**Software Requirements Specification**

**for**

**<Course and Lab Allocation System>**

**Version 1.0 approved**

**Prepared by <Noor Fatimah(20l-0999) ,Rabiya Kamran( 20l-2097), Sameera Adnan (20l-1101)>**

**<FAST-NUCES>**

**<March 27,2023.>**

**Table of Contents**

**Revision History ii**

**1. Introduction** [**1**](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.32hioqz)

1.1 Purpose [1](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.1hmsyys)

1.2 Document Conventions [1](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.41mghml)

1.3 Intended Audience and Reading Suggestions [1](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.2grqrue)

1.4 Product Scope [1](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.vx1227)

1.5 References [1](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.3fwokq0)

**2. Overall Description** [**2**](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.1v1yuxt)

2.1 Product Perspective [2](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.4f1mdlm)

2.2 Product Functions [2](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.2u6wntf)

2.3 User Classes and Characteristics [2](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.19c6y18)

2.4 Operating Environment [2](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.3tbugp1)

2.5 Design and Implementation Constraints [2](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.28h4qwu)

2.6 User Documentation [2](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.nmf14n)

2.7 Assumptions and Dependencies [3](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.37m2jsg)

**3. External Interface Requirements** [**3**](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.1mrcu09)

3.1 User Interfaces 3

3.2 Hardware Interfaces [3](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.46r0co2)

3.3 Software Interfaces [3](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.2lwamvv)

3.4 Communications Interfaces [3](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.111kx3o)

**4. System Features** [**4**](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.3l18frh)

4.1 System Feature 1 [4](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.206ipza)

4.2 System Feature 2 (and so on) [4](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.4k668n3)

**5. Other Nonfunctional Requirements** [**4**](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.2zbgiuw)

5.1 Performance Requirements 4

5.2 Safety Requirements [5](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.1egqt2p)

5.3 Security Requirements [5](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.3ygebqi)

5.4 Software Quality Attributes [5](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.2dlolyb)

5.5 Business Rules [5](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.sqyw64)

**6. Other Requirements** [**5**](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.3cqmetx)

**Appendix A: Glossary** [**5**](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.1rvwp1q)

**Appendix B: Analysis Models** [**5**](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.4bvk7pj)

**Appendix C: To Be Determined List** [**6**](https://docs.google.com/document/d/1rovZC8BwrAelOASW9GRkrEdeP_z6V22G/edit#heading=h.2r0uhxc)

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Noor Fatimah,Rabiya Kamran, Sameera Adnan | **February 22, 2023.** | Initial Draft | 1.0 |
| Noor Fatimah,Rabiya Kamran, Sameera Adnan | **March 10, 2023.** | List of Requirements | 1.1 |

# **Introduction**

## **Purpose**

The product is the "Course and Lab Allocation System" for a university or educational institution. There is no specific revision or release number mentioned in the requirements.

The scope of this SRS covers the entire system, including various subsystems such as faculty member selection, lab committee selection, timetable committee functionality, HOD functionality, instructor preferences, course section planning, allocation of lab instructors, and hiring form for lab instructors.

Overall, this SRS describes a comprehensive software system for managing the allocation of courses and labs in an educational institution. It includes functionality for various stakeholders involved in the process, including faculty members, lab committees, timetable committees, and academic managers. The system is expected to be secure, expandable, and easy to use. The SRS covers all the major requirements and features of the system.

## **Document Conventions**

When writing the SRS document, there are several standards and typographical conventions that were followed to ensure that the document is clear, concise, and easy to understand. Some of these conventions include:

1. **Use of headings and subheadings:** To organize the document and make it easier to navigate, headings and subheadings are used to group related requirements together.
2. **Use of numbering and indentation**: Numbering and indentation is used to show the hierarchy of requirements and indicate which requirements are sub-requirements of others.
3. **Use of tables and diagrams:** Tables and diagrams were used to present complex information in a clear and concise way.
4. **Use of font styles and sizes**: Font styles and sizes were used to differentiate between different types of text, such as headings, requirements, and notes.
5. **Use of highlighting:** Bold or italicized text were used to emphasize important information or to draw attention to key points.
6. **Use of priority levels:** Every requirement statement has its own priority level, indicating its relative importance. The priority levels for higher-level requirements are not assumed to be inherited by detailed requirements, and each requirement is assigned its own priority level.
7. **Use of terminology:** A consistent terminology is used throughout the document, with each term defined clearly and unambiguously.

Overall, the SRS document follows a standard format and is written in clear, concise language, with a focus on describing the system requirements in detail while ensuring that the document is easy to read and understand.

## **Intended Audience and Reading Suggestions**

The SRS document is intended for various types of readers, such as:

1. **Developers** - The SRS document provides developers with a detailed understanding of the system requirements, including the functional and non-functional requirements, use cases, and system architecture. It can also help developers understand how the system will be tested and validated.
2. **Project Managers** - The SRS document provides project managers with a high-level overview of the project's goals, objectives, and scope. It can also provide information on the project's timeline, budget, and resource requirements.
3. **Marketing Staff** - The SRS document provides marketing staff with an overview of the system's features and capabilities. It can also provide information on how the system will benefit users and differentiate from other systems in the market.
4. **Users** - The SRS document provides users with a detailed understanding of the system's functionality, user interface, and usability requirements. It also provides information on how to use the system, including user manuals and training materials.
5. **Testers** - The SRS document provides testers with a detailed understanding of the system's requirements, including the testing approach, test scenarios, and acceptance criteria.

