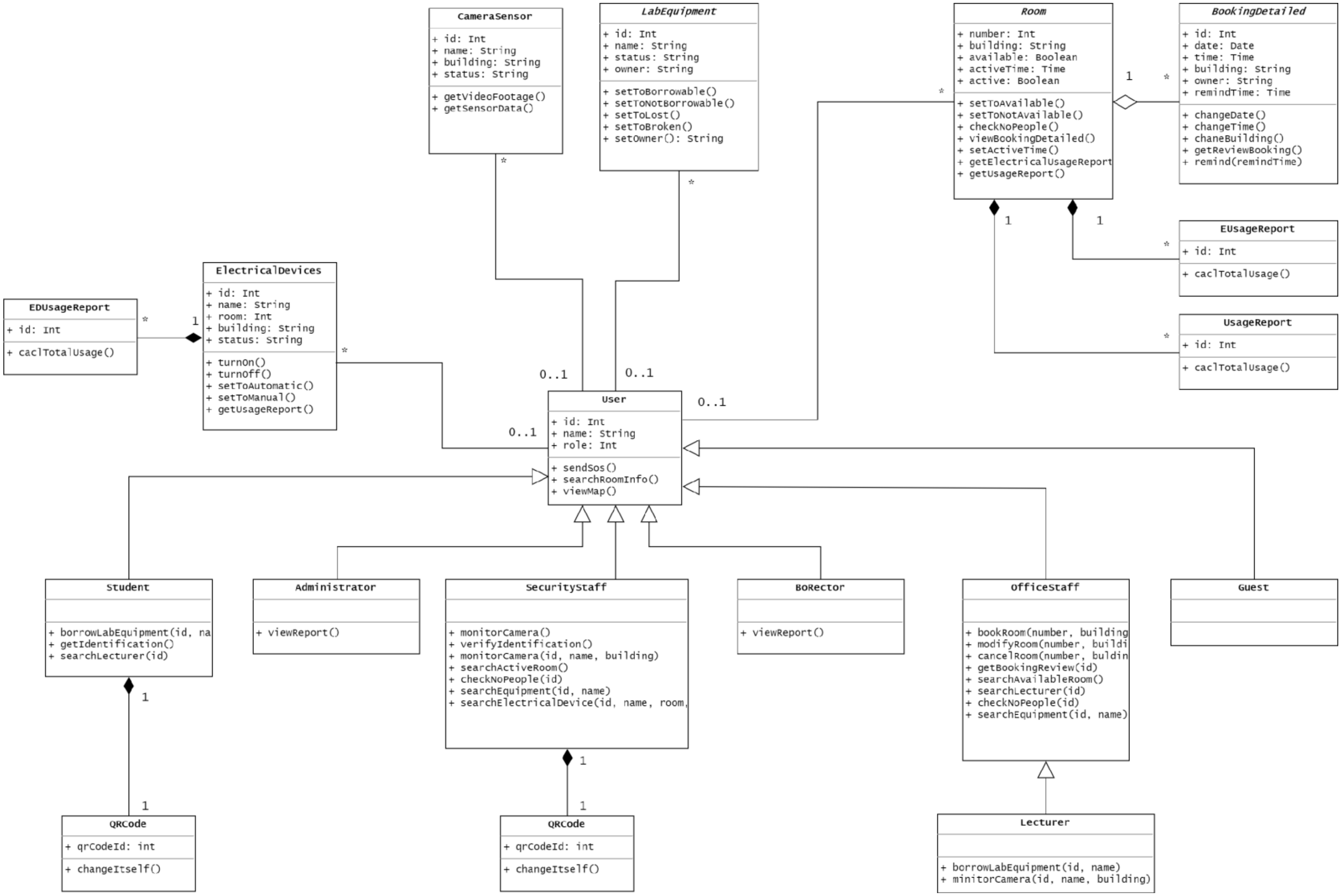


Figure 4.2: SCAMS Class Diagram

2. Individual work

In charge functions:

- Search room for information & booking
- Book Room
- Cancel Booking
- Modify Booking

Since these functions relates to each other, there will be two Class Diagram in this assignment, which are:

- Class Diagram for Search Room and Book Room
- Class Diagram for Modify and Cancel Booking

2.1. Search Room and Book Room

2.1.1. Class Diagram for Search Room and Book Room

In this Class Diagram, only classes related to the function Search Room and Book Room are implemented and displayed.

Let’s take a look back these functions descriptions:

- Every user can search room for information.
- Only Office Worker (Lecturer and Office Staff) is allowed to book room.

This diagram shows how related classes are implemented and relationships between them during the searching and booking process. They do not specify how external systems interact with each other than SCAMS.

