

Software Requirement Specification

Course Evaluation System



Prepared by:

Muhammad Bilal 04072113022

Ali Ahmad Mumtaz 04072113023

Salman Ahmad 04072113050

Contents

1	Introduction	3
1.1	Purpose	3
1.1.1	Delineation of the Purpose	3
1.1.2	Intended Audience	3
1.2	Scope	3
1.2.1	Software Product(s) to be Produced	3
1.2.2	Functionality and Capabilities	4
1.2.3	Application and Objectives	4
1.2.4	Consistency with Higher-Level Specifications	4
1.3	Definitions, Acronyms, and Abbreviations	5
1.3.1	Terms	5
1.3.2	Acronyms and Abbreviations	5
1.4	References	5
1.4.1	List of Referenced Documents	5
1.4.2	Document Details	6
1.4.3	Source of References	6
1.5	Overview	6
1.5.1	Description of the SRS Contents	6
1.5.2	Organization of the SRS	6
2	Overall Description	7
2.1	Product Perspective	7
2.1.1	System Interfaces	8
2.1.2	User Interfaces	9
2.1.3	Hardware Interfaces	10
2.1.4	Software Interfaces	10
2.1.5	Communications Interfaces	11
2.1.6	Memory Constraints	11
2.1.7	Operations	11
2.1.8	Site Adaptation Requirements	11
2.2	Product Functions	11
2.2.1	Course Evaluation Functions	12
2.2.2	User Management Functions	12
2.2.3	System Administration Functions	12
2.2.4	Communication and Integration Functions	12
2.3	User Characteristics	13
2.3.1	Administrators	13
2.3.2	Instructors	13
2.3.3	Students	13
2.4	Constraints	13
2.5	Assumptions and Dependencies	14
2.6	Apportioning of Requirements	14

3	Specific Requirements	15
3.1	External Interfaces	15
3.1.1	Course Data Import	15
3.1.2	User Authentication	15
3.1.3	Data Export for Reporting:	16
3.1.4	User Notifications	16
3.1.5	Data Backup and Recovery	16
3.1.6	Data Security Configuration	17
3.1.7	System Notifications	17
3.2	Functional Requirements	18
3.2.1	Input Validation	18
3.2.2	Course Information Retrieval	18
3.2.3	Total Marks Entry	18
3.2.4	Instrument Definition	18
3.2.5	Mark Division Specification	18
3.2.6	Student Mark Entry	18
3.2.7	Grade and Position Calculation	19
3.2.8	Pass/Fail Reporting	19
3.2.9	Department-Level Deployment	19
3.2.10	University-Wide Data Access	19
3.3	Performance Requirements	19
3.4	Non-Functional Requirements	20
3.4.1	Usability	20
3.4.2	Scalability	20
3.4.3	Performance	20
3.4.4	Security	20
3.4.5	Reliability	20
3.4.6	Availability	20
3.4.7	Data Storage and Backup	21
3.4.8	Accessibility	21
3.4.9	Compliance	21
3.4.10	Error Handling:	21
4	Appendixes	21

1 Introduction

1.1 Purpose

1.1.1 Delineation of the Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive understanding of the Course Evaluation System, its functionalities, and the requirements for its development. This document serves as a detailed guide for software developers, designers, project managers, and stakeholders involved in the development and implementation of the Course Evaluation System.

1.1.2 Intended Audience

The intended audience for this SRS includes, but is not limited to:

1. **Software Developers:** Those responsible for designing, coding, and testing the Course Evaluation System.
2. **Project Managers:** Individuals overseeing the development process and project timelines.
3. **System Designers:** Professionals responsible for architecting the system's structure.
4. **Quality Assurance Teams:** Those conducting testing and quality assurance activities.
5. **Department of Computer Science Staff:** Faculty and staff who will utilize and administer the system.
6. **University Authorities:** Institutional decision-makers interested in the system's functionality and performance.
7. **Other Stakeholders:** Anyone with an interest in the successful development, deployment, and utilization of the Course Evaluation System.

1.2 Scope

1.2.1 Software Product(s) to be Produced

The software product to be produced is the "Course Evaluation System." This system is designed to support the Department of Computer Science at the university in managing the course evaluation process, tracking student performance, and providing valuable insights into course effectiveness.

1.2.2 Functionality and Capabilities

The Course Evaluation System will:

1. Allow course instructors, administrators, and students to access academic results.
2. Provide course instructors with the ability to manage course data, including defining major and minor instruments, percentage divisions, and entering total marks.
3. Enable the entry of student marks against each minor instrument.
4. Calculate and display grades and positions for each student based on the entered marks.
5. Determine pass/fail information for each student.
6. Allow the printing and saving of academic results for reference.

The Course Evaluation System will not:

1. Manage student enrollment in courses.
2. Handle financial transactions or payments.
3. Provide features unrelated to academic evaluation and course management.

1.2.3 Application and Objectives

The software's application is primarily within the Department of Computer Science, with the potential for university-wide deployment. The main objectives and goals of the Course Evaluation System are as follows:

1. Enhance the efficiency of course evaluation and academic result management.
2. Improve data accuracy and reduce manual data entry errors.
3. Provide timely access to academic results for instructors, students, and administrators.
4. Support data-driven decision-making for curriculum improvements and resource allocation.
5. Ensure compliance with accreditation and reporting requirements.

1.2.4 Consistency with Higher-Level Specifications

This SRS document aligns with higher-level specifications, including the system requirements specification. It is consistent with the broader objectives of enhancing the academic evaluation process and aligns with the university's goals for improved data management and decision support.

