Software Requirements Specification

for

Lecturer Appointment Schedule Application

**Version 0.0.1 approved**

**Prepared by Trong, Nhung, Hai Anh, Tung**

**Group 10**

**October 4, 2021**

Table of Contents

Table of Contents 2

Revision History .

1. Introduction

1.1 Purpose 3

1.2 Document Conventions 3

1.3 Project Vision and Product Features…………………………………...………………………..3

1.4 References 3

2. Overall Description

2.1 Product Perspective 4

2.2 User Classes and Characteristics 5

2.3 Operating Environment 5

2.4 Design and Implementation Constraints 5

2.5 Assumptions and Dependencies 5

3. System Features

3.1 Use case diagram……….………………………………………………………………6

3.2 System Feature 1……………………………………………………………………………….15

3.3 System Feature 2 (and so on)…………………………………………………………………..19

4. Data Requirements

4.1 Logical Data Model…………………………………………………………………………….21

4.2 Data Dictionary 22

4.3 Reports 26

4.4 Data Acquisition, Integrity, Retention, and Disposal 26

5. External Interface Requirements

5.1 User Interfaces 27

5.2 Software Interfaces 27

5.3 Hardware Interfaces 27

5.4 Communications Interfaces 27

6. Quality Attributes

6.1 Usability 27

6.2 Performance 27

6.3 Security 28

6.4 Safety 28

7. Internationalization and Localization Requirements

8. Other Requirements .

Appendix A: Glossary

Appendix B: Analysis Models

# Introduction

Students in FPT University have many problems related to study, learning-oriented, or skills that are needed for the major that they are studying. However, most of them do not know how to get help from the lecturer, and if they could, they are too afraid to contact their lecturer directly. University leaders recognize the need for a system to ease students seeking help known as *Lecturer Appointment Scheduling Application*

## Purpose

University leaders came up with an idea for an application, where students could search for the right lecturer who could solve their problems, book a meet with that lecturer, and ask questions. At the side of lecturers, they could arrange their free time to mentor student by create a schedule, and then students could do booking on it. Such an application could help students for timely support, and it would increase interactive ability between lecturer and students.  Choosing right lecturer’s topic and knowing what students want to ask before meeting would improve the efficiency of the meeting. Therefore, the application would give the convenient and effective, optimize for students and lecturers.

## Document Conventions

No special typographical conventions are used in this SRS.

## Project Vision and Product Features

The LASA will permit making a slot booking on the lecturer schedule, view, modify, and remove request .Create, view, modify, and delete available slots. Student can search for lecturer base on topic, lecturer’s name, time. The system will send notification wherever a booking request is created, or when booking request is accept/deny. A detailed description is available in the *Lecturer Appointment Scheduling Application* Vision and Scope Document [1], along with the features that are scheduled for full or partial implementation in this release.

## References

[www.perforce.com](http://www.perforce.com)

……………………

# Overall Description

## Product Perspective

The application would be updated to the latest version. We have plans to develop products that could work on many different devices: IOS, iPad, Android, etc..by setting configuration changes that are more suited to a production environment. Besides, the application could extend to other universities.

Diagram

Description automatically generated

**Figure: Context diagram for release 1.0 of the Lecturer Appointment Scheduling Application**

## User Classes and Characteristics

For students who want to get help from lecturers. The Lecturer Appointment Scheduling Application is a system that would help students easily search for their needs, make a slot booking, and ask questions before meeting. Unlike the present, when students do not know who to ask and how to find, our product could help students save their time, increase their choices and thereby achieve their goals.

For lecturers who need to choose an available slot. The Lecturer Appointment Scheduling Application is a system that would help lecturers spend their free time to make a schedule by create available slot with topic on the calendar, accept-deny booking requests, etc. Unlike the present, our product could help lecturers easily manage their free time, see the question before the meeting for better preparation.

For guests who want to experience the website. The Lecturer Appointment Scheduling Application is a system that would help guests sign up an account, search for lecturers before they sign in, etc. Unlike the present, our product would help guest save time easy to search

For admin who need to manage the system. The Lecturer Appointment Scheduling Application is a system that offer tools for management like ban an account, create new topic or edit it, etc. Unlike the present, our product would help admin easy to information management.

## Operating Environment

The application would be updated to the latest version. We have plans to develop products that could work on many different devices: IOS, iPad, Android by setting configuration changes that are more suited to a production environment. Besides, the application could extend to other universities.