The SRS document typically contains the following sections:

1. **Introduction:** This section provides an overview of the SRS, its purpose, and its scope. It also includes information about the document's intended audience, revision history, and references.
2. **Overall Description:** This section provides a general description of the system and its requirements. It includes information about the system's goals, objectives, constraints, assumptions, and dependencies.
3. **External Interface Requirements:** This section describes the system's interfaces with other systems or external entities. It includes information about the hardware and software interfaces, communication protocols, data formats, and other relevant details.
4. **System Features:** This section provides a detailed description of the system's functional requirements. It includes a list of all the system features, their corresponding requirements, and acceptance criteria. This section is organized according to functional areas or subsystems.
5. **Other Non-Functional Requirements:** This section describes the system's non-functional requirements, such as performance, reliability, security, usability, and maintainability. It includes a list of all non-functional requirements, their corresponding requirements, and acceptance criteria.
6. **Appendices:** This section includes additional information that supports the SRS, such as glossary, use cases, diagrams, and other relevant documents.

To read the document, it is recommended to start with the overview sections, including the introduction and overall description. Then, depending on the reader's role, they can focus on the sections that are most pertinent to them. Developers may be interested in the system features section, while testers may want to focus on the external interface requirements and non-functional requirements sections. Project managers may want to review the overall description and all the requirements sections to ensure the project is on track.

## **Product Scope**

The Course and Lab Allocation System is a software system designed to facilitate the allocation of courses and labs to instructors in an academic institution. The system aims to streamline the allocation process and reduce the workload of the administrators and coordinators.

The main objective of the system is to ensure that each instructor is assigned to the appropriate course and lab based on their academic requirements and preferences. The system will also take into account the availability of instructors and facilities to ensure optimal allocation.

The benefits of the system include improved efficiency, reduced errors, and increased transparency in the allocation process. By automating the allocation process, the system will reduce the workload of administrators and coordinators, enabling them to focus on other important tasks.

The software is aligned with the corporate goals of the academic institution by enhancing the quality of education and improving the student experience. The system will help ensure that each instructor is assigned to the appropriate course and lab, reducing the likelihood of scheduling conflicts and ensuring that students can complete their academic requirements in a timely and efficient manner.

## **References**

*The* diagrams’ images are present in appendix A. Additionally, t*he reference links for diagrams’ files are as follows:*

***Use Case diagram:***

[***https://drive.google.com/file/d/1eMHYJLTgO9Nxhq5GqycFK-HIyzJr3ynP/view?usp=share\_link***](https://drive.google.com/file/d/1eMHYJLTgO9Nxhq5GqycFK-HIyzJr3ynP/view?usp=share_link)

***ER diagram:***

[***https://drive.google.com/file/d/1rxN4VCyZ2Ue9svsAWliWwcP4mnRFsRVm/view?usp=share\_link***](https://drive.google.com/file/d/1rxN4VCyZ2Ue9svsAWliWwcP4mnRFsRVm/view?usp=share_link)

***Class Diagram(Initial level):*** [*https://drive.google.com/file/d/1knTFn3xaMLeFVYZQ5KpZI84XbfX8uNPy/view?usp=share\_link*](https://drive.google.com/file/d/1knTFn3xaMLeFVYZQ5KpZI84XbfX8uNPy/view?usp=share_link)

# **Overall Description**

## **Product Perspective**

This system seems to be a brand-new, standalone product based on the specifications presented. It is made to make it easier to schedule classes and labs, assign instructors, and resolve clashes in timetables for the following semester.

Several stakeholders, including faculty members, the lab committee, the timetable committee, the HOD, academic managers of each department, and lab instructors, will use the system. Users will be able to browse and choose the courses or labs they want to teach or supervise, view schedules, and specify preferred time slots thanks to the system's straightforward interface and database orientation. Also, the system will aid in the selection of lab teachers..

The timetable committee, which will supply the initial edition of the timetable, will be a key source of information for the system.The approach will also permit the lab committee to divide up the teachers across the labs according to their preferences and availability, while maintaining a balance of seniority and permanency.

By eight weeks, the system should be 90% operational, and Python should be used for implementation. Secure data storage and transmission procedures are non-functional requirements for safeguarding private information and other sensitive data. The system should also be extensible and ready to accommodate an increasing number of users and courses as needed.

## **Product Functions**

* *Course registration*
* *Lab allocation*
* *Timetable management*
* *Venue/Room allocation*
* *User information management*
* *Notifications and Pop Ups*
* *Integration with other systems*
* *Security and privacy*
* *Scalability*

## **User Classes and Characteristics**

**Faculty Members:**

This user class must view and select the classes they wish to instruct in the following semester. Also, they must view the course and lab schedules they are teaching or supervising, which includes information on the times, venues, and enrollments.

**Lab Committee:**

This user class must view and decide which labs will be instructed by which instructors for the following term. In addition, lab instructors need to be allocated, ranked according to their grade and prior experience and according to their time availability and preferences. The allocation of lab venues and dealing with clashes in lab schedules is also the responsibility of the lab committee.

**Timetable Committee:**

This user class needs to interface with the lab committee and HOD functionalities. They must decide how many sections each course will have and look at all instructors' and departments' conflicting venue schedules.

**Lab Instructors:**

This user class needs to apply for lab instructor positions through a hiring form. Similar to the user class of faculty members ,they must view the lab schedules they are teaching which includes information on the lab times, lab venues, and student enrollments.

**Head of Department (HOD):**

This user class needs to decide the teachers for courses.