1.3 Definitions, Acronyms, and Abbreviations

This subsection provides definitions for terms, acronyms, and abbreviations used throughout the Software Requirements Specification (SRS) to ensure a clear and consistent interpretation of the document.

1.3.1 Terms

1. **Course Evaluation System:** The software product under development, designed to facilitate course evaluation, academic result management, and data analysis.
2. **Major Instrument:** A significant assessment component within a course, such as a final exam or a major project.
3. **Minor Instrument:** A sub-component of a major instrument, which further breaks down the evaluation of a course, such as quizzes, assignments, or presentations.
4. **Percentage Division:** The allocation of weightage or percentage value to different assessment components, helping calculate final grades.
5. **Academic Result:** The outcome of a student's performance in a course, including marks, grades, and positions.
6. **Pass/Fail Information:** Indicates whether a student has passed or failed a course based on predetermined criteria.
7. **Department of Computer Science:** The specific department within the university that serves as the initial target for the Course Evaluation System.

1.3.2 Acronyms and Abbreviations

1. **SRS:** Software Requirements Specification
2. **UI:** User Interface
3. **API:** Application Programming Interface
4. **PDF:** Portable Document Format
5. **XLS:** Microsoft Excel Spreadsheet Format

1.4 References

1.4.1 List of Referenced Documents

This subsection provides a list of documents referenced elsewhere in this Software Requirements Specification (SRS).

1. Systems and Software engineering — Life cycle processes — Project Management ISO/IEC/IEEE 16326:2019(E) Second edition 2019-12
2. IEEE Recommended Practice for Software Requirements Specification - IEEE Std 830-1998 (Revision of IEEE Std 830-1993)

1.4.2 Document Details

Below are the details for the referenced documents:

1. **Title:** Systems and Software engineering — Life cycle processes — Project Management ISO/IEC/IEEE 16326:2019(E) Second edition 2019-12
Report Number: N/A
Date: December 2019
Publishing Organization: ISO/IEC/IEEE
2. **Title:** IEEE Recommended Practice for Software Requirements Specification - IEEE Std 830-1998 (Revision of IEEE Std 830-1993)
Report Number: IEEE Std 830-1998
Date: 1998
Publishing Organization: IEEE (Institute of Electrical and Electronics Engineers)

1.4.3 Source of References

The referenced documents are available from the respective publishing organizations and can be obtained through official channels or online resources. These documents are valuable references to ensure compliance with established standards and practices in software engineering and requirements specification.

1.5 Overview

1.5.1 Description of the SRS Contents

The Software Requirements Specification (SRS) provides a detailed insight into the Course Evaluation System, its functionalities, and the requirements necessary for its development and implementation. This document outlines the objectives and scope of the system, details its features and functionalities, and specifies the requirements that need to be fulfilled. It is intended to serve as a comprehensive reference for software developers, designers, project managers, and stakeholders involved in the development process.

1.5.2 Organization of the SRS

The SRS is organized into the following sections:

1. **Introduction:** Provides an overview of the purpose, scope, definitions, acronyms, and abbreviations used in the document.

2. **Overall Description:** Details the product perspective, functions, user characteristics, constraints, assumptions, and dependencies.
3. **Specific Requirements:** Specifies the detailed requirements that the Course Evaluation System must satisfy.
4. **Appendixes:** Contains additional information, such as diagrams, charts, and supporting documentation.
5. **Index:** Provides an index for easy reference.

The SRS is structured to ensure clarity and ease of navigation, allowing readers to understand, evaluate, and implement the requirements for the Course Evaluation System effectively.

2 Overall Description

2.1 Product Perspective

The Course Evaluation System, as defined in this SRS, is an independent software product primarily designed for the Department of Computer Science. However, it is a component of the larger university's information management ecosystem, which includes the following related systems:

1. **Course Allocation System (Case Study #1):** The Course Allocation System is responsible for allocating courses to students. The Course Evaluation System needs to interface with this system to provide pass/fail information for students.
2. **Course Management System (Case Study #3):** The Course Management System manages course-related data and information. While not explicitly mentioned in the provided case study, the Course Evaluation System may need to access course data from this system.
3. **Other Departmental Systems:** In the future, when the Course Evaluation System is deployed across all departments of the university, it will interact with similar systems in other departments, allowing the university to obtain information about students from all departments.
4. **University Database:** The Course Evaluation System may need to connect to a university-wide database to retrieve or store information about students, courses, and evaluations. This database will serve as an external interface.

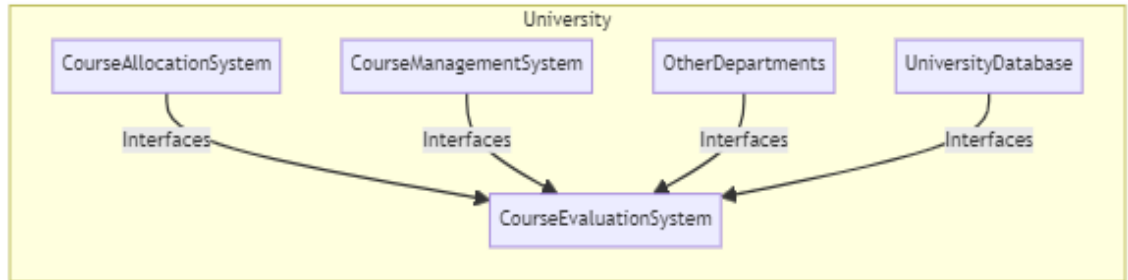


Figure 1: Block Diagram: Course Evaluation System Integration

2.1.1 System Interfaces

The Course Evaluation System interfaces with the following systems to accomplish specific requirements and functionality:

1. Course Allocation System (Case Study #1)

Functionality: The Course Evaluation System interfaces with the Course Allocation System to provide pass/fail information for students. This interface ensures that student evaluation results are communicated to the Course Allocation System for appropriate course allocation.