## Design and Implementation Constraints

CO-2: The system shall use the current corporate standard SQL Server database

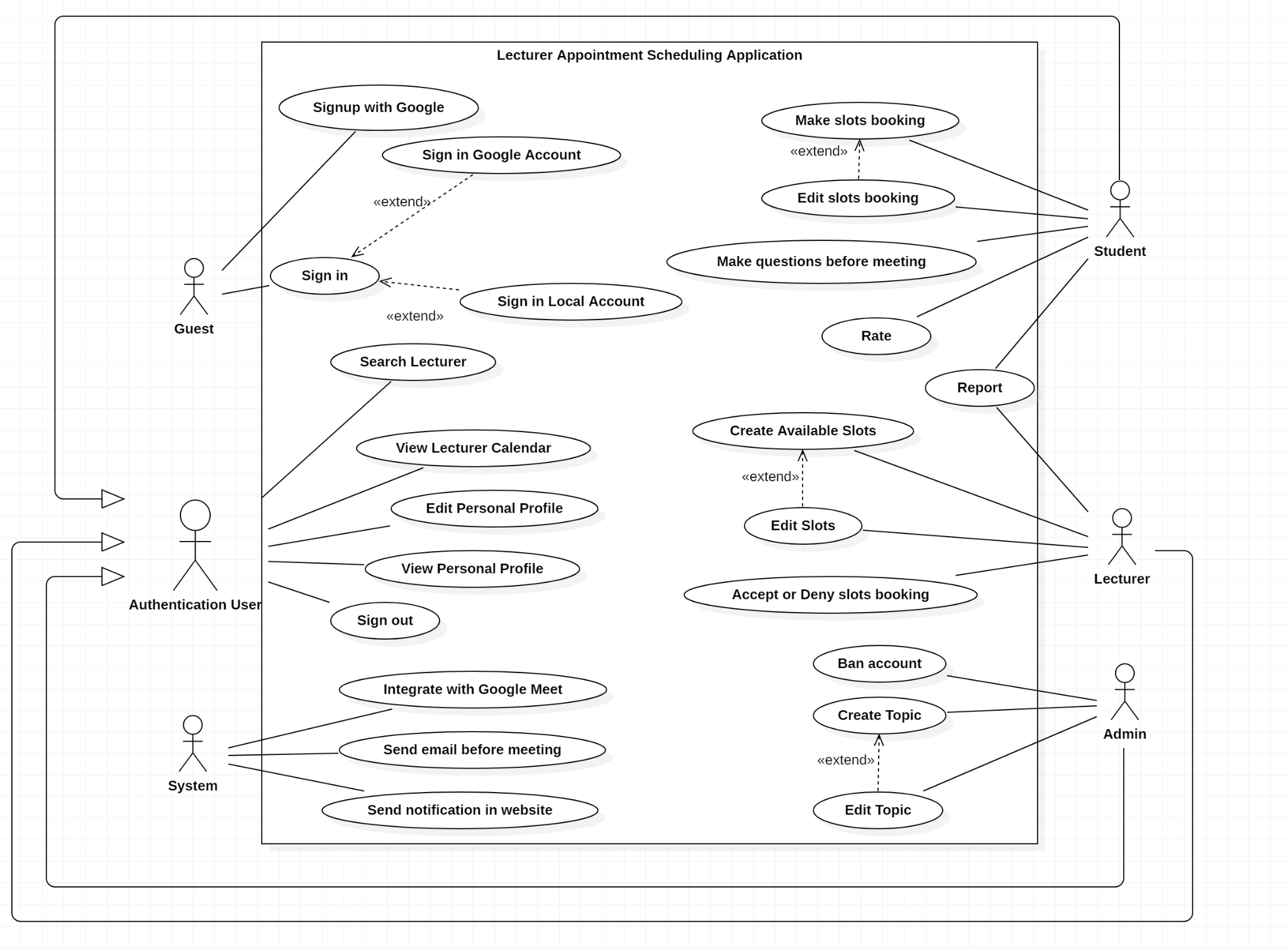
CO-2:  All HTML code shall conform to the HTML 5.0 standard

## Assumptions and Dependencies

There are no known assumptions or dependencies for this website.

# System Features

**3.1 Use case Diagram**



**Use Cases**

Sign in

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC - 1 Sign up** | | |
| Created By: | Trong | Date Created: | 20/6/2021 |
| Primary Actor: | Guest | Secondary Actors: | System |
| Description: | Guest visits the application through the website to search for some lecturers, then he/she wants to make a booking request, but the service needs sign-in before it can be used. This use case allows guests to sign in to the system. | | |
| Trigger: | A guest decides to sign in to use the service. | | |
| Preconditions: | PRE-1. The guest is not signed into the application. | | |
| Postcondition: | POST-1. The guest is an authenticated user, then the system displays the welcome page. | | |
| Normal Flow: | **1. Sign up**   1. The user goes to the sign-in page. 2. The system prompts the user for username and password or the “Sign-in with Google account” button. 3. The guest selects “Sign-in with Google account” as student or lecturer, or enters username and password if he/she is admin. 4. The system does validate. 5. Guest is signed in and returned to the service page as authenticated User. | | |
| Alternative Flows: | 4.1.This is the first-time guest signs in:  - The system will ask for permission to get her/his details for the sign-up process.  - Back to step 2 to resign-in. | | |
| Exceptions: | 4.1. Sign-in email is not an FPT email:  1. The system describes the email as an invalid email to sign in.  2. The system prompts the user to resign again in step 2.  4.2. Admin user enters an invalid username or password:  1. The system alerts the reasons why it failed.  2. The system prompts the user to resign again in step 2.  4.3. Email or account got ban:  1.The system alerts the account that it has got banned.  2. The system does not allow access, then back to step 2. | | |
| Priority: | High | | |
| Frequency of Use: | None. | | |
| Business Rules: | BR-1: Email used to sign-in as lecturer or student must be an FPT email.  BR-2: Any admin account must be a local account.  BR-3: User must sign up for the first time.  BR-4: A sign in session will expire after 30 days from the successful sign in. | | |
| Other Information: | None. | | |
| Assumptions: | None. | | |

Search Lecturer

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC - 2 Search lecturer** | | |
| Created By: | Trong | Date Created: | 20/6/2021 |
| Primary Actor: | Authenticated user | Secondary Actors: | System |
| Description: | A user has some problem with the lesson, he goes to the search page, enters the criteria for his search, then the system shows search results. | | |
| Trigger: | A user decides to search for a mentor. | | |
| Preconditions: | PRE-1. The user must be a signed-in user. | | |
| Postconditions: | POST-1. Search results show on the page.  POST-2. The user is a student, then he/she can make a booking request. | | |
| Normal Flow: | 1. The user selects the search service. 2. The system shows the search submission box. 3. The user enters criteria for search. 4. The system shows prompts for search criteria. 5. The user submits. 6. The system displays search results on-page. 7. The user selects one of these lecturers for more information. | | |
| Alternative Flows: | None. | | |
| Exceptions: | 2.1 Input is empty or all-whitespace characters:  1. The system will do nothing.  6.1.No lecturer fit the criteria:  1. The system describes “no search result”, then back to step 3. | | |
| Priority: | Medium | | |
| Frequency of Use: | On average, each student searches 3 times per day. | | |
| Business Rules: | BR-4: Only authenticated users can use the search service.  BR-5: The system does not allow searching for empty or all-whitespace values.  BR-6: Search results must be sorted in relation to search criteria from high to low. | | |
| Other Information: | None. | | |
| Assumptions: | None. | | |