**Academic Manager:**

This user class needs to make the list of courses of that department and assign permanent and visiting instructors to each course.

The above mentioned user classes may have different levels of technical expertise, but they all require basic computer skills to use the system.

## **Operating Environment**

According to the requirements, the software system appears to be a web-based application that users can access via a web browser. Desktop computers, laptop computers, tablets, and mobile phones can all be used to access the system via the hardware platform. The system will require a web server to host the application and a database server to store data in order to function.

The web server's operating system can be any that supports the required web server software, such as Linux, Windows, or MacOS. Python is the primary programming language used to develop the system, but other technologies such as HTML, CSS, and JavaScript may be used for the user interface.

The system must coexist with other software components or applications used by the organization, such as web browsers, email clients, and web-based tools.

To replace the existing Excel-based system, non-functional requirements dictate that the system have a simple interface and be database-oriented. To meet the deadline, the system must be 90% functional in eight weeks, and the development team must plan project timelines and resource allocation accordingly.

## **Design and Implementation Constraints**

*There are some issues that will limit the options available to the developers as :*

**Corporate or regulatory policies:** The system's compliance with any corporate or governmental policies regarding data storage, privacy, and security will be the responsibility of the developers.

**Hardware limitations:**When selecting a hardware platform, developers must consider the system's timing and memory requirements.

**Interfaces to other applications:** The system may need to interface with other applications, such as email clients, web-based tools, or databases, which will require the developers to ensure compatibility and integration.

**Specific technologies, tools, and databases:** The system may need to use specific tools, or databases, such as Python, HTML, CSS, JavaScript, and database servers, to develop the user interface and store data.

**Security considerations:** To protect users' data, login passwords, private information, and other sensitive data, the developers must ensure that the system has secure data storage and transfer protocols.

**Timelines and resource allocation:** The developers will need to plan the project timelines and resource allocation accordingly to meet the 8-week deadline for the system to be 90% functional. This will require careful planning, prioritization, and efficient allocation of resources.

## **User Documentation**

**User manual:** A detailed document that explains how to use the software and its features. It could include examples and step-by-step instructions. It may also include the diagrams that can be used to visually understand the functionality.

**Online help or tutorials:** Contextual assistance is available from within the software. It could include details on specific features, troubleshooting advice, and frequently asked questions.

**Release notes:** Each software release includes a document that details the changes and improvements made like revision history file. It could contain information about new features, bug fixes, and known problems.

**Technical documentation:** Comprehensive details on the software architecture, data structures, and algorithms. It may be beneficial to software developers and IT administrators.This one is optional.

**Formats:** The user documentation delivery formats could include PDFs, videos, and interactive tutorials. The documentation should be written in clear and concise language, and follow any relevant industry standards or guidelines. It should also be easily accessible to users, preferably within the software itself.

## **Assumptions and Dependencies**

Assumed factors that could affect the requirements include:

* **Third-party or commercial components:** Third-party or commercial components, such as a database management system or authentication service, may need to be integrated into the system.

* **Development environment:**The development environment, which includes the programming language, development tools, and software, can have an impact on the system's implementation and performance.

* **Operating environment**: he operating environment, including the hardware and software infrastructure required to run the system, may have an impact on the system's performance and scalability.

* **Constraints:** The system may be subject to constraints such as budget, time, and resources, which could affect its development and implementation.Few were discussed in detail in section 2.5.

* **External dependencies:** External factors, such as software components from another project or APIs provided by external services, may be required by the system

* **Changes in requirements:** Changes in needs, regulatory requirements, or other factors may cause the requirements to evolve over time.

* **Availability of data:**The system may require data from multiple sources, such as course schedules, faculty availability, and lab assignments, which may require access from multiple departments or systems.

* **Privacy and security:** To protect sensitive data such as login passwords, private information, and other user data, the system must comply with privacy and security requirements.

* **Training and support:** The system may require training and support for faculty members, lab committee, and timetable committee, who may be unfamiliar with the new system.
* **Performance:** The system may need to handle large volumes of data and user requests, and hence performance testing may be necessary to ensure that the system can handle the expected load.
* **Testing:** The system may require testing to ensure that it meets the functional and non-functional requirements and to identify and fix any defects.

# **External Interface Requirements**

## **User Interfaces**

**User Login Interface:** It includes the fields that are used to enter the login credentials such as email and password. It should consist of logical characteristics such as validation of user credentials, session management and password reset functionality. It must consist of the link for instance of “forget password” that allows the user to reset it.

**User Registration Interface:** It includes the fields for the user so that user can enter here personal information such as name, email and password. It should have the logical characteristics of user input, unique email address requirements and password complexity rules. To submit the registration form, the interface should also include a button.

**Dashboard Interface:** It displays the overview of the user’s course and labs as well as for any other notification or messages. This interface may include the navigation link to other parts of the system. It provides us with the real time updates of course and lab schedules, easy navigation to other parts of it and graphical user interface.

**Course Catalog Interface:** It displays a list of available courses, with details such as course name, course schedule, description of courses and prerequisites. It should include the search option which would be achieved by filtering and sorting.

**Lab Catalog Interface:** It displays a list of available labs, with details such as lab name, lab schedule, lab description and prerequisites. It should include the search option to search a specific lab which would be achieved by filtering and sorting.

**Course Registration Interface:** It allows the user to search a course and enroll in it. It displays the details such as course name, schedule,description and prerequisites. It also provides feedback on the enrollment status.