Interface Description: The interface with the Course Allocation System includes the exchange of pass/fail status for each student. It may involve data transmission through a secure and reliable API or integration point.

2. Course Management System (Case Study #3)

Functionality: The Course Evaluation System may interface with the Course Management System to access course-related data. This interface may involve retrieving course details, instructor information, or other relevant data.

Interface Description: The specific interface details, such as data exchange formats or API endpoints, will be determined based on system requirements and discussions with stakeholders.

3. Other Departmental Systems

Functionality: In the future, when the Course Evaluation System is deployed across all departments, it will interface with similar systems in other departments. These interfaces will allow the university to obtain information about students from all departments, enabling comprehensive student data management.

Interface Description: The interfaces with other departmental systems will be designed to accommodate data sharing, student information transfer, and cross-departmental data access.

4. University Database

Functionality: The Course Evaluation System may need to interface with a university-wide database to retrieve or store information about students, courses, and evaluations. This database serves as a central data repository for the university's information management ecosystem.

Interface Description: The interface with the university database may involve SQL queries, data synchronization protocols, or other methods for accessing and updating relevant data.

2.1.2 User Interfaces

This section specifies the logical characteristics and optimization aspects of the user interfaces in the Course Evaluation System. The user interfaces are designed to facilitate interaction between the software product and its users, ensuring efficient and user-friendly access to system functionalities.

1. Logical Characteristics of Interfaces

User Authentication: The system will provide a login interface with fields for user identification (ID) and password.

Dashboard: After login, users will be presented with a dashboard displaying relevant information and options.

Navigation Menu: A navigation menu will allow users to access various system features, such as viewing results, managing courses, and printing/saving results.

Course Selection: Instructors and students can select specific courses for which they want to view results.

Course Management: Instructors and administrators will have interfaces to define major and minor instruments, percentage divisions, and enter total marks for courses.

Result Display: Users will see academic results, including marks, grades, and positions, presented in a clear and organized manner.

Printing/Exporting: An option to print or export results in PDF or XLS format for reference.

Error Messages: User-friendly error messages will provide clear feedback for any input or system errors.

2. Optimizing the User Interface

Customizability: Users may have options to customize the appearance of their interface, such as choosing a light or dark theme or arranging dashboard widgets based on personal preferences.

Keyboard Shortcuts: To optimize user interaction, the system may provide keyboard shortcuts for common tasks, allowing power users to perform actions quickly.

2.1.3 Hardware Interfaces

The Course Evaluation System does not require specific hardware interfaces or configurations. It is designed to be platform-agnostic and accessible from standard personal computers, laptops, and mobile devices using common web browsers. It adapts to various screen sizes and operates on multiple operating systems.

The system does not rely on specialized hardware, ports, or protocols and is accessible via typical web interfaces. It ensures ease of access and usability on a wide range of devices without the need for unique hardware configurations or specifications.

2.1.4 Software Interfaces

The Course Evaluation System relies on various software products and may have interfaces with other application systems. These include:

1. Database Management System (DBMS):

- **Name:** The system supports multiple DBMS, including MySQL, PostgreSQL, and Microsoft SQL Server.
- **Mnemonic:** N/A
- **Specification Number:** N/A
- **Version Number:** Versions compatible with the system's data access libraries.
- **Source:** Open-source or licensed versions.

Interface Purpose: The Course Evaluation System interfaces with the DBMS to store and retrieve data related to courses, students, evaluations, and user accounts.

Interface Definition: The interface with the DBMS includes SQL queries for data retrieval and modification. Specific query content and format may vary but should follow standard SQL syntax. No detailed interface specification is provided here; instead, reference is made to the respective DBMS documentation.

2. Web Server Software:

- **Name:** Apache, Nginx, or similar web server software.
- **Mnemonic:** N/A
- **Specification Number:** N/A
- **Version Number:** Versions compatible with the web application framework.
- **Source:** Open-source or licensed versions.

Interface Purpose: The web server software is responsible for hosting and serving the Course Evaluation System to users over the internet.

Interface Definition: The interface involves the configuration of the web server to host the web application and handle incoming HTTP/HTTPS requests. It should include information on virtual hosts, server settings, and request routing. Specific configurations may vary depending on the chosen web server software.

2.1.5 Communications Interfaces

The Course Evaluation System uses communication interfaces for local network protocols, including standard internet protocols like HTTP and HTTPS for web-based communication within the local network. The system ensures secure and efficient communication, with specific protocols and encryption defined during deployment.

2.1.6 Memory Constraints

The system adheres to memory constraints:

- **Primary Memory:** Efficient operation within available RAM.
- **Secondary Memory:** Effective data storage and retrieval on secondary devices. Specific limits determined during deployment.

2.1.7 Operations

The Course Evaluation System supports various modes of operation, including user-initiated tasks, automated data processing, and backup and recovery. It operates interactively and unattended, with data processing functions, backup, and recovery operations.

2.1.8 Site Adaptation Requirements

The system can be adapted to specific installations, allowing customization of features, user roles, and data storage locations. Site-specific data or initialization sequences can be defined to align with the unique needs of each installation or user organization.

2.2 Product Functions

The Course Evaluation System performs a range of key functions, ensuring efficient and comprehensive course evaluation processes. These functions are organized to provide clarity and an understanding of the software's capabilities. A graphical representation of these functions is also available.