Edit personal profile

|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC - 3 Edit personal profile** | | |
| Created By: | Trong | Date Created: | 20/6/2021 |
| Primary Actor: | Authenticated user | Secondary Actors: | System |
| Description: | A user accesses the profile page, then the system shows her/his details, he views and makes some updates. | | |
| Trigger: | A user wants to view or edit his/her personal profile. | | |
| Preconditions: | PRE-1. The user must be a signed-in user. | | |
| Postconditions: | POST-1. Updated details are applied. | | |
| Normal Flow: | 1. The system displays the user’s profile on-page. 2. The user views his/her profile. 3. The user selects the detail he/she wants to edit. 4. The system enables editing on the selected detail. 5. The user enters a new value for detail and submits. 6. The system validates the value. 7. The system renders new value to the view. 8. The user closes the edit. | | |
| Alternative Flows: | None | | |
| Exceptions: | 5.1 The user clicks the cancel button, back to step 2.  6.1 Invalid value:  1. The system displays the reason invalid.  2. The system asks the user to re-enter a new value in step 5.  6.2 New value is the same as the old value.  1. The system displays the reason.  2. Back to step 5. | | |
| Priority: | Medium | | |
| Frequency of Use: | Users usually change their avatar 1 per month. | | |
| Business Rules: | BR-7: The authenticated users can view and edit their personal profiles.  BR-8: The admin user can view but cannot edit any other user’s profile. | | |
| Other Information: | None. | | |
| Assumptions: | None. | | |

Make slots booking

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC – 6 Make slots booking** | | |
| Created By: | Nhung | Date Created: | 20/09/2021 |
| Primary Actor: | Student | Secondary Actors: | System |
| Trigger: | A student indicates that he/she wants to make a slot booking. | | |
| Description: | A student searches for lecturers by criteria, he/she selects an available slot on search results and makes a booking request to ask for help. | | |
| Preconditions: | PRE-1: The student did the search.  PRE-2: There are available slots in search results. | | |
| Postconditions: | POST-1 : The booking request with its information is sent to the slot. The system also sends notification to the lecturer. | | |
| Normal Flow: | **6. Make slots booking**   1. The student selects a lecturer on search results. 2. The system opens the lecturer schedule. 3. The student selects the “Make booking request” button on an available slot. 4. The system displays details of the slot and a form to create a booking request. 5. The student selects a topic, enters the questions, and clicks create. 6. The system displays a success message. | | |
| Alternative Flows: | 6.1 Creation of booking request is successful:   1. The system sends notification to the lecturer. | | |
| Exceptions: | 3.1 The student reached the limit on the number of booking requests:   1. The system alerts the reason. 2. Return to step 2.   6.1. Invalid input questions:   1. The system describes the invalid issue. 2. Back to step 5. | | |
| Priority: | High | | |
| Frequency of Use: | Students usually make slots booking. | | |
| Business Rules: | BR-15: The student can only make booking requests on the available slot.  BR-16: The number of booking requests of a student has a limit of 5.  BR-17: A booking request must be attached to one or more questions.  BR-18: A question has at least 20 characters and a limit of 1000 characters.  BR-19: If a student wants to cancel the meeting (accepted booking request), the student must give the reason. | | |
| Other Information: | None | | |
| Assumptions: | After choosing a slot, the application will load to the website normally.  There is no problem | | |

Rate lecturer after meeting

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC – 7 Rate lecturer after meeting** | | |
| Created By: | Anh | Date Created: | 20/09/2021 |
| Primary Actor: | Student | Secondary Actors: | System, lecturer |
| Trigger: | A student indicates that he wants to rate lecturer after meeting | | |
| Description: | A lecturer accepts a booking request from a student, then the system sends to them a link (google meet, zoom, etc.) to do a meeting. After the meeting, when the slot’s end time has expired, the student can give a vote for the lecturer. | | |
| Preconditions: | PRE-1: The lecturer accepted the student’s booking request.  PRE-2: The student completed the meeting with the lecturer. | | |
| Postconditions: | POST-1: The student’s votes are recorded for lecturers to improve search service. | | |
| Normal Flow: | **7. Rate lecturer after meeting**   1. After the meeting, the system sends a notification to the student for rating. 2. The student clicks on the notification. 3. The system shows a rating page. 4. The student makes a rating and submit. 5. System shows a message “Thank you for rating”, then back to the main page. | | |
| Alternative Flows: | 4.1. The student cancels the rating. | | |
| Exceptions: | 4.1. The student does not rate but press submit.   1. The system shows a warning message “you haven't rated yet”. 2. The system goes back to step 3. | | |
| Priority: | Low | | |
| Frequency of Use: | After the meeting, 50% of students do rating for lecturers. | | |
| Business Rules: | BR-20: The rating process only starts when the slot has a meeting, and its end time has expired.  BR-21: The rating is optional for the student. | | |
| Other Information: | None | | |
| Assumptions: | After voting, the application will load to the website normally.  There are no problem | | |

Ban account

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC – 8 Ban account** | | |
| Created By: | Anh | Date Created: | 20/09/2021 |
| Primary Actor: | Admin | Secondary Actors: | System |
| Trigger: | An admin indicates that he/she wants to ban account | | |
| Description: | There are some students who misbehave, violate rules, or get a report. An admin decides to ban the student’s account. | | |
| Preconditions: | PRE-1: The user signs in as admin. | | |
| Postconditions: | POST-1: After the admin bans an account, the system will send email notification to the student with reason.  POST-2: The student’s account is banned, cannot be signed in to the system. | | |
| Normal Flow: | **8. Ban account**   1. Admin goes to the dashboard page. 2. Admin searches the student by email or id. 3. The system shows the search result. 4. Admin clicks on the account he/she wants to ban. 5. The system shows the account information. 6. Admin clicks on the “Ban account” button. 7. The system changes the account status to banned, sign out the account, then reload the status of the account in the page. | | |
| Alternative Flows: | None | | |
| Exceptions: | 3.1 No account has found; the search result is empty:   1. The system displays on page “Your search does not match anything” 2. The system goes back to step 2. | | |
| Priority: | Medium | | |
| Frequency of Use: | The admin rarely bans accounts, because most students follow the rules. | | |
| Business Rules: | BR-22: Admin cannot ban another admin account.  BR-23: Admin cannot search admin account.  BR-24: Admin can search accounts for viewing profile or ban purposes. | | |
| Other Information: | None | | |
| Assumptions: | After banning an account, the application will load to the website normally. | | |

