|  |
| --- |
| Name: R.M.S.D.Samarakoon |
| Student Reference Number: 10820263 |



|  |  |  |
| --- | --- | --- |
| Module Code: PUSL2023 | Module Name: Mobile App Development | |
| Coursework Title: LMS Mobile Application | | |
| Deadline Date:18/05/2023 | | Member of staff responsible for coursework: 06 |
| Programme: | | |
| Please note that University Academic Regulations are available under Rules and Regulations on the University website [www.plymouth.ac.uk/studenthandbook](http://www.plymouth.ac.uk/studenthandbook). | | |
| Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.  R.M.S.D.Samarakoon - 10820263  Thellambure Hettige Amantha - 10820863  Galappaththi Dinujaya -10820276  Rankoth Jayasekara – 10820280  Maleesha S Ramasinghe -10820265  Navoda Manamperi - 10820266  ***We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.***  Signed on behalf of the group: | | |
| Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***  A close-up of a word  Description automatically generated with low confidence  Signed : | | |
| Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.  I \*have used/not used translation software.  If used, please state name of software………………………………………………………………… | | |
| **Overall mark \_\_\_\_\_% Assessors Initials \_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_** | | |

\*Please delete as appropriateSci/ps/d:/students/cwkfrontcover/2013/14

**Table of contents**

1. Project Description
2. Requirements and the project scope
3. Planning and organizing of the activities.
4. Declaration of the APIs, templates, components, technologies used.
5. Issues faced during the implementation.
6. Approach adopted to resolve.
7. Individual effort
8. **Project Description**

The LMS project is a software development initiative aimed at creating a Learning Management System (LMS). An LMS is a web-based platform that facilitates the management, delivery, and tracking of educational content and resources. The primary objective of this project is to develop a user-friendly and scalable LMS to enhance the learning experience for students and streamline administrative tasks for educational institutions. Through this project, we aim to provide a robust solution that promotes effective online learning and supports the diverse needs of educators and learners.

The objectives of a learning management system (LMS) typically include:

1. Centralized content management: An LMS aims to provide a centralized platform for organizing and managing educational content, including course materials, videos, documents, quizzes, and assignments. This objective ensures easy access to learning resources for both educators and students.

2. Seamless course delivery: The LMS facilitates the delivery of courses and learning materials in a structured manner. It allows educators to create and publish courses, set up modules or lessons, and track progress. Students can access the courses, navigate through the content, and engage in interactive learning activities.

3. Collaboration and communication: An LMS promotes effective communication and collaboration among students and educators. It provides features such as discussion boards, messaging systems, and virtual classrooms, enabling learners to interact with peers and instructors, ask questions, share ideas, and participate in group activities.

4. Assessment and feedback: The LMS enables educators to create and administer assessments, such as quizzes and exams, online. It automates grading processes and provides instant feedback to students. This objective ensures efficient evaluation of student progress and performance.

5. Progress tracking and reporting: An LMS allows educators and administrators to monitor and track the progress of learners. It generates reports on individual and class-level performance, providing insights into areas of improvement and identifying learning gaps.

6. Personalized learning experience: With the help of an LMS, educators can customize learning paths and content based on individual student needs and preferences. Adaptive learning features and personalized recommendations enhance the learning experience and cater to diverse learning styles.

7. Administrative efficiency: The LMS simplifies administrative tasks associated with course management, enrolment, scheduling, and record-keeping. It reduces manual effort, streamlines workflows, and enhances overall operational efficiency for educational institutions.

8. Accessibility and flexibility: An LMS aims to provide anytime, anywhere access to learning materials, allowing students to learn at their own pace and convenience. It supports various devices and ensures accessibility for learners with disabilities, fostering inclusivity in education.

These objectives collectively contribute to improving the learning process, increasing engagement, and enhancing educational outcomes for both students and educators.

1. **Requirements and the project scope**

REQUIREMENTS:

1.Determine the various user roles (such as students, teachers, and administrators) and provide

their unique access levels and permissions inside the LMS.

2.Describe the elements necessary for managing courses, such as developing and arranging

course materials, establishing evaluations, controlling enrolments, and monitoring student

progress.

3.The supported material types (such as documents, videos, and quizzes), file formats, and any

particular specifications for uploading, organizing, and distributing content inside the LMS

should be specified.

4.To encourage engagement between students and instructors, decide which communication and

collaboration options, such as discussion boards, message services, and live chat features,

are most helpful.

5.Determine the necessary assessment kinds (such as quizzes, assignments, and examinations),

grading methods, and any particular needs for automated grading or scoring.