2.2.1 Course Evaluation Functions

1. **Course Data Management:** Allows course details to be added, updated, and retrieved.
2. **Instrument Definition:** Permits the creation and configuration of major and minor evaluation instruments.
3. **Mark Entry:** Enables entry of student marks, including support for both major and minor instruments.
4. **Grade Calculation:** Automatically computes student grades based on defined evaluation criteria.
5. **Position Determination:** Determines student positions within a course according to their performance.
6. **Pass/Fail Evaluation:** Evaluates student performance and marks students as pass or fail.
7. **Reporting:** Generates reports, including grade sheets, position lists, and overall course evaluations.

2.2.2 User Management Functions

1. **User Authentication:** Manages user login and authentication processes.
2. **Role-Based Access Control:** Ensures that users have appropriate access based on their roles (admin, instructor, student).
3. **User Profile Management:** Allows users to update and manage their profiles.

2.2.3 System Administration Functions

1. **System Configuration:** Provides administrators with tools to configure system settings.
2. **Data Backup and Recovery:** Supports data backup and recovery operations for data integrity.

2.2.4 Communication and Integration Functions

1. **Data Exchange with Course Management System:** Interfaces with the Course Management System for course-related data.
2. **Data Synchronization with University Database:** Maintains data consistency with the University Database.

2.3 User Characteristics

2.3.1 Administrators

- **Educational Level:** Administrators typically hold degrees in related fields, such as education, computer science, or information management.
- **Experience:** They have experience in managing educational software systems and administrative tasks.
- **Technical Expertise:** Administrators possess a good understanding of software configuration and system administration.

2.3.2 Instructors

- **Educational Level:** Instructors are typically educators with academic qualifications related to the subjects they teach.
- **Experience:** They have experience in teaching and evaluating students.
- **Technical Expertise:** Instructors may have varying degrees of technical expertise, with some being highly proficient in educational technology, while others may require more user-friendly interfaces.

2.3.3 Students

- **Educational Level:** Students are enrolled in various academic programs, spanning from undergraduate to postgraduate levels.
- **Experience:** They have experience with academic courses and grading processes.
- **Technical Expertise:** Student users may have different levels of technical proficiency, with some being comfortable with web-based tools and others requiring intuitive interfaces.

2.4 Constraints

The Course Evaluation System operates within a framework of constraints and limitations that influence its development and functionality:

- **Regulatory Compliance:** The system must adhere to relevant regulatory policies.
- **Hardware Limits:** Compliance with hardware limitations, including signal timing, is essential.
- **Interfaces:** Interfaces with other applications must align with their protocols.
- **Parallel Operation:** Supporting concurrent user operations is crucial.

- **Audit and Control:** Robust audit and control functions are necessary.
- **Language Compatibility:** Code should meet higher-order language requirements.
- **Signal Handshake:** Implement signal handshake protocols as needed.
- **Reliability:** Ensure system reliability, especially during evaluations.
- **Criticality:** The system is critical for academic assessment.
- **Safety and Security:** Stringent safety and security measures protect academic data.

2.5 Assumptions and Dependencies

The Course Evaluation System relies on key assumptions and external dependencies that can impact its functionality and performance:

- **Hardware Availability:** Assumes the availability of designated hardware.
- **Operating System Compatibility:** Assumes compatibility with the specified OS.
- **Network Infrastructure:** Assumes a stable network for data exchange.
- **Data Availability:** Assumes the availability of accurate data.
- **Stakeholder Cooperation:** Assumes active stakeholder engagement.
- **Timely Maintenance:** Assumes regular system maintenance.
- **External Integration:** Depends on cooperation with external applications.
- **Change Management:** Assumes the ability to adapt to policy changes.
- **Data Security:** Assumes robust data security measures.
- **Documentation:** Assumes the availability of up-to-date system documentation.
- **User Training:** Assumes users receive necessary training for system usage.

2.6 Apportioning of Requirements

Certain requirements, such as advanced reporting features, extended integrations, and enhanced security measures, may be deferred for future system versions to meet immediate project timelines and priorities. These deferred requirements will be considered for implementation in subsequent system updates.

3 Specific Requirements

3.1 External Interfaces

This section provides a detailed description of all inputs into and outputs from the Course Evaluation System. Each interface is defined with the following attributes:

3.1.1 Course Data Import

- **Purpose:** To import course data from the Course Management System.
- **Source:** Course Management System (Case Study 3).
- **Valid Range:** All courses offered in the current semester.
- **Units of Measure:** N/A
- **Timing:** Scheduled weekly data synchronization.
- **Relationships:** Course data is linked to student evaluations.
- **Data Format:** XML or CSV format.
- **Command Format:** Automated data import commands.
- **End Messages:** Confirmation of successful data import.

3.1.2 User Authentication

- **Purpose:** To authenticate users before system access.
- **Source:** User login screen.
- **Valid Range:** Valid user credentials.
- **Units of Measure:** N/A
- **Timing:** User-initiated login.
- **Relationships:** Authentication is the initial step for all system operations.
- **Data Format:** User ID and password.
- **Command Format:** Secure login procedure.
- **End Messages:** Login success or failure messages.

3.1.3 Data Export for Reporting:

- **Purpose:** To export evaluation data for reporting.
- **Destination:** Reporting and analysis tools.
- **Valid Range:** Specific evaluation data sets.
- **Units of Measure:** N/A
- **Timing:** User-initiated data export.
- **Relationships:** Exported data is used for generating reports.
- **Data Format:** Various formats, including PDF, Excel, and CSV.
- **Command Format:** User-defined report parameters.
- **End Messages:** Confirmation of successful data export.