Create topics

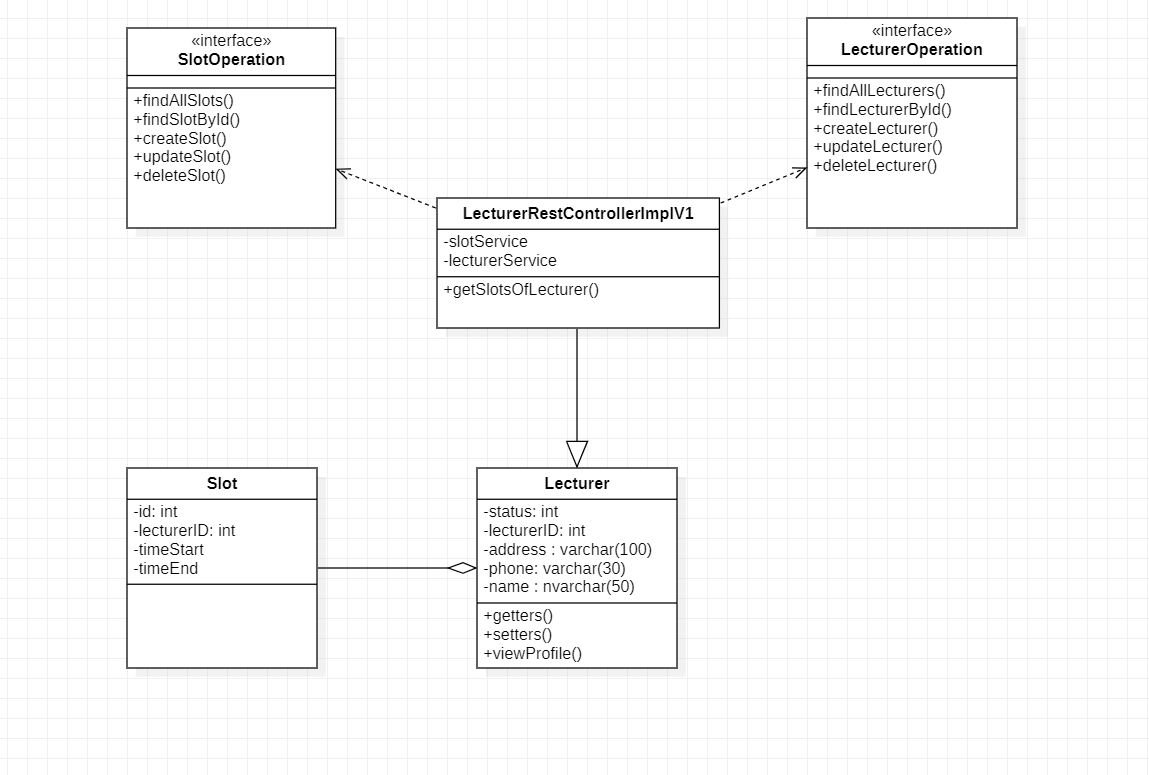
|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC – 9 Create topic** | | |
| Created By: | Anh | Date Created: | 20/09/2021 |
| Primary Actor: | Admin | Secondary Actors: | System |
| Trigger: | An admin indicates that he/she wants to create a new topic. | | |
| Description: | Admin login by account for the admin role. So admin can create a topic for lecturer | | |
| Preconditions: | PRE-1: The user signs in as admin. | | |
| Postconditions: | POST-1: The system publishes new topics to the topic list. | | |
| Normal Flow: | **9. Create topic**   1. The admin goes to the dashboard page. 2. The admin opens the topic list. 3. The system shows the topic dashboard. 4. The admin clicks on the “Add new topic” button. 5. The system shows a form to create a new topic. 6. The admin fills in the form and submits. 7. The system validates input. 8. The system adds a new topic to the topic list, then returns back to the topic dashboard. | | |
| Alternative Flows: | 6.1 The admin cancels the creation. | | |
| Exceptions: | 7.1 Invalid input   1. The system describes the reasons. 2. The system goes back to step 6. | | |
| Priority: | High | | |
| Frequency of Use: | Admin sometimes creates some new topics. | | |
| Business Rules: | BR-25: Topic name can be duplicate but not in the same subject or major.  BR-26: Only admin can add and edit topics. | | |
| Other Information: | None | | |
| Assumptions: | After creating a topic, the application will load to the website normally.  There are no problem | | |