**Lab Reservation Interface:** It should allow students to select and reserve the lab. It should display the details such as lab name, schedule, description and prerequisites. On the reservation status it also provides status to the users. It is used to validate the availability of lab and provide feedback on enrolment.

**Instructor Interface:** It allows the instructor to manage their courses and labs. It should include options to manage accounts, add and delete courses and labs and generate reports.

**Admin Interface:** It allows the administrator to manage the system. It includes the options such as managing user accounts, add and delete courses and labs and generate esports.

## **Hardware Interfaces**

**User Interface:** It follows GUI standards having layout which is optimized for usability and ease of navigation. The screen consists of buttons and functions, keyboard shortcuts are present and the error message displayed according to some standard rules. The interfaces should be responsive such that they adjust to the screen and orientation of the device. The interfaces of the system should be adaptable to be used in a variety of devices including desktop, laptops, tablets and smartphones.The interfaces can include large buttons and text.

**Data Input and Output:** It allows the user to enter the login credentials, course and lab selections and registrations. The output is displayed by the system having a screen which shows for instance the enrollment status of the student in a particular course and lab. The system should use reliable communication protocols i.e HTTP or TCP or IP. It should be designed in such a way that various hardware components can be used by it such as display screens or printers.

**Database Interface:** The system consists of the database interface for storing and retrieving course and lab information, as well as user data. It must allow the administrators to manage this information such as updating schedules, adding and removing courses and labs. The data storage and retrieval should be done in an optimized way. Care should be taken of data protection.

**System Administration Interface:** It is used to manage user accounts, manage the performance of the system and perform the administrative tasks. It must be easy to use and accessible in various device types. It should allow efficient use of resources available on devices.

## **Software Interfaces**

**Database Interface:** It is used to retrieve the data of user and course and lab information. It uses MySQL 8.0 for the storage and retrieval of data. Data items coming into the system are user data, information of courses and labs. Data items going out of the system are courses and lab schedules, status for enrollement and user information. Overall the services used by the interface is the storage and retrieval of databases. For the communication between the user and system, HTTP or TCP/IP protocols are used. The API protocols used by the interface are MySQL connector or python.

**Operating System Interface:** It is used to provide a stable and secure operating system of the course and lab allocation system to run. It uses Ubuntu version 20.04 LTS for running. In the operating system there are no data items coming in and going out of the system. This interface is used for the resource management of the system. For the communication with the system the API protocols such as POSIX API, Linux Standard Base can be used.

**Web Server Interface:** It is used to serve web pages and handle HTTP requests. It uses Apache HTTP v erosion 2.4. Data Items coming in the system are HTTP requests and going out is web pages and dynamic content. It is used to handle HTTP requests and serve the web pages. Communication with the system requires the use of HTTP, TCP/IP protocols. API protocols used are Apache modules.

**Email Notification Interface:** It is used to send emails and notifications to the users. It uses the Sendmail 8.15 version for the running interface. Data items coming in are the notifications of email and going out are the emails which are sent. SMTP,TCP or IP protocols are used for communication with the system. Sendmail is the API protocol used by the sytem.

**Authentication Interface:** It is used to authenticate users and manage the user accounts. It uses the Django Authentication system. The data items coming in are the login details of the user and data items going out are the authentication status. It uses HTTP or TCP/IP as modes of communication. The API protocol used by the system is Django Authentication.

## **Communications Interfaces**

**Emails:** It is used to send notifications to the users about the user login, the description of lab and course of students. The protocol which is used to send email is SMTP in which the HTML format is used followed by the CSS styling. The standard for message passing used here is Internet Message Format (IMF) and the security of the message is confirmed by encrypting it using SSL/TLS encryption.The rate of data transfer will be dependent on user preference and frequency of updates.

**Web Browser:** It is used to provide users with the interface which is web based so that course and lab schedules can be accessed and courses and labs can be enrolled. Communication with the web browser is done using HTTP following HTML, CSS and JavaScript format. HTML version 5 is used, CSS version 3 and JavaScript uses the standard ECMA Scrip6. The security of a web browser is ensured by encrypting it using SSL/ TLS encryption. The rate of transfer of a web browser depends on the user's request.

**Network Server Communication Protocols:** It is used to enable course and lab allocation systems for communication with network servers for instance database servers. For communication twitch the system TCP/IP is used as a communication protocol where the formatting of messages is in the binary format. Its encryption can be achieved using IPsec. Among various rates to transfer data, the data is transferred where the data transfer rate is maximum.

**Electronic Forms:**  It uses users to input data which include course and preferences and their enrollment information. It uses HTML as a communication protocol following the html format to pass messages. The version of HTML used by the system is HTML 5. CSS3 is also used for communication. It also achieves i’s security using SSL/TLS encryption.

# **System Features**

## **Feature: Course and lab selection**

### **4.1.1 Description and Priority**

Description: This feature allows faculty members to view and select the courses and labs they want to teach or supervise for the upcoming semester or academic year. This includes providing information on the courses and labs available, their schedules, and enrollment numbers, and allowing faculty to indicate their preferences and availability.

Priority: Medium-High

Benefit: This feature is essential for faculty members to have control over the courses and labs they will be teaching or supervising, which can lead to higher job satisfaction and performance. It also allows the department to better allocate resources based on faculty preferences and availability.

Penalty: If this feature is not implemented or poorly implemented, faculty members may not be able to teach the courses they want, which could lead to dissatisfaction and decreased performance.

Cost: The cost of implementing this feature is relatively low, as it mainly involves developing an interface for faculty members to view and select courses and labs.