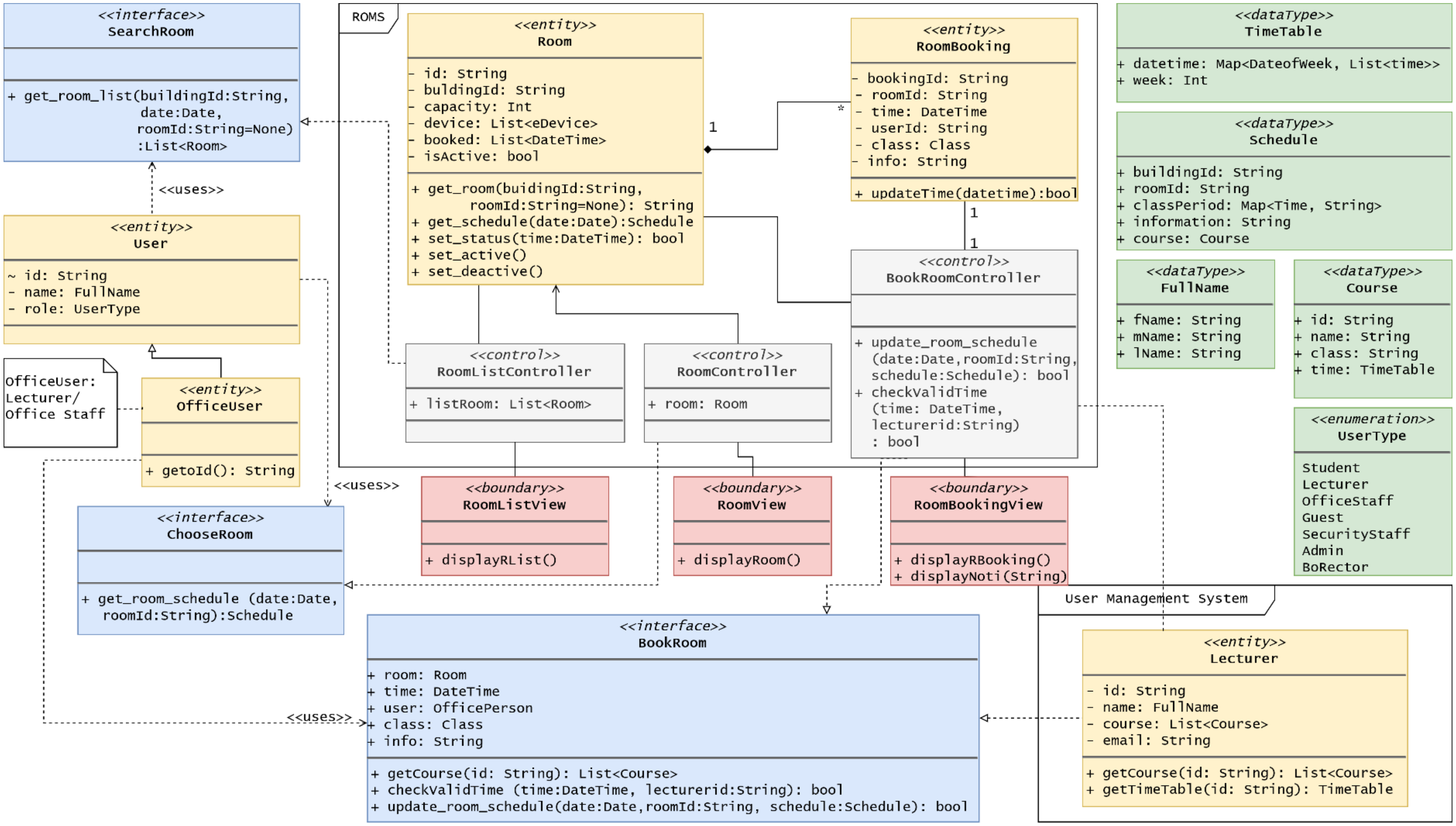


Figure 4.3: Class Diagram for Search Room and Book Room

2.1.2. Module Programming Interface

Interface	Description
List<Room> **get__room__list** (buildingId: String, date: Date, roomId: String=None)	Return the list of room(s) in the search result view after user entered search criteria and clicked "Search".
Schedule **get__room__schedule** (date: Date, roomId: String)	Return the schedule of the room in a particular date, which contains list of 24 hours with the occupancy information.
List<Course> **getCourse**(id: string)	Return a list of course(s) of the Lecturer.
bool **checkValidTime** (time: DateTime, lecturerTime: TimeTable, id: userId)	Return a boolean value indicates if the chosen time is conflict with the timetable of Lecturer or not.
bool update__room__schedule (date: Date, roomId: String, schedule: Schedule)	Update the schedule into the ROMS with the Schedule of the room. Return a boolean value indicate if the process is successful or not.

2.1.3. Method Descriptions

Method	Description
getoId (): String	Return Office User Id
updateTime(datetime): bool	Update time and date for a room booking and return a boolean value indicates if the process is successful.
get_room (buidingId: String, roomId: String=None): String	Return Room ID if it exists in the Database
get_schedule (date:Date): Schedule	Return 24 hours schedule usage of the room.
set_status (time: DateTime): bool	Set the room to booked status and return a boolean value indicates if the process was successful.
set_active (): void	Set the room state from deactive to active.
set_deactive (): void	Set the room state from active to deactive.
getTimeTable(id: String): TimeTable	Return a timetable of a particular Lecturer in order to check if the chosen DateTime does not conflict with his/her timetable.
displayRList()	Display List of Room on result page after User entered search criteria and hit "Search".

displayRoom()	Display Room detailed information after User selected a room on search result.
displayRBooking()	Display Room Booking page after User selected "Book" on the room page.
displayNoti(String)	Display notifications on screen for user

2.1.4. Sequence Diagrams

Due to the large amount of interactions between classes in the system and external system, the Search and Book Room process will be divided into two Sequence Diagrams as below.