6.Reporting and Analytics: Describe the reporting and analytics capabilities necessary to

monitor student performance, provide progress reports, and compile information for evaluating

the success of curricula and teaching strategies.

7.Integration: Identify any requirements for integrating your software with external systems,

such as student information systems, authentication systems, or other third-party applications.

SCOPE:

1.Functional Scope: Clearly outline the main features and functions that will be incorporated

into the LMS's initial release, making sure that they comply with the cited standards.

2.To manage expectations and prevent scope creep, identify any features or functions that

are expressly prohibited from the project's scope.

3.modification and branding: Describe the degree of modification and branding necessary

to align the LMS with the visual identity and needs of the enterprise.

4.Technical scope: Describe the necessary technical infrastructure components, including the

hosting environment, database, programming languages, and any specialized system integrations.

5.timeline and Milestones: Establish a clear roadmap for development, testing, and deployment

by defining the project's timetable, which should include important milestones, deliverables,

and dependencies.

6.Support and Maintenance: Talk about the LMS's continuing support and maintenance needs,

such as bug repairs, software upgrades, and user assistance.

1. **Planning and organizing of the activities.**

Our LMS project plan, like all projects, requires a time frame. So, we have created a time frame to accomplish certain tasks after the team members are included. We also spent a significant amount of the time we were given to implement the LMS and resolve some of its issues.

We thought it best to always leave room for LMS deployment to debug and resolve certain situations. We also made plans to conduct activities in a collaborative environment.

Because market testing can be a tedious process that takes some time and the people who are dedicated to each task may not have good knowledge about that type of software, so our plan for that was the most honest and clear communication between the members. All those who know the unknown come together to do it. We also dedicate enough time to identify our training need.

Creating a list of tasks to complete the process.

Having access to all the LMS configuration information we need. User profiles include security settings for the LMS, administrator roles and course structures used. Also, various protocols and procedures are included.

And suggesting ways we can make implementation easier.

Another aspect of effective LMS project planning is having a list of clearly defined expectations and goals in advance. Include the issues the online platform will address and the goals we hope to achieve.

This ensures that the primary objectives of the LMS are met.

Collectively drawing a map and a path from where we are to where we want to get. It grows and strengthens the relationship between the vendor and the client. Also gain an understanding of how LMS can support training needs

Finally, learners can log in and see how they interact and respond.

1. **Declaration of the APIs, templates, components, technologies used.**

We may make use of a variety of APIs, templates, components, and technologies to create a mobile-based Learning Management System (LMS) for students. How to use them is as follows:

1. API(Application Programming Interfaces)

* Students can safely access the LMS by using the authentication API, which is used for user authentication and login functions. This API can provide its own user management or interact with current authentication solutions.
* File Storage API: This feature enables students to upload and download lecture materials from the LMS, including PDFs, videos, and slides.
* The Schedule API allows students to view forthcoming classes, their times, and any modifications or notices by retrieving and displaying lecture schedules for them.
* Grading API: Retrieves exam and assignment results for each module and displays the information to students in the LMS.

1. Templates

Login Page Template: Offers a pre-made login page with login buttons and input areas for username and password. The LMS branding can be matched to this template.

* The lecture materials are arranged by modules or themes on the lecture materials page template, which offers an organized arrangement for displaying lecture materials. It might have functions like search, filtering, and sorting choices.

Lecture Schedule Page Template: Provides a calendar or list view of the lecture schedule in a user-friendly way. Date, time, module, speaker, and any other remarks should all be included.

* Marks Page Template: Enables students to monitor their progress by displaying the test and assignment grades for each module. Charts and progress bars are examples of graphical elements that might be used in this template.

1. Components

* A constant navigation bar is provided throughout all pages of the LMS, making it simple for students to navigate between its various components.

Button Component: Used for navigational functions like returning to the app's home page or going to the next page. A button to return to the home page ought to be present on every page.

* The actual lecture materials, the schedule details, and the marks information are shown on the corresponding pages by the content display component (c).

Search Component: This feature enables students to look for particular modules or lecture materials within the LMS.

* Filter and Sort Component: This feature enables students to filter and arrange the lecture materials or schedule according to a variety of factors, including date, lecturer, and subject.

1. Technologies

* Mobile App Development Framework: To create the mobile app, use a framework like React Native, Flutter, or Xamarin. Because these frameworks provide cross-platform development, both iOS and Android device compatibility is guaranteed.
* Backend development: To handle server-side logic, database administration, and API connections, use a backend technology like Node.js, Django, or Ruby on Rails.
* Database: Keep user information, course materials, timetables, and grades in a database system like MySQL, PostgreSQL, or MongoDB.
* User Interface (UI) Design: To ensure a smooth user experience, utilize UI design tools like Sketch, Adobe XD, or Figma to develop aesthetically pleasing and simple app interfaces.
* Version Control: Use Git and platforms like GitHub or GitLab for version control to facilitate developer collaboration and guarantee the stability and upkeep of the code.