Risk: The main risk associated with this feature is that some faculty members may not be able to teach the courses they want if there is high demand or limited availability, which could lead to dissatisfaction and decreased performance.

### 

### **4.1.2 Stimulus/Response Sequences**

**User**: Faculty member

1. User opens the system and logs in with their credentials.
2. User navigates to the course selection section.
3. Users view a list of available courses and labs for the upcoming semester or academic year.
4. User selects the desired courses and labs by checking the corresponding checkboxes.
5. User submits the selection form.
6. System stores the selected courses and labs for the user.

### **4.1.3 Functional Requirements**

1. Allow faculty members to view the list of available courses and labs for the upcoming semester or academic year.
2. Allow faculty members to select the courses and labs they want to teach or supervise.
3. Display the selected courses and labs to the faculty member and send a confirmation email.
4. Notify the timetable committee about the faculty member's selection.

## **Feature: Timetable planning**

### **4.2.1 Description and Priority**

Description: The Timetable planning feature is essential for managing the course schedules and allocating faculty members and resources efficiently. It involves creating a master timetable for the upcoming semester or academic year, considering various constraints such as faculty availability, course schedules, venues, and other resources. It also involves the allocation of instructors to each course and ensuring that no scheduling conflicts exist.

Priority: This feature is of high priority as it is critical for ensuring that courses and labs are scheduled efficiently and that faculty members are allocated to courses according to their availability and expertise. Efficient timetable planning can also help reduce scheduling conflicts and optimize the use of resources, resulting in improved academic outcomes.

Benefit: The benefits of this feature include improved efficiency in course scheduling, reduced scheduling conflicts, optimized use of resources, and improved academic outcomes.

Penalty: The penalty for not having this feature could result in scheduling conflicts, inefficient use of resources, and poor academic outcomes.

Cost: The cost of implementing this feature could vary depending on the complexity of the scheduling process and the level of automation required. However, the benefits of having an efficient timetable planning system could outweigh the cost in the long run.

Risk: The risk associated with this feature is the potential for errors in scheduling or conflicts arising from scheduling changes. However, with careful planning and the use of appropriate tools and techniques, such risks can be minimized.

### 

### **4.2.2 Stimulus/Response Sequences:**

**User:** Timetable committee member

1. User opens the system and logs in with their credentials.
2. User navigates to the timetable planning section.
3. User views the list of courses of their department.
4. User assigns the instructors to each course by selecting the instructors from the dropdown menu.
5. User submits the timetable for the department.
6. System stores the timetable for the department.

### **4.2.3 Functional Requirements:**

1. Provide an interface for the timetable committee to plan and allocate courses and labs based on the availability of faculty members, venues, and other resources.
2. Allow the academic manager of each department to make the list of courses of that department and assign instructors to each course.
3. Ensure that each course has at least one assigned instructor and that each instructor is assigned to at least one course.
4. Allow the timetable committee to set time slots for each course and lab.
5. Notify the faculty members about the schedule of the courses and labs they are teaching or supervising.

## 

## **Feature: Schedule viewing**

### **4.3.1 Description and Priority**

Description: This feature provides faculty members with an interface to view the schedules of the courses and labs they are teaching or supervising. The information includes details such as class times, venues, enrollment numbers, and preferences. This feature can help faculty members plan their work schedule and ensure they are available for their teaching responsibilities.

Priority: This feature can be considered a medium to high priority, as it provides important information to faculty members and can improve their work efficiency. The benefit rating can be around 7 or 8, as it can improve faculty members' work-life balance and help them manage their time effectively.

Cost: The cost rating can be around 4 or 5, as it may require some development effort to create a user-friendly interface.

Penalty: The penalty rating can be around 3 or 4, as the absence of this feature may not have severe consequences, but it may lead to some inconvenience for faculty members. The risk rating can be around 2 or 3, as this feature does not involve any significant risks or potential negative consequences.

### 

### **4.3.2 Stimulus/Response Sequences:**

**User**: Faculty member

1. User opens the system and logs in with their credentials.
2. User navigates to the schedule viewing section.
3. User views the schedule of the courses and labs they are teaching or supervising.
4. System displays details such as times, venues, enrollment numbers, and preferences.

### 

### **4.3.3 Functional Requirements:**

1. Allow faculty members to view the schedules of the courses and labs they are teaching or supervising.
2. Display details such as times, venues, enrollment numbers, and preferences.
3. Allow faculty members to request changes in their schedules.
4. Notify the timetable committee about the requested changes.

## **Feature: Conflict detection and resolution:**

### **4.4.1 Description and Priority**

Description: The system should allow the timetable committee to view the schedules of all instructors and venues of all departments having conflicts. The system should notify the lab and timetable committee if the changes led to clashes in preferences of instructors.

Priority: This feature is of High priority as it ensures that the scheduling process is efficient and conflict-free, thereby reducing disruptions and ensuring that classes and labs run smoothly.

Benefit: 8 - This feature will greatly benefit the timetable committee by providing them with a comprehensive overview of scheduling conflicts, allowing them to resolve issues before they occur.

Penalty: 6 - Failure to implement this feature could result in scheduling conflicts, which could lead to disruptions and inconvenience for faculty and students.

Cost: 7 - Developing this feature may require significant resources, such as time and personnel, to create an efficient and effective system.

Risk: 7 - There is a moderate risk of errors occurring during the implementation of this feature, which could result in further scheduling conflicts or errors. Additionally, faculty and staff may resist the change or encounter technical difficulties in adapting to the new system.