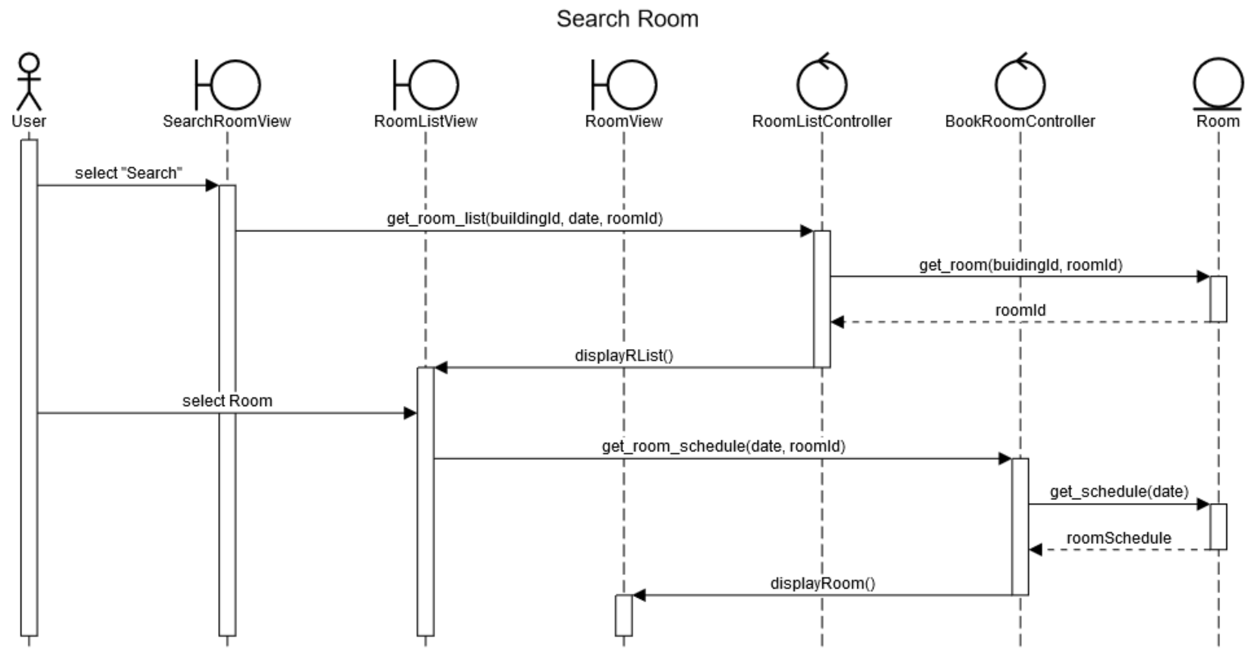


Figure 4.4: Sequence Diagram for Search Room

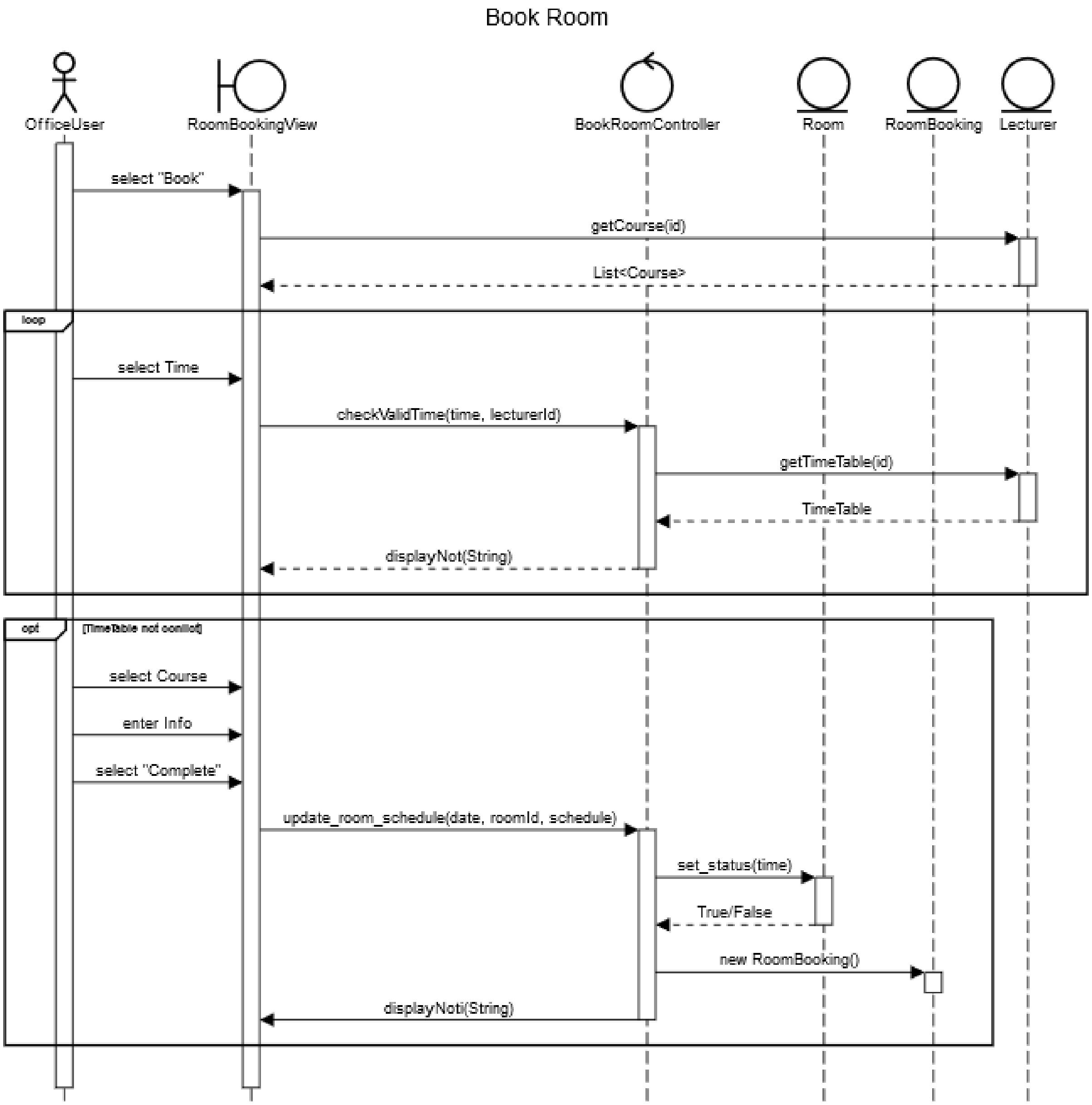


Figure 4.5: Sequence Diagram for Book Room

2.2. Modify and Cancel Room Booking

2.2.1. Class Diagram for Modify and Cancel Room Booking

In this Class Diagram, only classes related to the function Modify and Cancel Booking are implemented and displayed. The differences between it and the previous class diagram are bold bordered classes/interfaces. Furthermore, there will be modifications to simplify the system. Please see sequence diagrams for more details.