3.1.4 User Notifications

- **Purpose:** To send notifications to users.
- **Destination:** User interfaces and email.
- **Valid Range:** Various notification types.
- **Units of Measure:** N/A
- **Timing:** Event-triggered notifications.
- **Relationships:** Notifications inform users about system updates and evaluation results.
- **Data Format:** Text-based messages.
- **Command Format:** Automated notification generation.
- **End Messages:** Confirmation of notification delivery.

3.1.5 Data Backup and Recovery

- **Purpose:** To perform data backup and recovery operations.
- **Destination:** Backup storage and system recovery.
- **Valid Range:** System data and configurations.
- **Units of Measure:** Data volume and storage capacity.
- **Timing:** Scheduled backups and on-demand recovery.
- **Relationships:** Data backups are crucial for disaster recovery.

- **Data Format:** Encrypted backup files.
- **Command Format:** User-initiated backup and recovery commands.
- **End Messages:** Confirmation of backup and recovery processes.

3.1.6 Data Security Configuration

- **Purpose:** To configure security settings.
- **Destination:** System security controls.
- **Valid Range:** Security parameters and access controls.
- **Units of Measure:** N/A
- **Timing:** User-initiated security configuration.
- **Relationships:** Security settings determine user access and data protection.
- **Data Format:** Security policy settings.
- **Command Format:** User-defined security configurations.
- **End Messages:** Confirmation of security settings updates.

3.1.7 System Notifications

- **Purpose:** To notify users of system events.
- **Destination:** User interfaces and email.
- **Valid Range:** Various system event types.
- **Units of Measure:** N/A
- **Timing:** Event-triggered system notifications.
- **Relationships:** System notifications inform users about important events and updates.
- **Data Format:** Text-based messages.
- **Command Format:** Automated notification generation.
- **End Messages:** Confirmation of notification delivery.

These external interfaces play essential roles in the Course Evaluation System, facilitating data exchange, user interaction, and system operation. Each interface is defined with its purpose, source or destination, data specifics, format, and timing, ensuring seamless integration and functionality.

3.2 Functional Requirements

3.2.1 Input Validation

- The system shall validate all input data for accuracy, completeness, and adherence to specified formats.
- Input validation checks should include verification of course details, student information, and evaluation scores.

3.2.2 Course Information Retrieval

- The system shall retrieve information about courses offered in each semester, including course codes, titles, and other relevant details.
- It should connect with the course allocation system (case study #1) to obtain this information.

3.2.3 Total Marks Entry

- Instructors shall be able to enter total marks against each course.
- The system should allow easy data entry and verification of total marks for accurate evaluation.

3.2.4 Instrument Definition

- The system shall permit the definition of major and minor instruments for each course.
- Major instruments, such as 'sessional,' and minor instruments, e.g., 'sessional 1' and 'sessional 2,' should be configurable.

3.2.5 Mark Division Specification

- The percentage division of marks within the instruments shall be customizable.
- Instructors or administrators should define how marks are distributed within major and minor instruments.

3.2.6 Student Mark Entry

- The system shall facilitate the entry of marks for each minor instrument for every student.
- Instructors and administrators should have access to mark entry capabilities.

3.2.7 Grade and Position Calculation

- The system shall automatically calculate student grades based on their performance.
- Positions within the course shall be determined, considering student marks and grading criteria.

3.2.8 Pass/Fail Reporting

- The system shall generate pass/fail information for each student.
- This information will be integrated with the course allocation system (case study #1).

3.2.9 Department-Level Deployment

- Initially, the system will be introduced in the Department of Computer Science.
- There should be plans to deploy the system to all departments in the University for cross-departmental access.

3.2.10 University-Wide Data Access

- Once deployed University-wide, the system should allow access to student information across all departments.
- Institutional authorities can obtain comprehensive data on students in various departments.

These functional requirements are vital for the Course Evaluation System to fulfill its role in academic evaluation and management. They outline the core actions and operations that the system must perform to meet the needs of the Department of Computer Science and, potentially, other University departments.

3.3 Performance Requirements

The system should support a minimum of 50 concurrent user terminals and handle at least 100 simultaneous users. It must efficiently manage data for 1,000 courses and 5,000 students per semester. Response times for user interactions should not exceed 2 seconds, and data retrieval times must be within 1 second.

Concurrent mark entry by 20 instructors should be supported, and automated calculations (grade and position determination) must complete within 5 seconds. Pass/fail data transfer to the course allocation system should occur within 1 second. These requirements aim to ensure system responsiveness and efficient data processing.

3.4 Non-Functional Requirements

3.4.1 Usability

- The system should have an intuitive user interface to ensure ease of use for instructors, admins, and students.
- The system should provide clear and user-friendly error messages to assist users in resolving issues.

3.4.2 Scalability

- The system should be designed to scale as the university's needs grow. It should support easy expansion to accommodate additional departments and courses.

3.4.3 Performance

- The system should exhibit high performance, ensuring minimal response times for user interactions.
- It should efficiently manage and retrieve data for a large number of courses and students, maintaining acceptable response times.

3.4.4 Security

- User authentication and authorization should be robust, ensuring that only authorized personnel can access sensitive data.
- The system should encrypt sensitive data, such as student marks and personal information.

3.4.5 Reliability

- The system should be highly reliable, minimizing downtime and ensuring data integrity.
- It should support regular backups to safeguard data and facilitate disaster recovery.