### **4.4.2 Stimulus/Response Sequences:**

**User**: Timetable committee member

1. User opens the system and logs in with their credentials.
2. User navigates to the conflict detection and resolution section.
3. Users view the schedules of all instructors and venues of all departments having conflicts.
4. System notifies the user if the changes led to clashes in preferences of instructors.
5. User resolves the conflict by making changes to the timetable.
6. System updates the timetable and notifies the affected users.

### **4.4.3 Functional Requirements:**

1. Allow the timetable committee to view the schedules of all instructors and venues of all departments having conflicts.
2. Notify the lab and timetable committee if the changes led to clashes in preferences of instructors.
3. Allow the timetable committee to resolve the conflicts by adjusting the schedules of the affected courses and labs.

## **Feature: Instructor preferences and ranking**

### **4.5.1 Description and Priority**

Description: Allow instructors to select up to 5 courses they want to teach and 2 slots they don't want to teach. The system will allow the instructor (courses and lab) to give the hard/soft preference of the time slots if any. The system should show the list of lab instructors in ranking order(grade and past experience) for allocation.

Priority: This feature is of Medium to High priority as it provides a more personalized approach for instructors and can lead to better allocation of courses and labs.

Benefit: The benefit of this feature is that it increases instructor satisfaction and engagement with the scheduling process, which can lead to higher quality teaching. Cost: There may be a higher cost in implementing this feature due to the need for additional functionality in the scheduling system.

Penalty: The penalty of not having this feature is that it may result in instructors being assigned courses and labs that do not align with their preferences or qualifications, which can lead to lower quality teaching and decreased job satisfaction.

Risk: The risk associated with this feature is that it may not be feasible to accommodate all of the instructor preferences, which could lead to further scheduling conflicts and challenges for the timetable committee.

### **4.5.2 Stimulus/Response Sequences**

**User**: Instructor

1. User opens the system and logs in with their credentials.
2. User navigates to the instructor preferences and ranking section.
3. Users select up to 5 courses they want to teach and 2 slots they don't want to teach.
4. User gives the hard/soft preference of the time slots if any.
5. System shows the list of lab instructors in ranking order(grade and past experience) for allocation.

### **4.5.3 Functional Requirements:**

1. Allow instructors to select up to 5 courses they want to teach and 2 slots they don't want to teach.
2. Allow instructors to specify hard/soft preferences for the time slots of the courses and labs.
3. Display the list of lab instructors in ranking order based on grade and past experience for allocation.
4. Notify the lab committee about the allocation of lab instructors.

## **Feature: Lab Instructor Allocation**

### 

### **4.6.1 Description and Priority**

Description: Allow the lab committee to add lab instructors as per time availability and preferences. The system should notify the lab committee if the allocation is not equal in a single lab i.e. each lab should have senior and junior rank based lab instructors and one of the two lab instructors should be permanent.

Priority: The feature is of high priority as the quality of lab instruction can be improved and it ensures the students receive the lab support they are in need of.

Benefit: It ensures that students are able to enroll in courses and labs so that they can complete their degree requirements on time. Hence there are more chances that students are able to complete their lab tasks on time.

Penalty: A poor design of the lab instructor allocation feature may lead to errors in assignments hence poorly impacting the knowledge and learning of students.

Cost: It requires initial investment of software and hardware infrastructure. Further it would require maintenance cost and the cost to train lab instructors on how to use the system.

Risk:There is a risk that the instructors would be having different points of views. The dissatisfaction among the instructors occurs because they all fail to reach a certain preference.

### **4.6.2 Stimulus/Response Sequences**

**User**: Lab instructor:

1. When lab instructors availability and qualification is provided by the system, it provides that with the lab instructor assignments.
2. When changes are made in the lab instructor’s availability and qualification is provided by the system, the system must be able to handle it and adjust the assignments accordingly.

### **4.6.3 Functional Requirements:**

1. Receive and process the availability and combination of lab instructors.
2. Generate the assignments of lab instructors based on the information provided by the lab instructor.
3. Allowance to lab instructors to view and confirm lab assignments.
4. Handle the changes made in the availability and qualification of lab instructors and accordingly make changes in the assignments.
5. The involvement of administrators and use of tools to handle the lab assignments.

## **Feature: Hiring form for Lab Instructors**

### 

### **4.7.1 Description and Priority**

Description:The system should have a hiring form for the lab instructors to apply. The hiring form should be simple and have basic questions like course/lab grade, CGA, availability, preferences. The system should store the record of lab instructors for future reference and hiring.

Priority: The priority of this feature is medium high where the qualified candidates are chosen to teach labs and it involves a lot of hiring process.

Benefit: It ensures that based on merit and in a standardized way all the candidates are chosen hence the administrators work to do effort manually in hiring the candidates. The system also reduces the errors.

Penalty: The poor design of the system may lead to error in the process of hiring and hence the negative impact would occur on the lab instructions quality.

Cost: The cost is required to hire the training managers so that they can use the system. The initial investment cost of software and hardware is also required along with the maintenance cost.

Risk: There is a risk that the system would not be able to fulfill all the requirements of hiring that may lead to a negative impact to the hiring managers and applicants.

### **4.7.2 Stimulus/Response Sequences**

**User**: Lab Instructors

1. Receive and process forms from applicants.
2. Evaluate the information based on the given requirements.
3. When changes are made to the hiring requirements, they must be handled by adjusting the evaluation form accordingly.