Create available slots

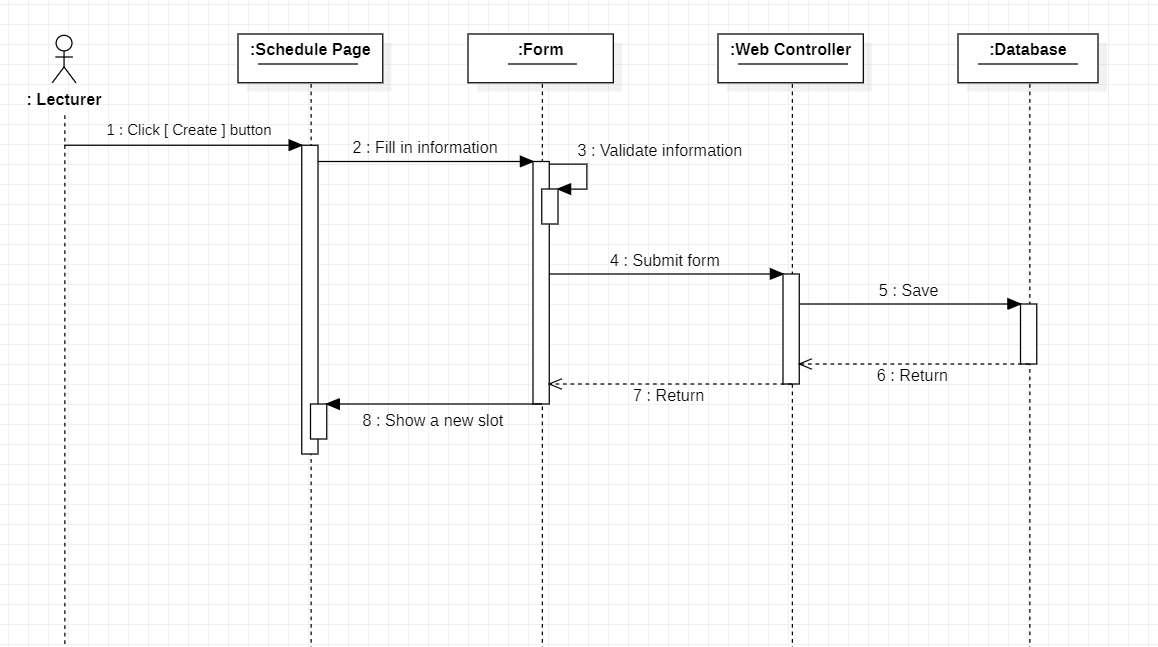
|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC - 4 Create available slots** | | |
| Created By: | Nhung | Date Created: | 20/09/2021 |
| Primary Actor: | Lecturer | Secondary Actors: | System |
| Trigger: | A lecturer indicates that he wants to create available slots | | |
| Description: | A lecturer accesses the system to add available slots, view the information, etc. | | |
| Preconditions: | PRE-1: Lecturer signed in by email FPT | | |
| Postconditions: | POST-1: An available slot is added to the lecturer schedule.  POST-2: The slot is public to the search service. | | |
| Normal Flow: | **4 Create available slots**   1. The lecturer clicks the “create a slot” button. 2. The system displays a form for the lecturer to create a slot. 3. The lecturer fills in information to complete the form, then clicks preview. 4. The system displays a preview for a new slot. 5. The lecturer clicks the submit button. 6. The system adds the slot to the lecturer schedule, returns to the schedule page, and displays a message: “Successfully created an available slot”. | | |
| Alternative Flows: | 3.1 The lecturer cancels the creation.  1. The system returns to the schedule page.  5.1 The lecturer clicks back to edit.  1. The system cancels the review page.  2. The system back to step 2, the input value is kept. | | |
| Exceptions: | 4.2. E1. Invalid input:       1. The system describes the reason.       2. Back to step 3.  4.2. E1. The system fails on adding a slot:       1. The system notifies the lecturer that an error has occurred       2. Return to the schedule page. | | |
| Priority: | High | | |
| Frequency of Use: | Lecturers usually create a slot to let students do the booking, on average it is 3 times per week. The time to load and add a slot into the website is 1 minutes. | | |
| Business Rules: | BR-9: The lecturer can create an unlimited number of slots, but the slots do not overlap in time.  BR-10: A slot must last at least 15 minutes or more.  BR-11: A slot can have one or more topics. | | |
| Other Information: | None | | |
| Assumptions: | After the lecturer creates a new slot, the application will load to the website normally. There is no problem. | | |