We can develop a mobile app that gives students quick access to lecture materials, lecture schedules, and their exam/assignment grades, improving their learning experience. This is accomplished by utilizing these APIs, templates, components, and technologies.

1. **Issues faced during the implementation.**

Compatibility: Due to differences in screen sizes, operating systems, and hardware capabilities, making sure the app functions flawlessly on various mobile platforms (iOS, Android) and devices can be difficult.

User experience: Because of the small screen size and the need for careful navigation, readability, and usability, designing an intuitive and user-friendly interface for a mobile app can be challenging.

Performance: It is important to optimize mobile apps for speed and effectiveness as long loading times or lags can negatively affect the user experience. Code must be optimized, network queries must be kept at a minimum, and caching techniques must be used.

Security: Sensitive user data, such as personal information and educational materials, must be handled by mobile apps. It is essential to have strong security mechanisms to safeguard data throughout transmission, storage, and user authentication.

Connectivity: For functions like content updates, synchronization, and real-time interactions, mobile apps primarily rely on network connectivity. It is important to consider all possible network connection limitations or instability.

Integration: It can be difficult to integrate the mobile app with already-used LMS platforms, databases, or outside systems. It takes careful planning and execution to provide flawless data synchronization, authentication, and interoperability.

Testing and quality assurance: To find and fix problems with functionality, performance, and compatibility, it is essential to test the app exhaustively across a range of platforms, operating systems, and usage scenarios.

Updates and maintenance: Developing mobile apps is a continuous process, and updates are required frequently to address issues, add new features, and guarantee compatibility with developing technologies and security norms.

User help: To keep users happy and engaged with the LMS mobile app, quick and effective help is crucial for answering their questions and resolving problems.

The performance and user experience of the app must be constantly improved, which requires careful planning, working with skilled developers, conducting rigorous testing, and obtaining user input.

1. **Approach adopted to resolve.**

The following strategies can be used to address problems with an LMS mobile app:

To clearly identify the aims and objectives for the app, conduct a thorough analysis of the requirements and user demands. This makes it easier to make sure the app caters to the requirements of the intended user base.

Create an intuitive and user-friendly interface by using a user-cantered design strategy. Usability testing, user research, and user persona creation should all be done to gather feedback and iterate the design for the best possible user experience.

Use cross-platform development frameworks to build the app, such as React Native or Flutter, which enable code reuse and concurrent development for several platforms. This promotes uniformity and lowers development effort.

Performance Optimization: To increase the speed and effectiveness of your program, use performance optimization techniques including code optimization, image compression, lazy loading, and caching systems. Performance testing should be done to locate bottlenecks and enhance overall performance.

Security Measures: To protect user data and maintain privacy, implement strong security measures such secure data transmission (HTTPS), encryption of sensitive data, secure user authentication methods, and compliance with applicable data protection laws (such as the GDPR).

Offline Capabilities: Incorporate offline capabilities into the app, allowing users to access and interact with content even when they have limited or no network connectivity. Implement local data storage and synchronization mechanisms to enable offline usage and seamless data synchronization.

Regular Maintenance: Schedule routine app upgrades to address bug problems, add new features, and enhance performance. To determine areas for improvement and prioritize updates, collect user input and track app usage trends.

User Support and Feedback: Establish explicit methods for receiving user support and feedback, such as an in-app chat feature or a separate support website. To better understand user demands and improve the app's functionality and usability, actively respond to user inquiries, quickly resolve problems, and collect feedback.

By adopting these approaches, development teams can address the challenges faced during the implementation of an LMS mobile app and deliver a high-quality and user-friendly solution.

1. **Individual effort**

|  |  |
| --- | --- |
| **Name** | **Individual contribution** |
| R.M.S.D.Samarakoon - 10820263 | Front End Coding  Database Connection |
| Thellambure Hettige Amantha - 10820863 | UI coding  Backend Coding |
| Galappaththi Dinujaya -10820276 | Backend Coding  Database connecting |
| Rankoth Jayasekara - 10820280 | Ui designing.  Creating Report |
| Maleesha S Ramasinghe - 10820265 | Ui designing.  Creating Report |
| Navoda Manamperi - 10820266 | Ui designing.  Creating Report |