### **4.7.3 Functional Requirements:**

1. Receive and process hiring form from the applicants.
2. Evaluate the processed information and generate the evaluation form.
3. Allow the hiring managers to give the review to the evaluation form.
4. Generate reports for administrators on the data given by the applicants.
5. Providing communicating tools to the applicants to notify them about the hiring decisions.

## **Feature: User Authentication and data Security**

### 

### **4.8.1 Description and Priority**

Description: The system should have secure data storage and transfer protocols. Users' data, including login passwords, private information, and other sensitive data, should be secure and private on the system.

Priority: The priority of the system is high as it is the critical feature of the system which ensures the security of the system.

Benefit: The access of intruders or the access which is unauthorized prevents sensitive data, for example the personal information of the students and the instructors and the lab assignments which need to be evaluated by the instructors.Hence the privacy of students and instructors are protected which in turn provide a good image of the educational institute.

Penalty: Poor design of the system makes it have high chances of it to be hacked and would have many security threats.

Cost: Initial investment for software and hardware infrastructure is required along with the maintenance cost. The cost is required to train the users on how to use the system securely.

Risk: There is a risk that the system may not be able to fulfill all the security and authentication requirements that may lead to the security breaches.

### **4.8.2 Stimulus/Response Sequences**

**User**: Lab instructor

1. When provided with username and password, the system must be able to authenticate the user.
2. Ensures the users can only access the authorized data and functions.
3. It must be able to handle unauthorized access attempts, security threats, malware and many more.

### **4.8.3 Functional Requirements:**

1. Authenticate the users based on the login information provided by them.
2. Authorize users based on how much access permission they are given.
3. Encryption should be done of sensitive data to prevent unauthorized access.
4. Monitor the system activity to detect any security threats.
5. Communicate and notify the users of the security incidents.

## **Feature: Expandability and Scalability:**

### 

### **4.9.1 Description and Priority**

Description: The system should be expandable and capable of supporting an increasing number of users and courses as required.

Priority: It has high priority as the system requires to accommodate the changing needs of educational institutions.

Benefit: It can accommodate the changes and growth of the user base without needing to do a lot of modifications and less amount of time is required to make changes. It makes the system responsive and reliable even after it’s scalabiity.

Penalty: There might be possibility that the expandability and scalability of the system may be very costly and time consuming

Cost: Other than the initial hardware and software infrastructure which is required for investment, the system also requires the maintenance cost and the cost to update and expand the system as required.

Risk: There is a risk that the system design is not proper leading to the performance issue as the usage of system increases.

### **4.9.2 Stimulus/Response Sequences**

**User**: System

1. Handle a large number of users and data without having any impact on the performance or reliability of the system.
2. Must be able to do changings on the scalability and expandability of the system depending on the usage and need of data.

**4.9.3 Functional Requirements:**

1. Handle a large number of users and data.
2. Allow scalability depending on the usage and data requirements.
3. Ensures the system is reliable when the system grows.
4. Can be easily scalable and expandable

# **Other Nonfunctional Requirements**

## **Performance Requirements**

Non-functional performance requirements for the system include:

**Scalability:**

To accommodate an increasing number of users and courses, the system should be scalable. It should be able to handle increasing data loads without experiencing significant performance degradation. The system should also support concurrent multi-user access without sacrificing performance.

**Response time:**

The system should respond quickly, especially when displaying schedules and timetables. Any user action should be responded to within two seconds.

**Availability:**

The system should be available 24 hours a day, with less than 1% downtime for maintenance or upgrades.The features will be under HOD control and by his order other users' interfaces will get blocked. It must be capable of handling large amounts of data and users without crashing or slowing down.

**Security:**

The system should have strong security features to prevent unauthorized access, alteration, or loss of data. To protect sensitive data, the system should employ encryption and strong authentication mechanisms.

**Reliability:**

The system should be dependable, consistent, and error-free. It should be able to gracefully recover from failures and errors without data loss or corruption.

**Usability:**

The system should be simple to use, intuitive, and simple to navigate. It should provide users with clear and concise instructions on how to complete tasks.

**Compatibility:**

The system should work with a variety of operating systems and web browsers. It should also be able to integrate with the institution's other software systems.

**Performance under heavy load:**

Under heavy load, the system should perform optimally, especially during peak periods such as the registration period. It should be able to handle multiple user requests and respond quickly.

**Real-time performance:**

Real-time information, particularly for scheduling and timetables, should be displayed and updated by the system. To ensure that users have accurate and up-to-date information, the system should update schedules and timetables in real-time, with no noticeable delay.

**Data backup and recovery:**

The system should back up all data on a regular basis and include mechanisms for data recovery in the event of data loss or corruption.

**Maintainability:**

The system must be simple to maintain and upgrade. To make it easier for developers to understand and modify the code, the system should have proper documentation and code comments.

The non-functional performance requirements listed above are intended to ensure that the system is efficient, dependable, secure, and user-friendly for all users. It also emphasizes the importance of real-time and scalable performance in order to meet the growing demands of users. Real-time system timing relationships involve strict deadlines, and the system should respond quickly to all user requests within the specified time limits.

## **Safety Requirements and Security Requirements**

**Possible loss, damage or harm that could result from the use of the product:**

* Schedule conflicts or overloading may occur for instructors, resulting in burnout and decreased effectiveness.
* Students may experience scheduling conflicts or enrollment difficulties in desired courses or labs, resulting in frustration and delays in degree completion.
* The system may fail to allocate resources properly, resulting in inefficient use of faculty, lab spaces, and time slots, as well as potentially higher costs.
* The system may fail to handle sensitive information properly, such as instructors' and students' personal data, resulting in privacy violations and potential legal liability.