**3.2 System Feature 1 (Create available slot)**

CLASS DIAGRAM



SEQUENCE DIAGRAM

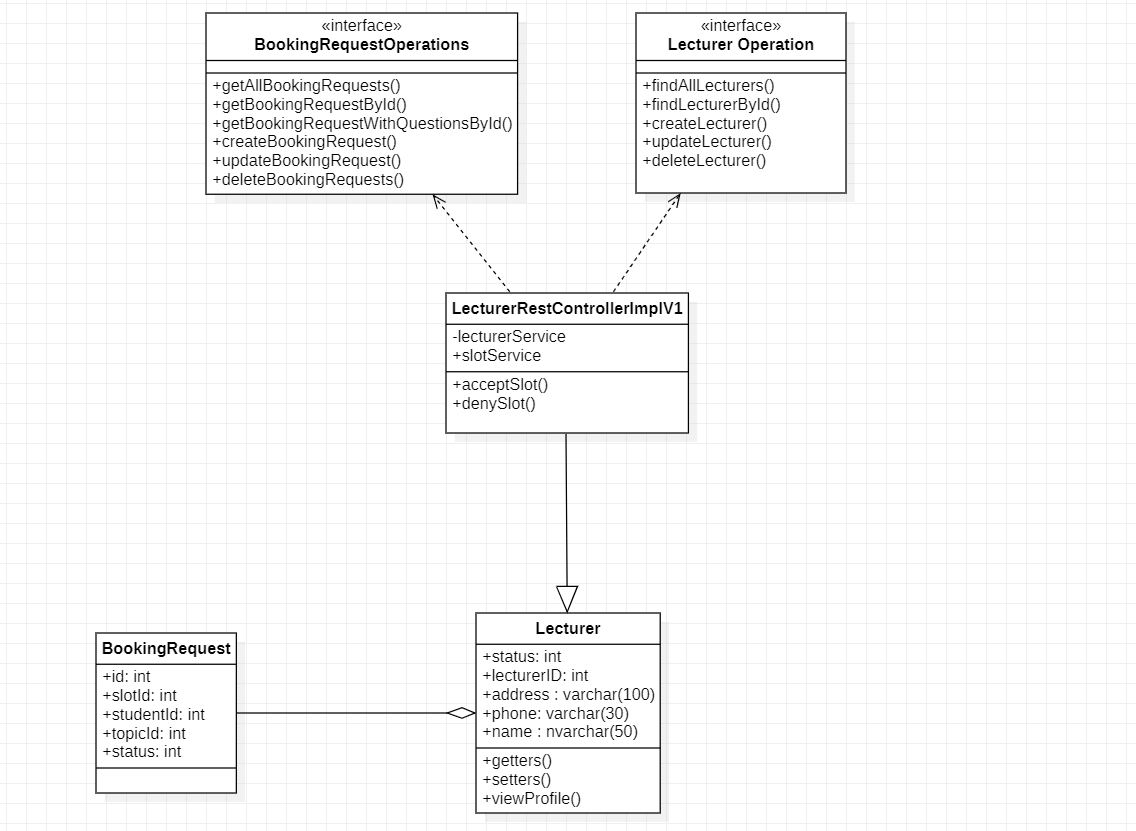


Accept or deny slots booking request

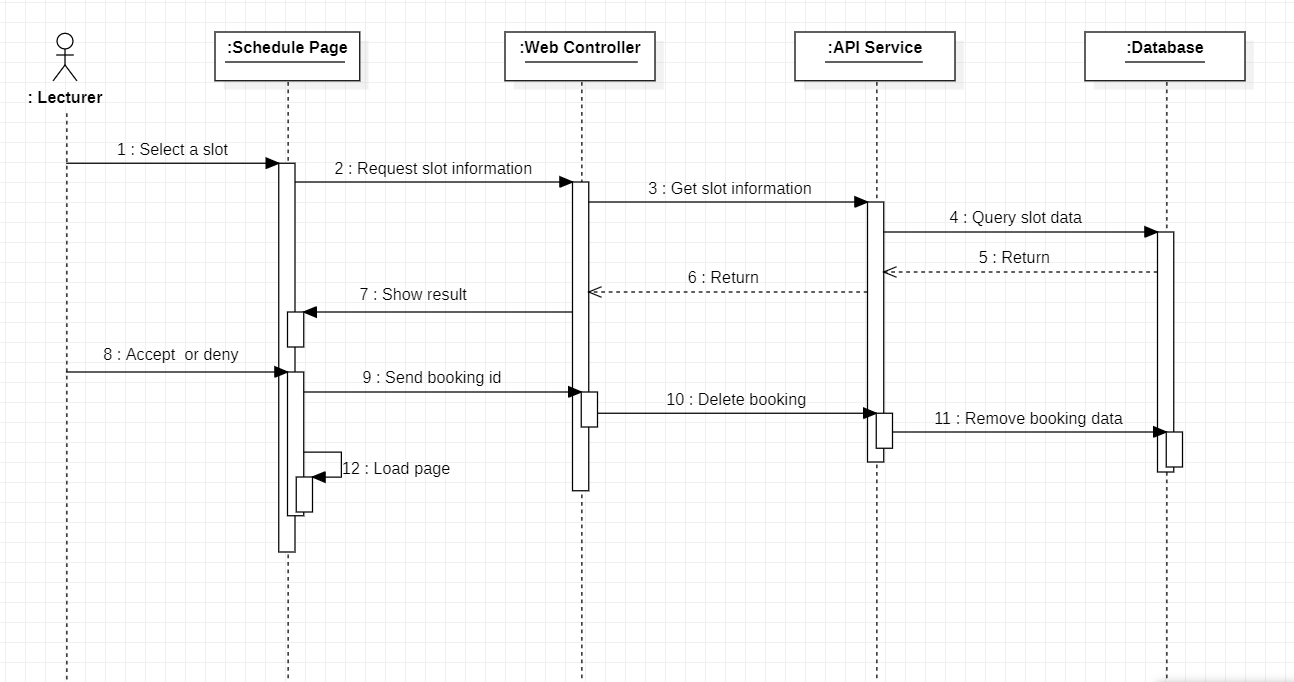
|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC - 5 Accept or deny slots booking request** | | |
| Created By: | Nhung | Date Created: | 20/09/2021 |
| Primary Actor: | Lecturer | Secondary Actors: | System |
| Trigger: | A lecturer indicates that he wants to accept or deny booking requests. | | |
| Description: | A lecturer creates some available slots, and it is public to the search service. Students can search and make a booking request to the slots. The lecturer can accept or deny these booking requests. | | |
| Preconditions: | PRE-1: The lecturer signed in to the system.  PRE-2: The lecturer has created some available slots. | | |
| Postconditions: | POST-1: The system sends a notification to the student whose booking request is accepted or denied.  POST-2: The system will send an email with the information about the meeting to both the student and lecturer. | | |
| Normal Flow: | **5. Accept or deny slots booking request**   1. The lecturer goes to his/her schedule page. 2. The lecturer selects a slot. 3. The system displays slot information and a list of booking requests. 4. The lecturer accepts or denies a booking request. 5. System sends notifications. | | |
| Alternative Flows: | 5.1 The lecturer denies a booking request   1. The system sends a notification to the owner of the booking request. 2. Back to step 3.   5.1 The lecturer accepts a booking request:   1. The system sends a notification on the website to both lecturer and student. 2. The system sends an email about the meeting to both lecturer and student 3. The system automatically denies all other booking requests on the slot and sends owners a notification. 4. The slot is unavailable for booking. | | |
| Exceptions: | * 1. The slot has expired:  1. The system displays “the slot has expired”. 2. Back to step 2.   4.1 System fails to accept or deny a booking request:   1. System notifies the lecturer that an error has occurred. 2. Return to the schedule page. | | |
| Priority: | High | | |
| Frequency of Use: | Every day, students usually choose a booking request . So, the lecturer also usually accepts or denies those requests. | | |
| Business Rules: | BR-12: A slot can receive many booking requests, but only one could be accepted.  BR-13: The system must send an email notification to the student and the lecturer before the meeting begins.  BR-14: If the slot’s start time is exceeded but no booking request is accepted, all booking requests must be denied. | | |
| Other Information: | None | | |
| Assumptions: | After the lecturer accepts or denies a booking request. The application will load to the website normally. | | |