3.4.6 Availability

- The system should aim for high availability, minimizing planned downtime for maintenance and updates.

3.4.7 Data Storage and Backup

- The system should efficiently store and manage a large volume of data, including marks, grades, and course information.
- Regular automated data backups should be performed to prevent data loss.

3.4.8 Accessibility

- The system should be accessible to individuals with disabilities, following accessibility standards to ensure inclusivity.

3.4.9 Compliance

- The system should adhere to relevant data protection and privacy regulations.
- It should align with university policies and procedures for data management.

3.4.10 Error Handling:

- The system should effectively handle and log errors, providing clear notifications to administrators for troubleshooting.

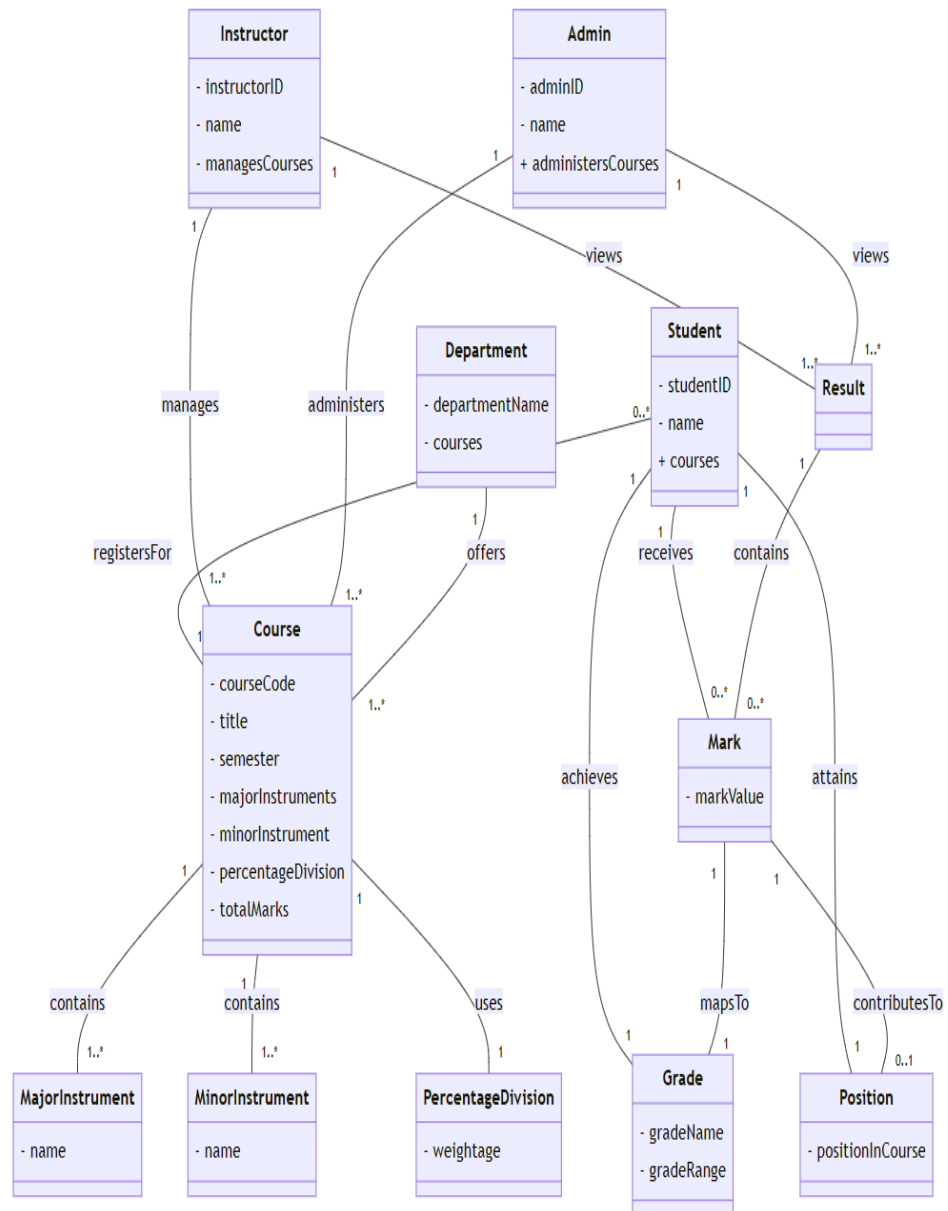
These non-functional requirements are essential to ensure that the Course Evaluation System not only functions properly but also provides a secure, efficient, and user-friendly experience for all stakeholders.