**Safeguards or actions that must be taken:**

* To prevent data entry errors and ensure the accuracy of scheduling information, the system must have robust error-checking and validation mechanisms.
* To prevent catastrophic errors or data loss, the system must have fail-safe mechanisms in place, such as regular backups and automatic save functions.
* To protect sensitive data, the system must use industry-standard security protocols and encryption methods.
* To avoid confusion and errors, the system must provide clear and user-friendly interfaces for all stakeholders.
* In the event of an emergency or unforeseen circumstance, the system must allow for manual override.

**Actions that must be prevented:**

* The system must protect sensitive data from unauthorized access and limit access privileges to those who have a need to know.
* The system must prevent conflicts of interest and ensure impartiality in course and lab allocation.
* The system must prevent instructors from being assigned to courses or workloads for which they are not qualified.
* The system must prevent instructors from being assigned to courses or labs that contradict their personal or religious beliefs or pose a safety risk.

**External policies or regulations that affect safety issues:**

* To ensure the privacy and security of personal data, the system must comply with applicable data protection and privacy regulations.
* To ensure the well-being of faculty and staff, the system must adhere to relevant labour laws and regulations, such as those governing work hours and compensation.
* To ensure the physical safety of instructors and students, the system must adhere to relevant safety regulations and guidelines, such as those pertaining to lab safety and accessibility.

**Safety certifications that must be satisfied**:

* To remain in compliance with industry standards and best practises, the system must be subjected to regular security audits and vulnerability testing.
* To demonstrate its commitment to security and data protection, the system must obtain relevant certifications.

## **Software Quality Attributes**

Additional quality characteristics that may be important to both customers and developers are:

* All users, including faculty, lab committee members, timetable committee members, and academic managers, should find the system simple to use and navigate. The system should be simple to use and require little to no training.
* The system should be reliable and have as little downtime as possible. It should be capable of handling heavy traffic and loads without crashing or slowing down.
* Unexpected inputs, errors, and exceptions should be handled gracefully by the system. It should be capable of recovering from errors without crashing or losing data.
* The system should be simple to maintain and update in the long run. It should be modular, with clearly defined concerns and interfaces.
* Unit tests, integration tests, and acceptance tests should all be performed on the system. The system should include a comprehensive test suite that can be run on a regular basis to ensure that everything works as it should.
* The system should be safe and secure, with user data protected from unauthorized access or theft. It should have safeguards in place to protect against hacking, phishing, and other forms of cybercrime.
* As the user base grows or shrinks, the system should be able to scale up or down. It must be capable of handling a large number of users and data without compromising performance.
* The system should be compatible with the institution's other systems and software. It should have well-defined APIs and data exchange standards.

The relative importance of these characteristics may differ depending on the stakeholders involved. For example, faculty members and academic managers may prioritize ease of use, whereas the IT department may prioritize reliability and security.

## **Business Rules**

Here are some possible operating principles that could apply to the product:

* Only authorized faculty members should be able to view and choose which courses to teach.
* Only authorized lab committee members should be able to view and choose which labs to supervise or teach.
* The timetable committee should have access to all functions associated with scheduling and resolving timetable conflicts.
* The HOD should have final say on which instructors are assigned to which courses.
* Instructors should be able to choose their preferred courses and time slots, as well as indicate which ones they do not wish to teach.
* When resolving scheduling conflicts, instructors' preferences should be considered.
* The lab committee should be in charge of assigning lab instructors based on availability and preferences, as well as ensuring that each lab has a senior and junior rank lab instructor, with one of the two lab instructors being permanent.
* Each department's academic manager should be in charge of creating a list of courses offered by the department and assigning instructors to those courses.
* The system should keep lab instructor records for future reference and hiring.

* If changes to the timetable cause clashes or conflicts with instructor preferences, the system should notify the appropriate parties.

# **Other Requirements**

Here are some additional requirements that could be considered for the project:

**Database Requirements:**

* The system should include a secure and dependable database to store all course, lab, instructor, and schedule information, as well as other pertinent data.
* The database should be scalable and capable of handling a large volume of data.
* In the event of data loss or system failure, the database should have appropriate backup and recovery mechanisms in place.

**Legal Requirements:**

In depth, legal requirements have already been discussed in section 5.2 i.e., safety and security requirements.

* The system must adhere to all applicable laws and regulations concerning data privacy, security, and accessibility.
* To prevent unauthorized access or data breaches, the system should ensure that all user data is secure and encrypted.
* To inform users about the data collection and usage policies, the system should provide appropriate terms of service and privacy policies.

**Reuse Objectives:**

* The system should be designed to be modular and extensible so that code components can be easily reused in future projects.
* To encourage community contributions and reuse, the system should be built using open standards and open-source software

**Project Management Requirements:**

* To ensure timely delivery and quality, the system development process should adhere to industry-standard project management methodologies such as Agile or Scrum.
* The project's scope, timeline, and budget should be clearly defined, with regular progress updates and status reports.
* The project team should have defined roles and responsibilities, as well as effective communication and collaboration tools.

**User Experience Requirements:**

* The system should have a user-friendly interface that is intuitive and simple to navigate for all users, regardless of technical proficiency.
* To guide users and assist them in recovering from errors, the system should provide appropriate feedback and error messages.
* The system should be designed to be responsive and work well on all devices and screen sizes, including mobile phones and tablets.

**Appendix A Analysis Models**