**3.3 System Feature 2 (Accept or deny booking request)**

CLASS DIAGRAM

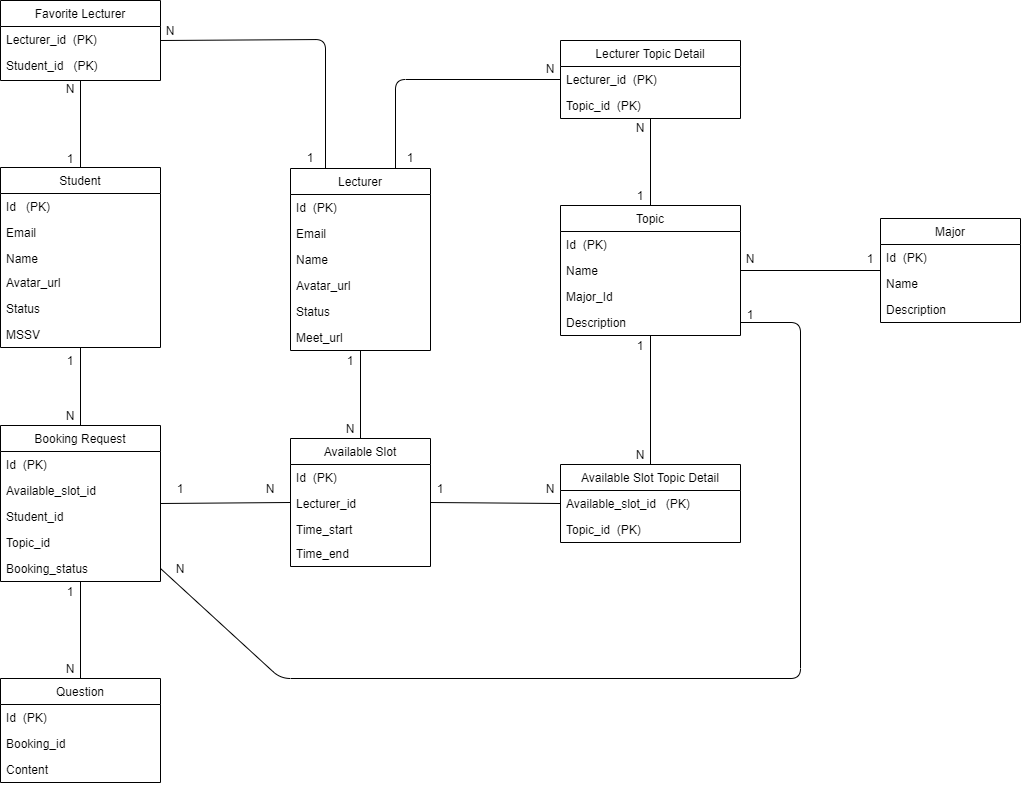


SEQUENCE DIAGRAM



# Data Requirements

## Logical Data Model



## Data Dictionary

**TABLES:**

**ADMIN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Composition or Data Type** | **Length** | **Values** |
| ID | ID of admin | Number | 6 | Positive number |
| Username | Username for sign in by LAS account | Alphanumeric | 20 |  |
| Password | Password for sign in by LAS account | Alphanumeric and special characters | 8-20 |  |
| Name | Name of admin | Alphabet | 50 | Capitalize first letter |

**STUDENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Composition or Data Type** | **Length** | **Values** |
| ID | ID of student | Number | 6 | Positive number |
| Email | Email for sign in or sign up | Alphabet, number, special characters |  | Local-part@domain |
| MSSV | Student code for each student | Alphabet and number | 8 | EX: SE\*\*\*\*\*\* |
| Name | Name in profile | Alphabet | 50 | Capitalize first letter |
| Phone | Phone of student | Number | 10-11 | Starts at 0 |
| Birthday | Date of birth in profile | Date  DD/MM/YYYY | 8 |  |
| Address | Address of student | Alphabet and number | 100 | Hyphens and commas permitted |

**LECTURER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Composition or Data Type** | **Length** | **Values** |
| ID | ID of lecturer | Number | 6 | Positive number |
| Email | Email for sign in or sign up | Alphabet, number, special characters |  | Local-part@domain |
| Name | Name in profile | Alphabet | 50 | Capitalize first letter |
| Phone | Phone number in profile | Integer | 10-11 | Starts at 0 |
| Birthday | Date of birth in profile | Date  DD/MM/YYYY | 8 |  |
| Address | Address of student in profile | Alphanumeric | 100 | Hyphens and commas permitted |
| MeetingURL | The link of lecturer to the meeting | Varchar(MAX) |  | http://........ |

**MAJOR**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Composition or Data Type** | **Length** | **Values** |
| ID | ID of each major | Alphabet | 2 | Capitalize |
| Name | Name of each major | Alphabet | 50 |  |
| Description | Description for each major | Alphabet | 100 |  |

**TOPIC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Composition or Data Type** | **Length** | **Values** |
| ID | ID of each topic | Number | 2 | Positive number |
| Name | Name of each topic | Alphabet | 50 |  |
| CourseID | ID for each course | Number | 10 |  |
| MajorID | ID of each major | Alphanumeric | 5 |  |

**SLOT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Composition or Data Type** | **Length** | **Values** |
| ID | ID of each slot | Number | 2 | Positive number |
| LecturerID | ID of each lecturer | Number | 6 | Positive number |
| TimeStart | Time start of each slot | DateTime | hh:mm | Ex: 00:00 |
| TimeEnd | Time end of each slot | DateTime | hh:mm | Ex: 00:00 |

**BOOKING REQUEST**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Composition or Data Type** | **Length** | **Values** |
| ID | ID of booking request | Number | 2 | Positive number |
| SlotID | ID of each slot | Number | 2 | Positive number |
| StudentID | ID of student | Number | 5 | Positive number |
| TopicID | ID of topic | Number | 2 |  |

**QUESTION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Composition or Data Type** | **Length** | **Values** |
| ID | ID of each question | Number | 2 | Positive number |
| BookingID | ID of booking requests | Number | 2 | Positive number |

**FAVORITE LECTURER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Composition or Data Type** | **Length** | **Values** |
| StudentID | ID of each student | Number | 2 | Positive number |
| LecturerID | ID of each lecturer | Number | 2 | Positive number |