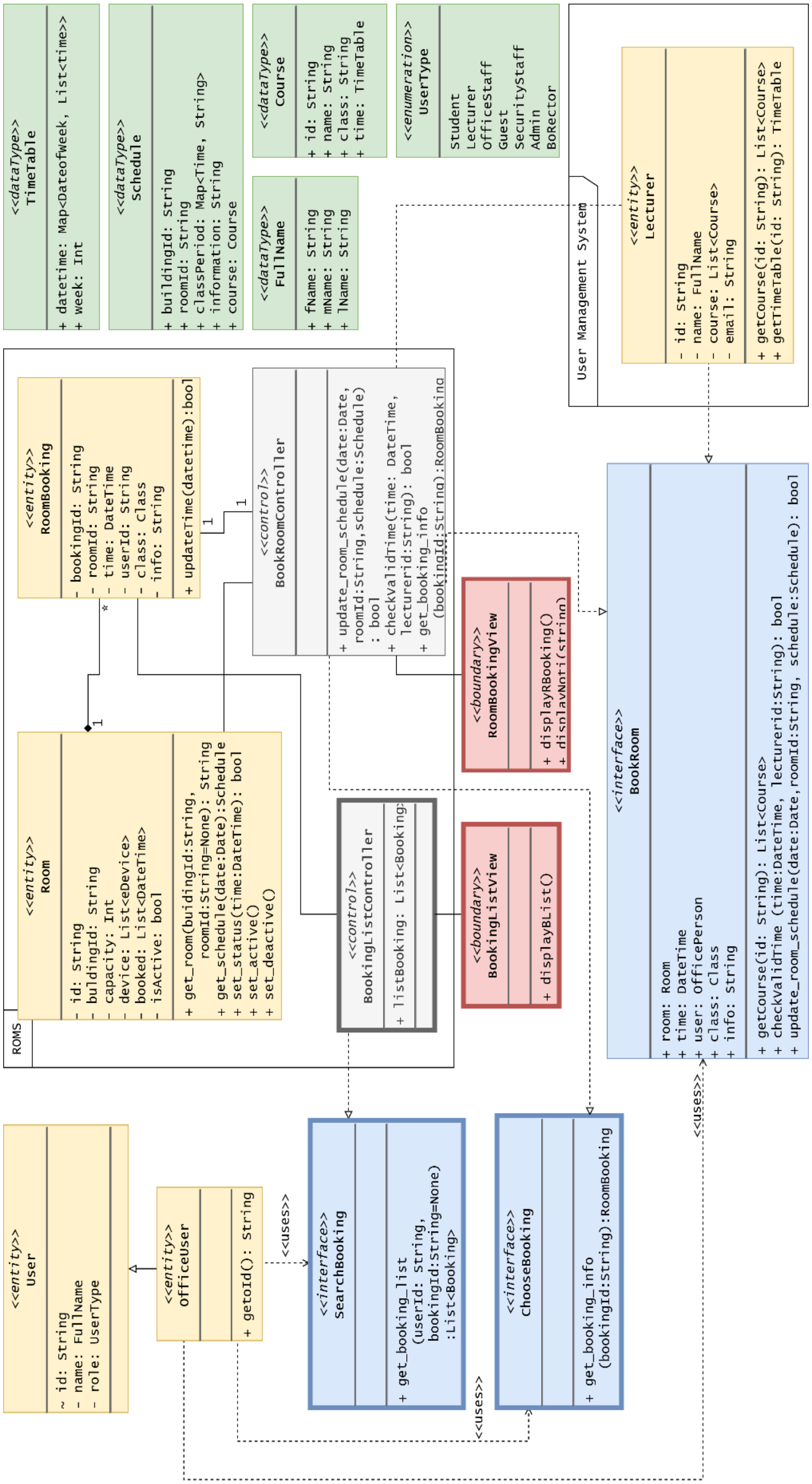


Figure 4.6: Class Diagram for Cancel and Modify Booking

2.2.2. Module Programming Interface

Notes: only different interfaces from the previous section's interfaces are listed.

Interface	Description
List<Booking> **get_booking_list** (userId: String, bookingId: String=None)	Return the list of booking(s) in the search result view after user entered search criteria and clicked "Search".
Schedule **get_booking_info** (bookingId: String)	Return the booking information of the user.

2.2.3. Method Descriptions

No new method descriptions.

