**A yellow and red logo

Description automatically generated**

**Department of Information and Communication Technology**

**Faculty of Technology**

**University of Ruhuna**

**SOFTWARE ENGINEERING**

**ICT2232**

**MINI PROJECT | GROUP ASSIGNMENT**

**Group Number 10**

**Submitted to: Ms. Chanduni Gamage**

|  |
| --- |
| Submitted By: |
| TG/2022/1342 - J.M.P.D.W Jayasundara |
| TG/2022/1343 – H.D.P.S Fernando |
| TG/2022/1344 – M.R Sandaruwan |
| TG/2022/1345 – W.A.N.A Perera  TG/2022/1363 – D.H.D Kulathunga  TG/2022/1400 – J.Dhanushiya  TG/2022/1401 – R.Kaviya  TG/2022/1371 – G.R.P.S Gamlath |

**Table of Contents**

1. Introduction
2. Problem Statement
3. System Description
4. Functional Requirements
5. Non-Functional Requirements
6. Methodology (Waterfall)
7. Expected Deliverables
8. Risk Management
9. Quality Assurance
10. Deployment Plan
11. Timeline / Work Plan
12. Resources Required
13. References
14. Appendices
15. Introduction

Background & Problem Domain: The use of online learning platforms by educational institutions is on the rise. But a lot of systems find that they can’t easily handle financial transactions, instructor scheduling, student enrollment and keeping track of courses in one easy-to-use package.

Project Importance: Application developed as a full-blown Java application Markup backed up with backend database as MySQL, to reduce the pain of Course Management and Students progress and receiving the course fee with complex business rules.

Stakeholders:

• Students (register, complete courses, monitor their progress)

• Teachers (monitor students upload materials)

• Administrative functions include managing users, funds and courses.

• Institution Administration (monitor reports to inform decisions)

1. Problem Statement

Existing e-learning platforms often require several tools for numerous tasks, are deficient in reporting and miss finance management modules. This is inefficient and difficult to passively monitor students’ progress.

Our solution: a lightweight Java based platform, with a MySQL database and clear-cut processes that can be used by small and medium sized companies.

1. System Description

System Environments

* Platform: Desktop (Java Application)
* Databases: MySQL
* Operation System Support: Windows/ Linux
* Deployment: Simple local host machine

Major Features

* Course Management
* User Management
* Enrolment & process Tracking
* Reporting & Analytics

Use Cases

1. Use case – User Registration and login

Primary actor – Admin, Instructor, Student

Pre- Condition – User has accessed the system

Main Success scenario-

1. user select login or register, then user enters credentials.
2. The system validates the credentials; system allow to the access the system processes.

Extensions-

1.a user enter invalid credentials,

1. system display “invalid username or password”

2.a user is not register,

1. system display” please register”.

1. Use case – Course management

Primary Actor – Admin/ instructor

Pre- Condition – Admin/ instructor is logged into the system.

Course lists are registered into the system

Main Success scenario-

1. Select “Manage courses”
2. Choose to add, update, delete a course details.
3. Save the details to update
4. Update the course list.

Extensions-

2.a. enter incomplete course details.

1. System display “fill all the details”.

3.a. Tries to delete an enrolled student’s course details

1. System display “cannot be deleted as student are enrolled”.

1. Use case – Course Enrollment and progress tracking

Primary Actor – Student

Pre- Condition – Student is logged into the system.

Courses are registered into the system.

Main Success scenario-

1. The student search course details
2. Student select a course and enroll.
3. System verifies course.
4. System enrolls the student in the course.
5. System monitors the student course details

Extensions-

2.a. Student tries to enroll without specific requirements.

1. system display “Enrollment failed.”

3.a. Course enrollment limit reached

1.system display” course is full”.

1. Use case – Update study martials

Primary Actor – Instructor

Pre- Condition – Instructor is logged into the system.

Courses are register in the system.