**LECTURER TOPIC DETAIL**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Composition or Data Type** | **Length** | **Values** |
| LecturerID | ID of each lecturer | Number | 2 | Positive number |
| TopicID | ID of each topic | Number | 2 | Positive number |

## Reports

|  |  |
| --- | --- |
| Report ID: | LAS-RPT-1 |
| Report Title: | Appointment Report |
| Report Purpose: | Lecturer and Administrator could see the appointment report from the schedule after the meeting is done. The appointment report will be available about information relating to appointment |
| Priority: | Low |
| Report Users: | Lecturer and Administrator |
| Data Sources: | Appointment information and database of list student’s participating |
| Frequency and Disposition; | Report is generated on demand by the Lecturer and Administration. Data in the report is static. Report is displayed on the user's web browser screen on a computer, tablet, or smartphone. It can be printed if the display device permits printing. |
| Latency: | Complete report must be displayed to the lecturer within 10 seconds after it is requested. |
| Visual Layout: |  |
| Header and Footer: | Report header shall contain the report title and report footer shall show the page number. |
| Report Body: | * Date and time * Lecturer’s name * The number of student participating * History * Summary of appointment * Rating |
| End-of-Report Indicator: | None |
| Interactivity: | None |
| Security Access Restrictions: | Lecturer and Administrator can view the appointment report in appointment list |

## Data Integrity, Retention, and Disposal

DI-1: The LAS shall retain rating appointments for three years

DI-2: The LAS shall store lecturer and student information for several years

# 5. External Interface Requirements

# 5.1 User Interfaces

UI - 1: The user input correct credentials (username, password). User shall be redirected to landing page within 10 seconds

UI - 2: “Forgot Password” link shall be present below the “Sign up” button

UI - 3: Each page in the website shall have a menu bar with links to the major pages .When the user hover the mouse over menu options that shall change font, color, style…

UI - 4: The system shall display a suitable responsive website page on many different

devices: IOS, iPad, Android and on desktop.

UI – 5: Each page in the web site shall display the following copyright notice at the bottom: Copyright © 2021 LASA.

**5.2 Software Interfaces**

SI-1: The system shall validate the student, lecturer’s information when they create an account before inserting the account into the database.

SI-2: To save information student, lecturer we have chosen SQL database

SI - 2: The system shall send link Goggle meet when the lecturer accept a booking request

SI - 3: The system shall send email when the admin ban an account

**5.3 Hardware Interfaces**

No hardware interfaces have been identified

**5.4 Communication Interfaces**

CI-1: The system shall send notify to student when they forgot password

CI-2: The system shall connect the postman to retrieve the data of appointment

# 6. Quality Attributes

**6.1 Usability Requirements**

USE-1: The system will return search results (filter) after 10 seconds.

USE -2: The student will find a desired lecturer in less than 15 seconds.

USE-3: The system will send email when the student banned account by admin after 2 minutes

USE-4: The user will easy using website on different devices

**6.2 Performance Requirements**

PER-1: The web pages shall load an average response time of 5 seconds or less over a broadband Internet connection.

PER-2: Searching for a lecturer shall take at most 10s.

PER - 3: The system shall be available 24/7 for 364 days

**6.3 Security Requirements**

SEC-1: The access permission must only for FPT email

SEC-2: Only authenticated admin can ban account

SEC-3: The system manages sessions for a logged in user.

SEC-4: The system will check input validate with format (ex: validate date with format dd/mm/yyyy)

**6.4 Safety Requirements**

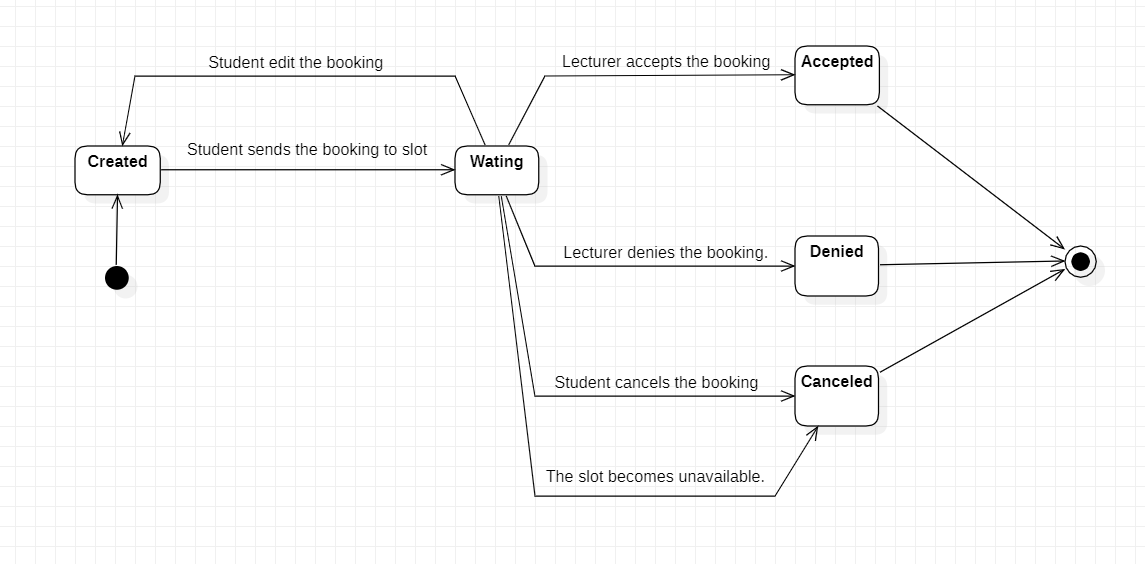
SE-1: The website shall use standard Web security protocols when transferring any private information regarding a student, lecturer.

**7. Internationalization and Localization Requirements**

**8.** **Other Requirements**

**Appendix B: Analysis Models**

**State Chart 1: Accept or deny booking request**

****

**State Chart 2**