2.2.4. Sequence Diagrams

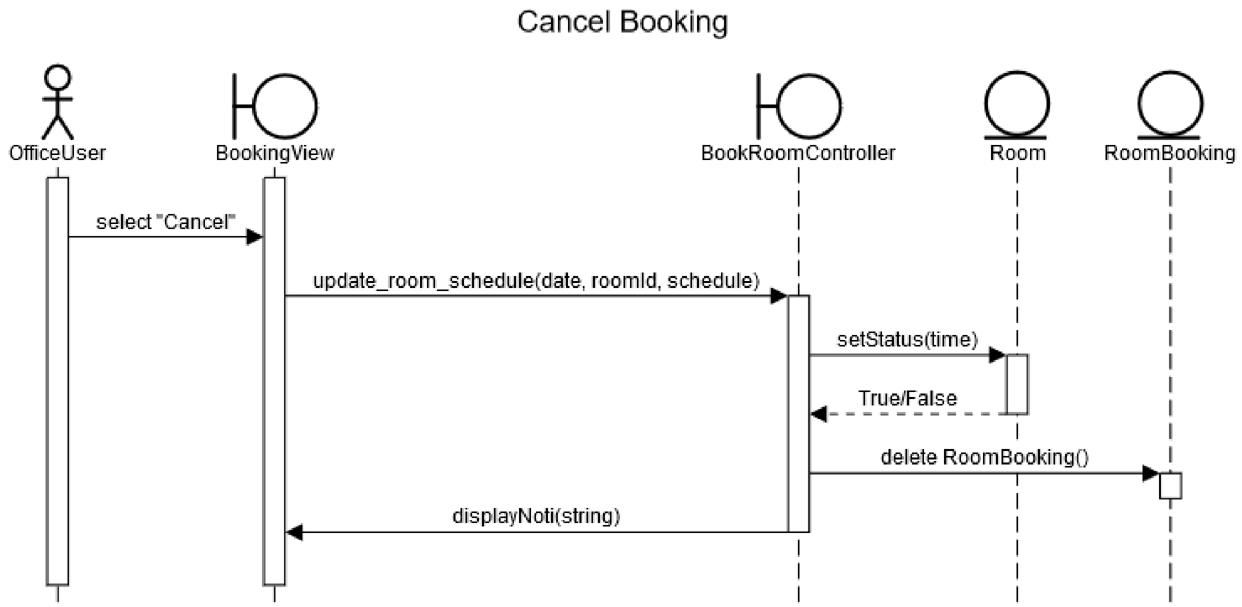


Figure 4.7: Sequence Diagram for Cancel Booking

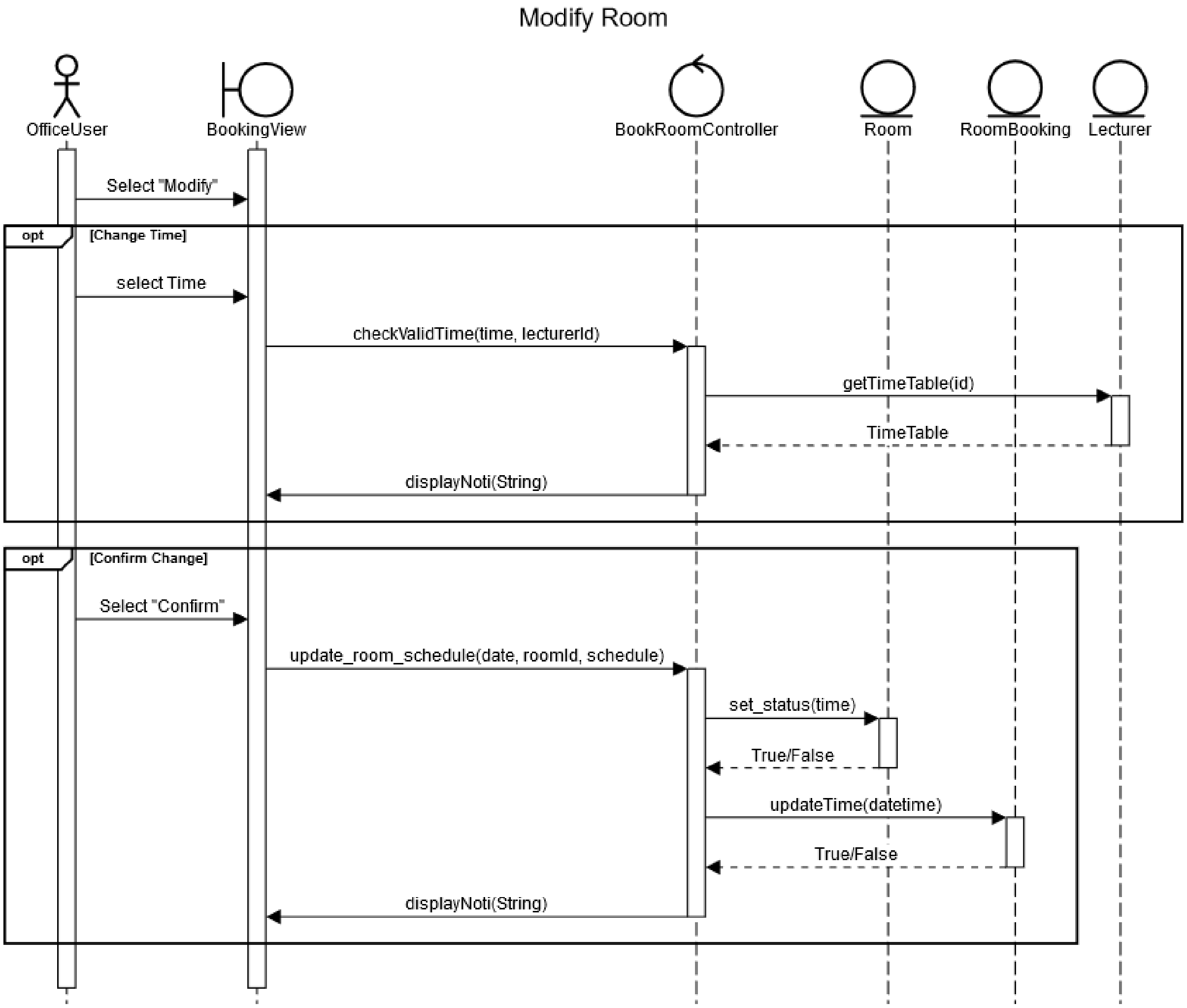


Figure 4.8: Sequence Diagram for Modify Booking

2.3. State Machine: Room

To simplify the SCAMS system, the Pending state is not implemented. There are three states for a room object as described in the following diagram.

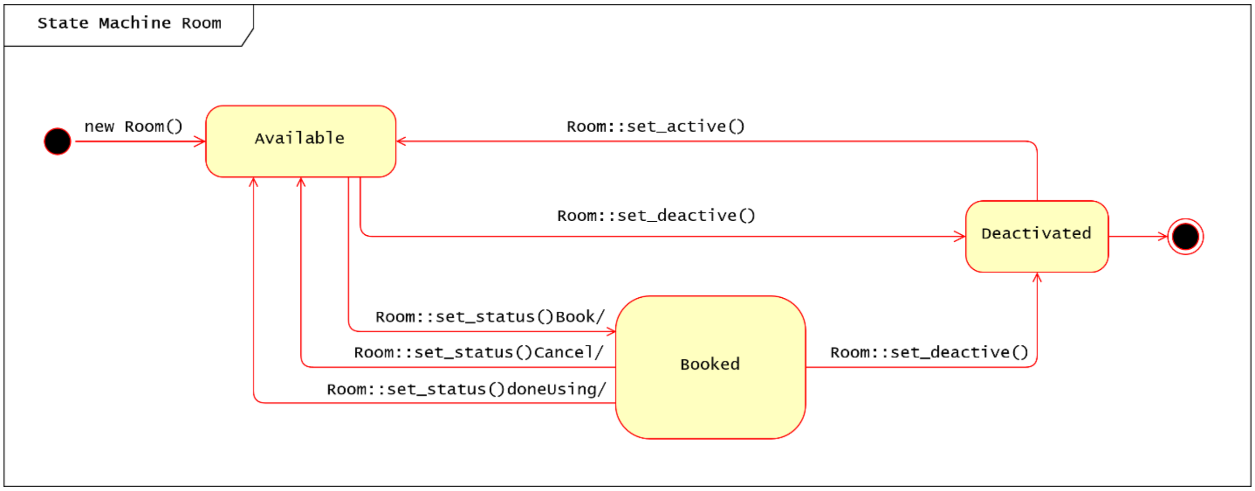


Figure 4.9: State Machine Diagram for Room

2.4. MVC Design Pattern

Our system design will be using the MVC design pattern.

For our program design we will be using the MVC model.

- View: Every view has a display method that is called by the Controller.
- Data Model: Every entity in our system is a Model, each will be implemented its own methods, which are called by Controllers.
- Controller: Every activity will be processed by Controller, it determines which method should be called and render views for Boundary Classes.

2.4.1. Data Model Class

The following entity classes are Data Model Classes. They implement methods on entity objects.

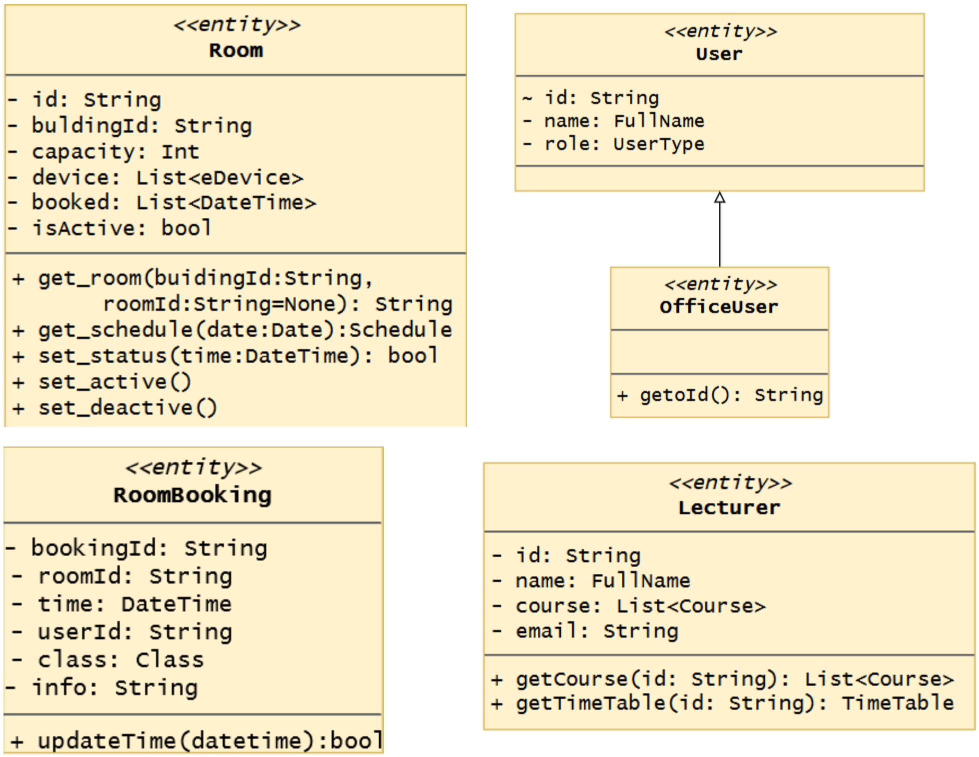


Figure 4.10: Data Model Classes

2.4.2. Controller

Controllers implements interfaces SCAMS provides.