Main Success scenario-

1. Instructor select a course.
2. Instructor chooses the file.
3. System validates the file format and size.
4. System stores the file and link to the particular course.
5. System confirms the upload

Extensions-

3.a. Instructor uploads unsupported file type

1.system display “Invalid file format, please upload PDF, DOC…”

3.b. File size exceeds the limit

1.system display “file too large.”

4.a. Upload fails due to server issue.

1.system display “please try again system failure”.

1. Use case – Reporting and analysis

Primary Actor – Admin, Instructor

Pre- Condition – Actor is logged into the system.

Courses and students are register in the system.

Main Success scenario-

1. Actor select the reports & analysis.
2. Actor choose a report type.
3. System retrieves particular data from the database.
4. System process and generate the report.
5. System displays the particular report to the actor.

Extensions-

3.a. Data unavailable for the selected report.

1. system display “no data for this report”.

3.b. Database error occurs

1.system display “Please try again, database error”.

4.a. System error occur due to report generate fails

1.System display “please try again report generate fails”

User Stories

1. **User Registration and Login**

* As a user, I want to register account for access the system.
* As a user, I want to login with my credentials to securely access the account.

**2. Course Management**

* As an admin, I want to add new courses
* As an admin, I want to update courses details
* As an admin, I want to delete courses details

**3. Course Enrollment and Progress Tracking**

* As a student, I want to search for available courses and I can find suitable course for me.
* As a student, I want to enroll in the course.
* As a student, I want to track my progress and I can monitor the course.

**4. Update Study Materials**

* As an instructor, I want to upload study materials for course.
* As an instructor, I want to check study materials type and size for course.
* As an instructor, I want to receive confirmation of upload

**5. Reporting and Analysis**

* As an admin, I want to generate reports.
* As an admin, I want to check the data to retrieve the accurate data

System architecture

* Presentation Layer: Java Swing
* Application Layer: Java business logic classes
* Database layer: MySQL with JDBC

1. Functional Requirements

|  |  |  |
| --- | --- | --- |
| Req. ID | Description | Priority |
| FR-1 | Users shall register, log in, and recover passwords securely. | High |
| FR-2 | Instructors shall create, update, delete courses and materials. | High |
| FR-3 | Students shall enroll and unenroll in courses. | High |
| FR-4 | System shall support multimedia content uploads and streaming. | Medium |
| FR-5 | Provide quizzes, assignments, and automated grading functions. | High |
| FR-6 | Enable communication via forums, messaging, and notifications. | Medium |
| FR-7 | Generate and display progress and performance reports. | High |
| FR-8 | Administrators shall manage roles and permissions. | High |
| FR-9 | Provide search functionality for courses and resources. | Medium |

1. Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| NFR ID | Category | Requirement Description |
| NFR-1 | Performance | Support up to 1000 concurrent users with < 3s response time. |
| NFR-2 | Security | Implement HTTPS, data encryption, multi-factor authentication. |
| NFR-3 | Scalability | Auto-scale resources to handle growing user base and data volume. |
| NFR-4 | Reliability | Ensure 99.5% uptime with backup and disaster recovery strategies. |
| NFR-5 | Compatibility | Compatible with major browsers & mobile OS (iOS, Android). |
| NFR-6 | Usability | Comply with WCAG 2.1 AA accessibility standards. |
| NFR-7 | Maintainability | Modular architecture and comprehensive documentation. |
| NFR-8 | Interoperability | Integrate with SIS, SSO, payment gateways, and third-party content providers. |
| NFR-9 | Localization | Support multi-language and regional settings for diverse users. |
| NFR-10 | Backup/Recovery | Regular data backups and fast recovery options. |

1. Methodology (Waterfall)

We use the waterfall SDLC for our development process

1. Requirement Analysis = Collect document system needs
2. System Design = Create database ER diagram, System architecture design using figma
3. Implementation = Write java code and integrate with database (MySQL)
4. Testing = Functionality, Usability, System test
5. Deployment= install system on local
6. Maintance= fix bugs and minor enhancement
7. Expected Deliverables

* Working Java- based E-learning management System
* MySQL database schema
* Documentation for user manual
* Presentation and demonstration