4 Appendixes

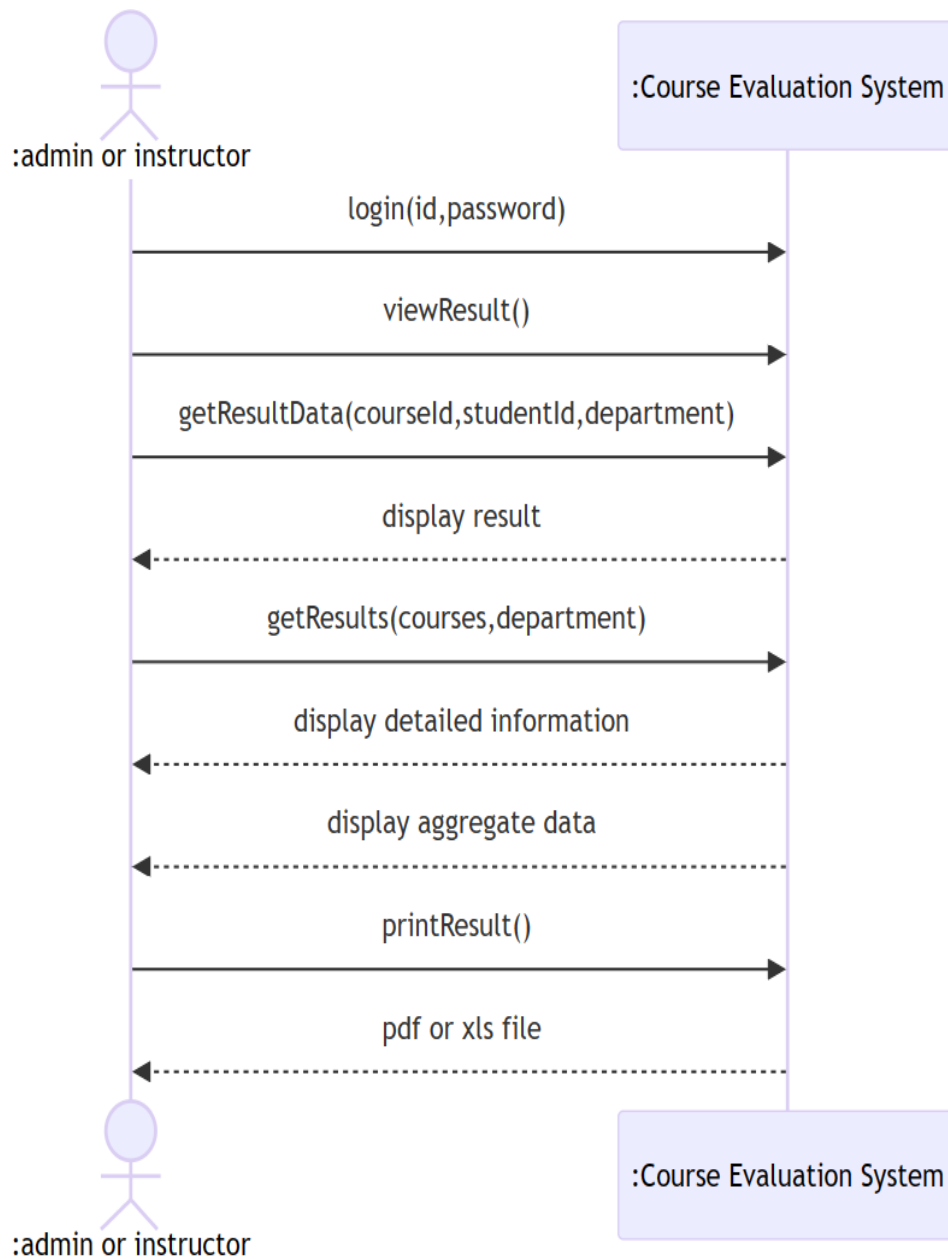
Use Case Diagram

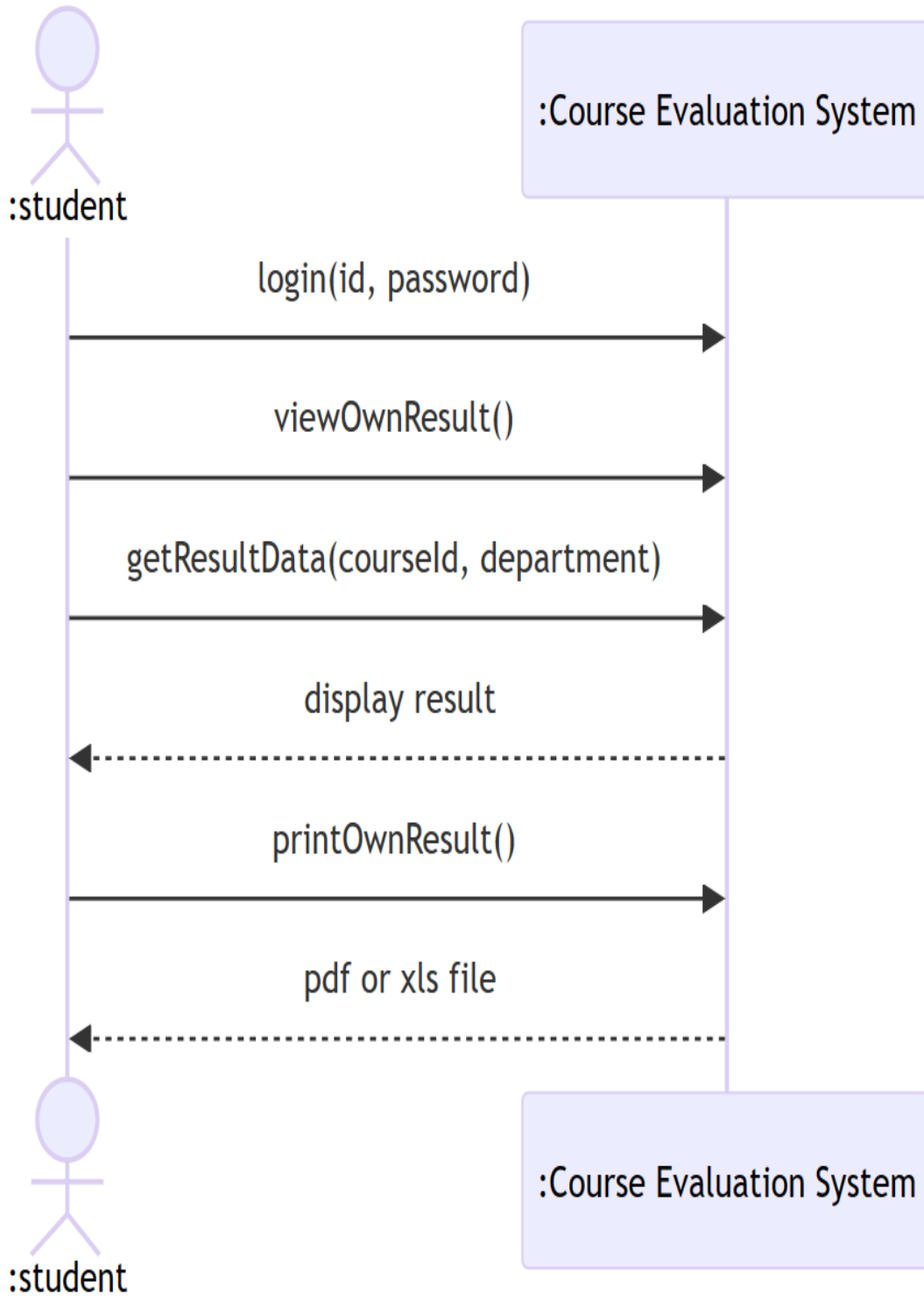


Domain Model

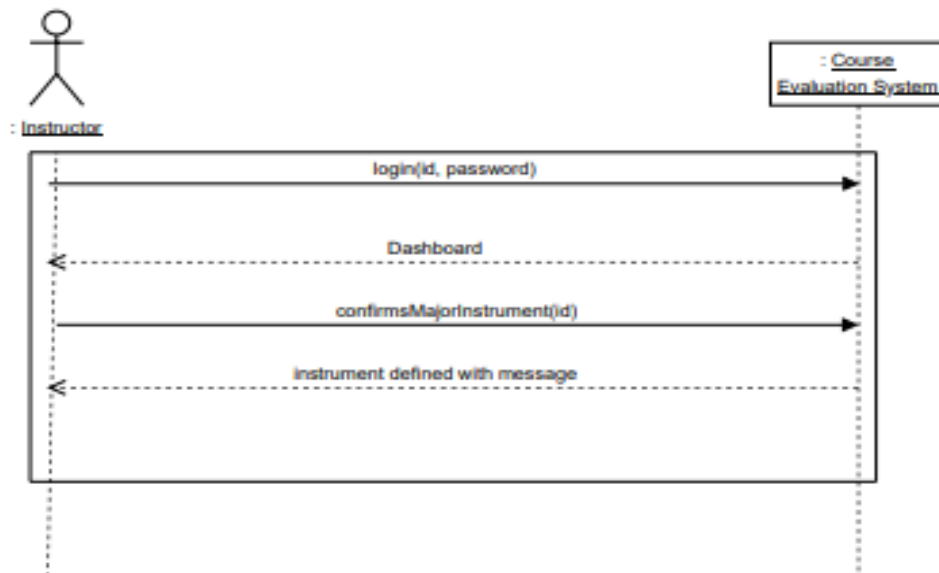