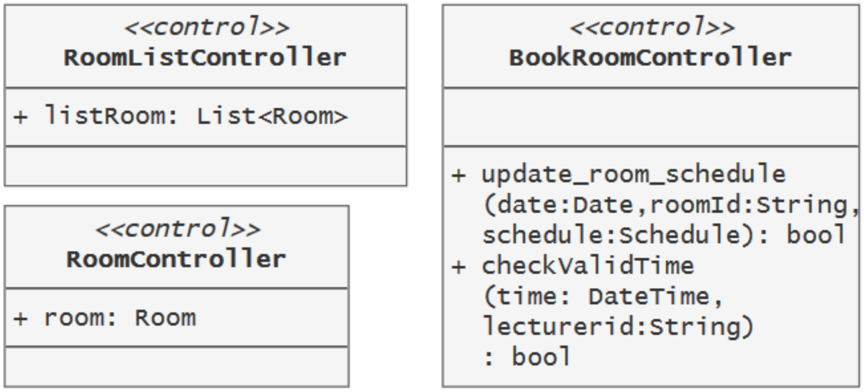


Figure 4.10: Control Classes

2.4.3. View

Views are updated by Controllers, each implements the display() method to display information to User.

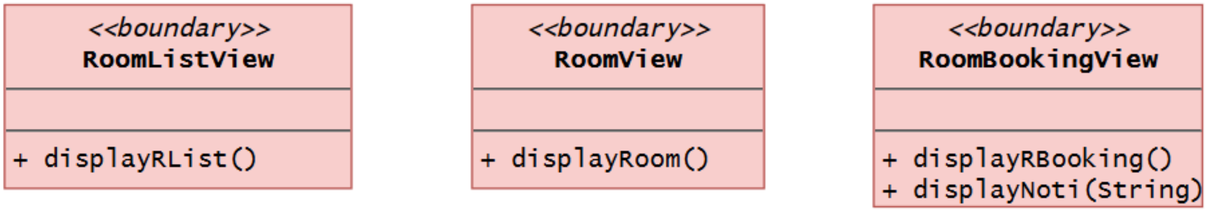
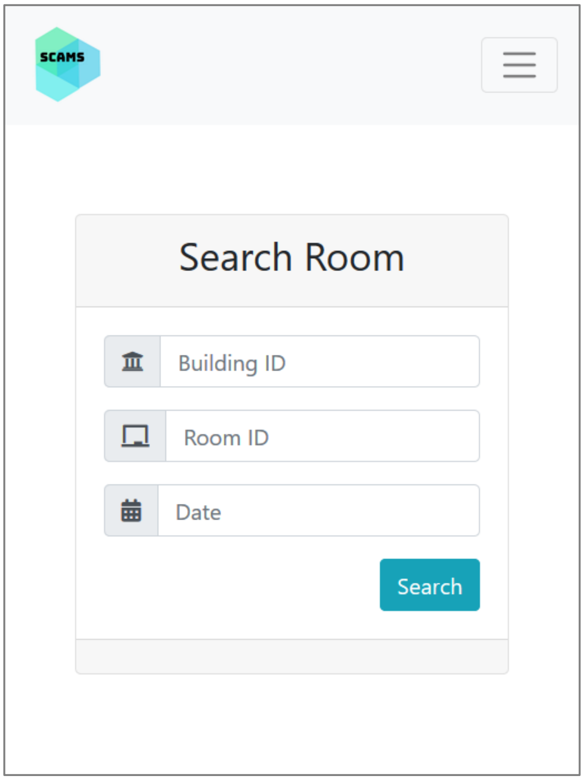


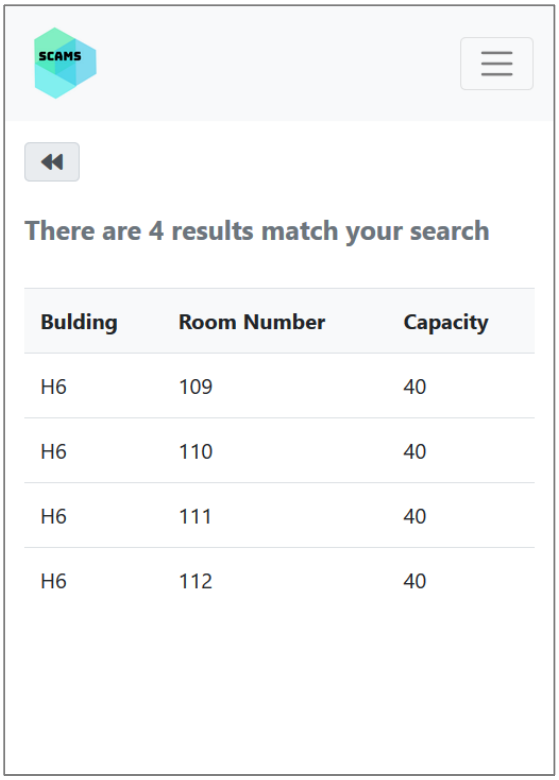
Figure 4.11: Boundary Classes

2.5. User GUI

This section shows sequences of screenshots of how User interacts with SCAMS.