1. Risk Management
   * **Security Risks**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Risk ID | Description | Likelihood | Impact | Mitigation | Risk ID |
| S-1 | SQL Injection from unsanitized JDBC queries exposing sensitive student and financial data | High | High | Use prepared statements, input validation, and encryption | S-1 |
| S-2 | Weak MySQL configurations (default credentials, poor access control, brute force vulnerability) | High | High | Strengthen database security, enforce strong credentials, enable intrusion monitoring | S-2 |
| S-3 | Java Swing risks (weak encryption, hardcoded credentials, memory dump leaks) | Medium | High | Use secure encryption libraries, avoid hardcoded credentials | S-3 |
| S-4 | E-learning system threats (phishing, DDoS, breaches, misuse of PII and academic/financial data) | High | High | Continuous monitoring, security training, layered defenses | S-4 |

* + **Methodological risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk ID | Description | Likelihood | Impact | Mitigation |
| M-1 | Requirements changes after initial phase causing rework and cost overrun | High | Medium | Use Agile increments with integrated feedback |
| M-2 | Late defect discovery in JAR files, leaving deprecated features unchecked | Medium | High | Continuous testing and code review |
| M-3 | Minimal end-user feedback, leading to features not aligned with user needs | Medium | Medium | Gather feedback early and iteratively |

* + **Data Protection Risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk ID | Description | Likelihood | Impact | Mitigation |
| D-1 | Financial data exposure (payment details) causing compliance and liability issues | Medium | High | Strong encryption, compliance frameworks |
| D-2 | Student privacy risks (PII, academic info leaks) | High | High | Enforce privacy policies, access control |
| D-3 | Backup/recovery gaps leading to permanent data loss | Medium | High | Regular automated backups, tested recovery plans |

* + **Operational Risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk ID | Description | Likelihood | Impact | Mitigation |
| O-1 | Local deployment with single point of failure, poor scalability | Medium | High | Plan scalable XAMPP/cloud deployment |
| O-2 | Maintenance complexity from 8 developers working in silos | Medium | Medium | Enforce coding standards, regular sync-ups |
| O-3 | Dependence on specific software versions causing failures | Medium | Medium | Version control, environment standardization |

* + **All technical risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk ID | Description | Likelihood | Impact | Mitigation |
| T-1 | Poorly handled JDBC connection strings exposing databases | High | High | Secure JDBC handling with strict encryption |
| T-2 | Sensitive data in JAR files exposed due to poor coding/configuration | Medium | High | Secure coding practices, code reviews |
| T-3 | Complexity from integrating Swing, MySQL, and XAMPP increases failure points | Medium | Medium | Modular design, testing integration points |

1. Quality Assurance

* Testing: Unit (Modules / classes), Integration (DB + UI), System
* Code Review: Peer review with team, Using GitHub
* Quality Metrics: Error rate per module, response time, successful enrollment

1. Deployment Plan

* Hosting: Local server by Xampp + Java
* Installation guide: Documented step by step for java Environment and MySQL DB import
* Future Scalability: option to migrate to cloud if needed later

1. Timeline / Work Plan

|  |  |  |
| --- | --- | --- |
| Phase | Duration | Responsibilities |
| Requirement Analysis | 1 week | All members |
| Design | 2 weeks | 4 members focus on DB design, 4 members focus on System Design |
| Implementation | 5 Week | 4 members code modules, 4 members integrate |
| Testing | 2 week | 5 members lead testing code and 3 members testing UI design |
| Development & Documentation | 1 week | 2 members prepare final report & Install System |
| Maintance | 1 week | All members find bugs and fix |

1. Resources Required

* Hardware Components = laptops, mouse, keyboard
* Software
* JDK
* MySQL Server and XAMPP Server
* IDE (JetBrains, NetBeans)
* GitHub

1. References

IEEE (1998). IEEE Std 830-1998 – IEEE Recommended Practice for Software Requirements Specifications. IEEE.

→ A standard reference for writing requirements, including non-functional requirements.

1. Appendices

* ER diagrams for MySQL Database
* USE case Diagram
* Class Diagram