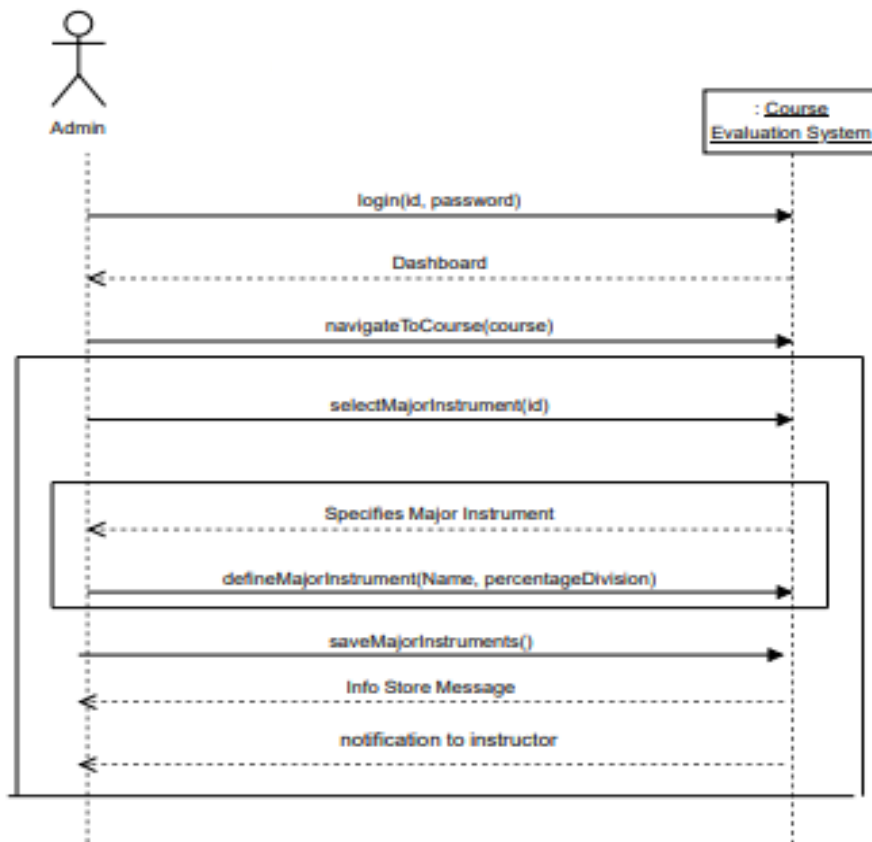


System Sequence Diagrams (View Result)





System Sequence Diagrams (Major Instruments)



System Sequence Diagrams (Minor Instruments)