2.5.1. Search Room





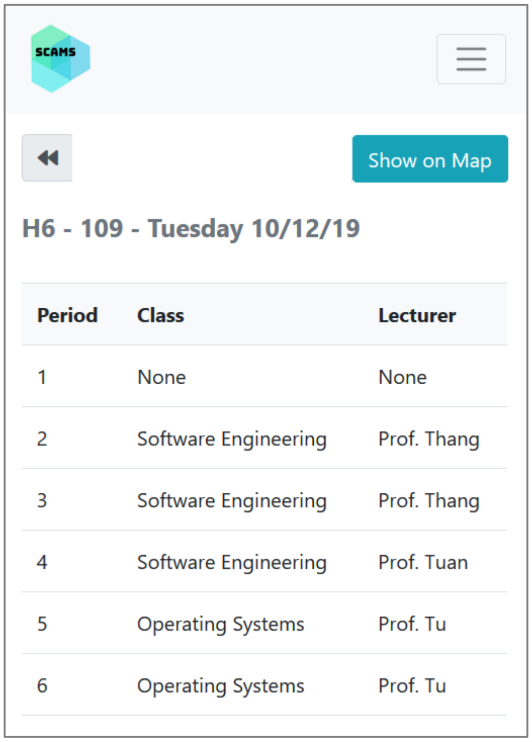
There are 4 results match your search

Bulding	Room Number	Capacity
H6	109	40
H6	110	40
H6	111	40
H6	112	40

Figure 4.12: Search Panel GUI

Figure 4.13: Room List GUI

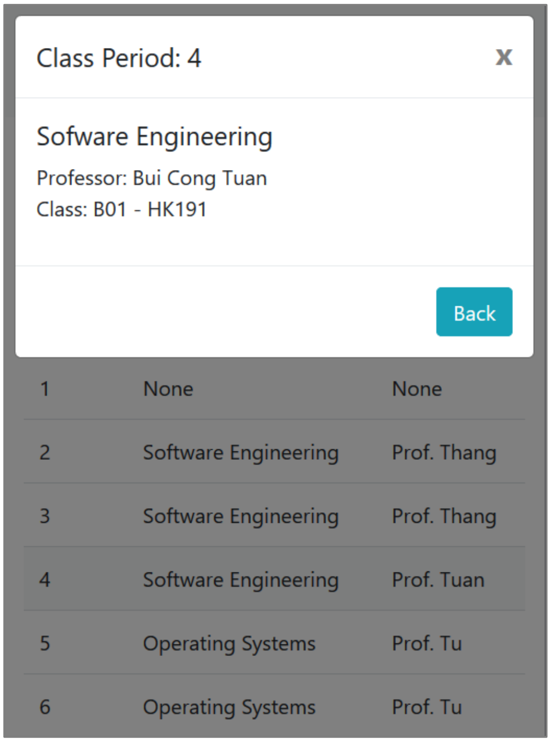
2.5.2. View Room Information



H6 - 109 - Tuesday 10/12/19

Period	Class	Lecturer
1	None	None
2	Software Engineering	Prof. Thang
3	Software Engineering	Prof. Thang
4	Software Engineering	Prof. Tuan
5	Operating Systems	Prof. Tu
6	Operating Systems	Prof. Tu

Figure 4.14: Room Page GUI



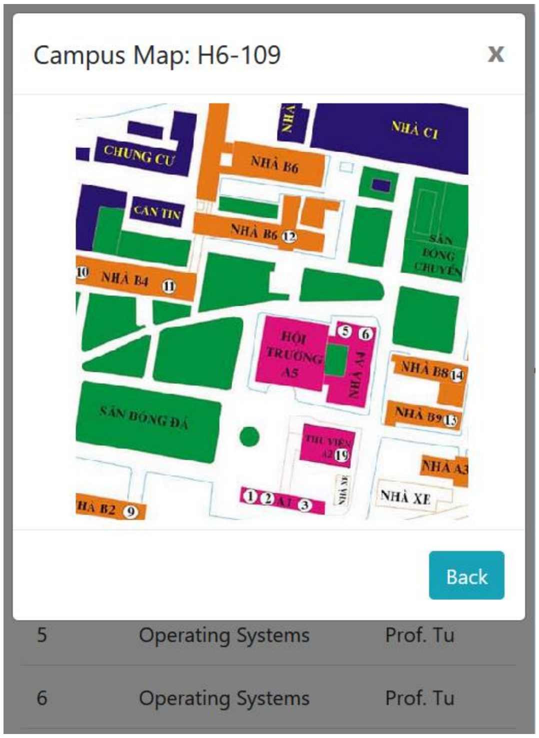
Class Period: 4

Sofware Engineering

Professor: Bui Cong Tuan
Class: B01 - HK191

1	None	None
2	Software Engineering	Prof. Thang
3	Software Engineering	Prof. Thang
4	Software Engineering	Prof. Tuan
5	Operating Systems	Prof. Tu
6	Operating Systems	Prof. Tu

Figure 4.15: Clicking on a Class Period



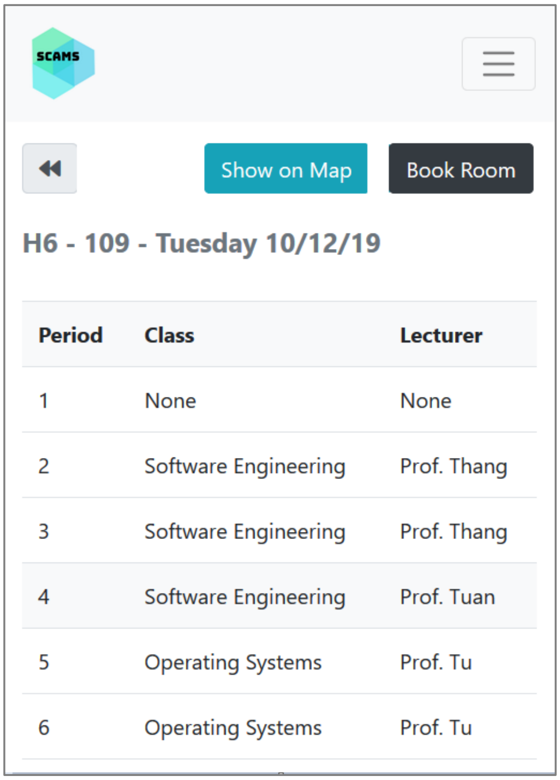
Campus Map: H6-109

5	Operating Systems	Prof. Tu
6	Operating Systems	Prof. Tu

Figure 4.16: Clicking on "View on Map"

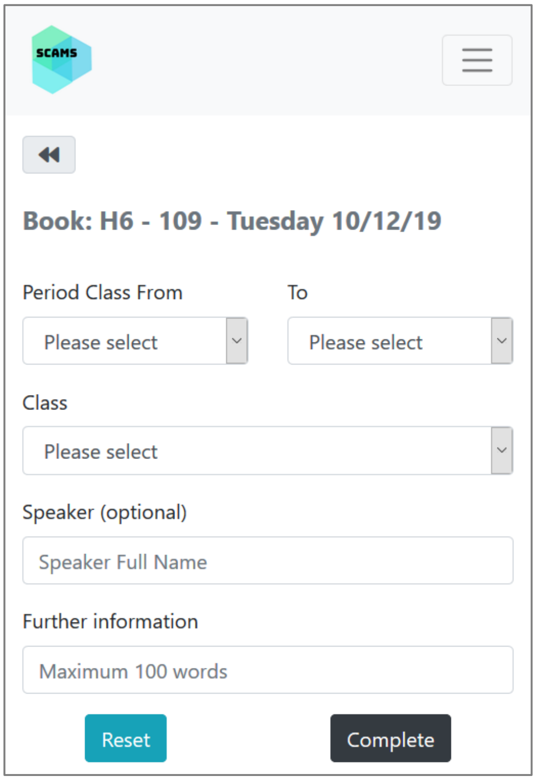
2.5.3. Book Room

If the user is Office User (Lecturer or Office Staff), he/she is allowed to book room on the Room Page GUI.



H6 - 109 - Tuesday 10/12/19

Period	Class	Lecturer
1	None	None
2	Software Engineering	Prof. Thang
3	Software Engineering	Prof. Thang
4	Software Engineering	Prof. Tuan
5	Operating Systems	Prof. Tu
6	Operating Systems	Prof. Tu



Book: H6 - 109 - Tuesday 10/12/19

Period Class From: Please select

To: Please select

Class: Please select

Speaker (optional): Speaker Full Name

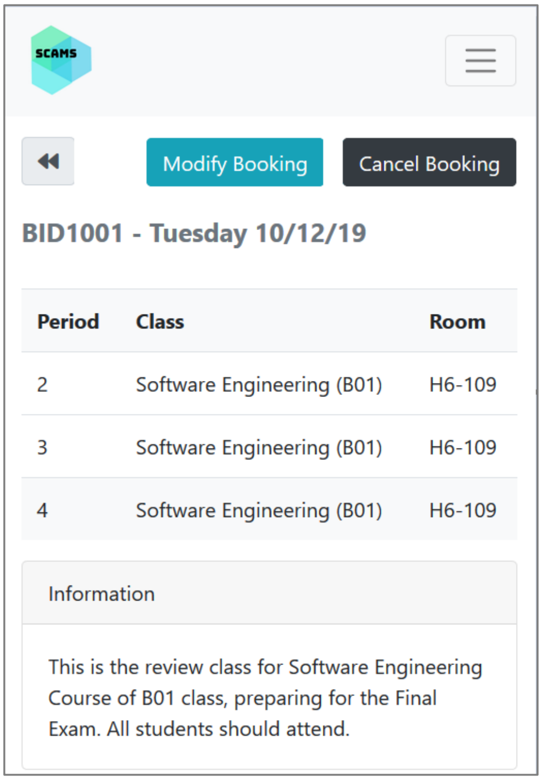
Further information: Maximum 100 words

Figure 4.17: Room Page GUI

Figure 4.18: Book Room GUI

2.5.4. Cancel and Modify Room Booking

After accessing to the Manage Booking Page and searching for the list of Bookings, User can review his/her booking and can also cancel/modify it.



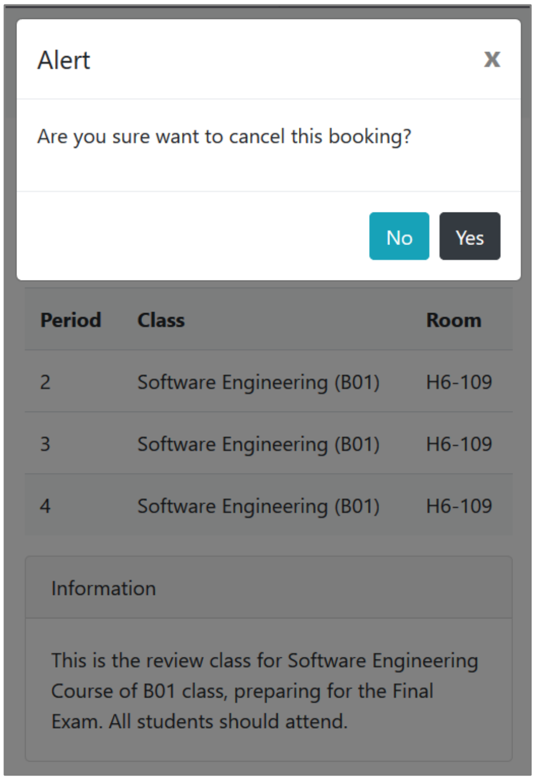
BID1001 - Tuesday 10/12/19

Period	Class	Room
2	Software Engineering (B01)	H6-109
3	Software Engineering (B01)	H6-109
4	Software Engineering (B01)	H6-109

Information

This is the review class for Software Engineering Course of B01 class, preparing for the Final Exam. All students should attend.

Figure 4.19: Booking Review Page GUI



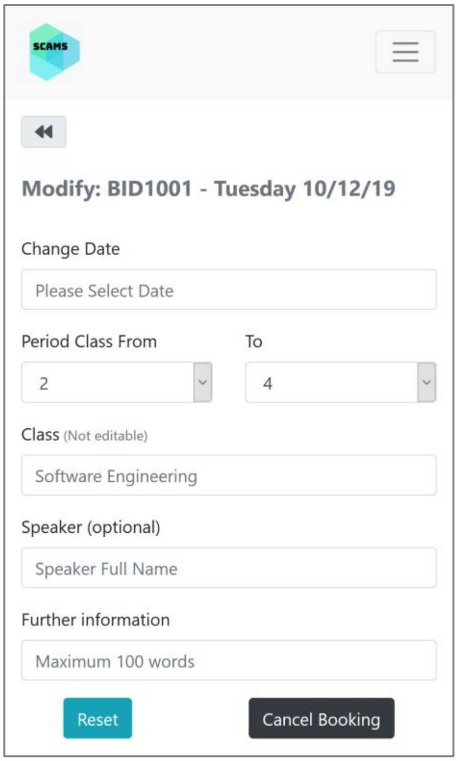
Are you sure want to cancel this booking?

Period	Class	Room
2	Software Engineering (B01)	H6-109
3	Software Engineering (B01)	H6-109
4	Software Engineering (B01)	H6-109

Information

This is the review class for Software Engineering Course of B01 class, preparing for the Final Exam. All students should attend.

Figure 4.20: Clicking on Cancel Booking



Modify: BID1001 - Tuesday 10/12/19

Change Date: Please Select Date

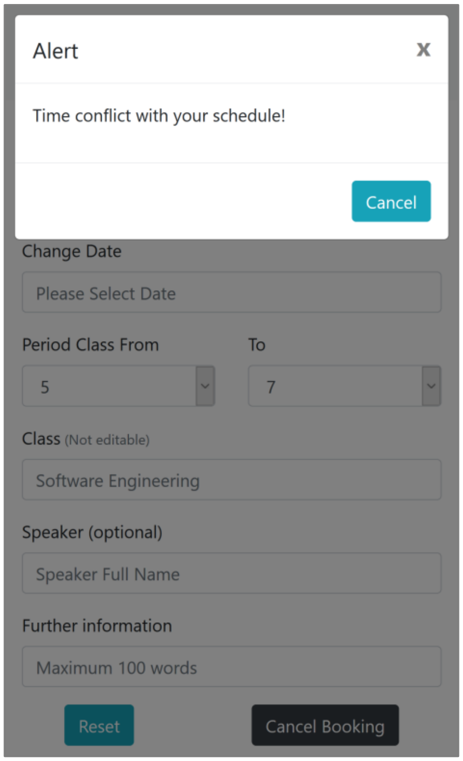
Period Class From: 2 To: 4

Class (Not editable): Software Engineering

Speaker (optional): Speaker Full Name

Further information: Maximum 100 words

Figure 4.21: Modify Booking Page GUI



Time conflict with your schedule!

Change Date: Please Select Date

Period Class From: 5 To: 7

Class (Not editable): Software Engineering

Speaker (optional): Speaker Full Name

Further information: Maximum 100 words

Figure 4.22: Time Conflict Alert

3. Conclusion

This assignment completes the Software Design Document (SDD), which include the system architecture and design descriptions for SCAMS.

During the implement process, some simplifies are made to the system to optimize its performance.

4. Group list

Student ID	Full Name
1827005	Tran Thi Ngoc Diep
1820047	Luong Thanh Nhan
1820034	Nguyen Dang Khoa
1820031	Pham Tran Anh Huy
1827030	Ngo Hoang Ngoc Thanh