**Chapter 3 Requirement Specifications**

**3.1 Existing System**

Existing Learning Management Systems (LMS) often struggle with clunky interfaces, limited content formats, and weak communication tools. This can hinder both instructor and learner engagement. Our proposed web-based LMS portal aims to address these issues by providing a user-friendly platform with diverse content delivery options, improved communication features, and robust reporting and analytics, fostering a more effective learning experience.

**3.2 Proposed System:**

The Prototype development methodology will be adopted in the development of SkillSync LMS portal website. The methodology process allows developers to create only the prototype of the solution to assess its functionality and make necessary modifications before developing the actual application. The best feature of this software development methodology is that it solves many issues which often occur in a traditional waterfall model. It is an example of a plan-driven process in principle, one must plan and schedule all of the process activities before starting work on them. This method drastically decreases the risk of failure, as probable risks can be detected in early stage and moderation steps can be taken quickly. Other than the prototype model there are other models such as the Waterfall, Agile, Spiral model etc. The waterfall model is based on a sequential flow requiring the previous phase to be completed before moving onto the next phase. Unlike the prototype model the waterfall model can only be adopted in the development of projects whose requirements are clearly stated. Agile software development methodology is a blend of both incremental and iterative process models which allows recurrent modification in development. One drawback in the agile development methodology is the possibility of drifting from the path to required goal as a result of ambiguity in the customers understanding of the outcome of the project. The prototype development methodology helps in requirement gathering and requirement analysis when there is absence of requirement documents. When a prototype is shown to the clients, they get a vibrant understanding and whole sense of the functionality of the software. Thus, the prototype methodology has been chosen to be the methodology to be used in the development of the SkillSync LMS portal website.

The current landscape of online learning platforms is filled with limitations. Instructors struggle with clunky interfaces and restricted content delivery options, while learners face limited engagement and interaction. Our proposed web-based LMS portal aims to revolutionize this by offering a user-centric platform that fosters a dynamic and effective learning experience.

Before designing any web based system, it is essential and helpful to establish the objectives that the web based system should satisfy. In addition, the relative importance of each objective should be established keeping in mind the drawbacks of the existing system.

1. **Efficiency:** Enhancing the efficiency of online learning through streamlined navigation and optimized performance.
2. **Data Security:** Ensuring the security of user data through robust encryption and access controls to protect sensitive information.
3. **Time Management:** Allowing users to schedule appointments and access course materials online to save time and improve convenience.
4. **Accuracy:** Maintaining accurate records of user progress and content updates to ensure reliable learning experiences.
5. **Flexibility:** Adapting to future requirements and technological advancements to meet the evolving needs of learners and educators.
6. **Reliability:** Prioritizing system stability and data integrity to deliver a consistent and dependable learning environment.
7. **Ease of Use:** Providing an intuitive interface and comprehensive support resources to facilitate seamless navigation for all users.

3.3 Requirement Specifications

A Requirement Specification encompasses all the requirements to be applied to the design and validation of a product. It also includes supplementary information essential for product design, validation, and maintenance.

3.3.1 Functional Requirements

Functional requirement is referring to the functionalities must be apply to a system. The Functional requirement of SkillSync LMS portal website is stated below:

* The portal must display information to users in real-time, ensuring that updates and changes are immediately visible.
* The portal must provide a comprehensive list of available services, including course offerings, resources, and support materials.
* The portal must include detailed profiles of educational institutions, instructors, and other relevant entities, making it easy for users to contact them directly.
* Users must be able to access and retrieve information from the LMS portal using any device with internet connectivity, ensuring flexibility and accessibility.
* The interface of the portal must be visually appealing, intuitive, and user-friendly, adhering to standard design principles including consistent use of colors, fonts, and buttons. This will enhance the user experience and promote engagement with the platform.

3.3.2 Non-Functional Requirement

* The system must feature a visually appealing Graphical User Interface (GUI) to enhance user experience and engagement.
* Security measures should be integrated into the system to protect records and ensure data security.
* The system should be capable of recovering from crashes or failures to minimize downtime and data loss.
* It should be designed to work seamlessly with different operating systems, allowing for flexibility and customization based on user requirements.
* Usability goals include creating a self-explanatory system that does not require extensive tutorials for users to understand and navigate.
* **Performance Requirement:** The system must deliver responsive performance, with quick response times to user inputs. Load times may vary based on internet connection strength and hardware components, ensuring compatibility across a range of devices and operating systems.
* **Operational Requirements:** The LMS portal will be used by a large number of users and should be easily accessible and beneficial to all.
* **Security Requirements:** Given the storage of classified data in the Firebase database, stringent security measures must be implemented to protect sensitive information.
* **Access Requirements:** Administrators should have full access to all system functionalities, including database management and user activity monitoring. Users should have access to their profiles and relevant information, such as courses.

**3.3.3 Requirements:**

System requirements, including hardware and software specifications, serve as guidelines rather than strict rules. For the Transcodes website, the following requirements are defined:

**3.3.4 Hardware Requirements:**

**Device:** Desktop, Laptop, Tablet

**Operating System:** Any modern operating system (e.g., Windows, macOS, Linux)

**3**.**3.5 Software Requirements:**

Web Browser, Visual Studio Code, Firebase, React.js, Node.js

**Web Browser:** A web browser is a software application used to access and navigate the World Wide Web. Users interact with websites by entering URLs (Uniform Resource Locators) or clicking on hyperlinks, and the browser retrieves and displays web pages from remote servers.

For the Transcodes website, compatibility with modern web browsers ensures that users can access and interact with the website's content seamlessly. Here's a brief overview of the mentioned browsers:

**Google Chrome:** Developed by Google, Chrome is one of the most popular web browsers known for its speed, stability, and extensive support for web technologies.

**Mozilla Firefox:** Firefox is an open-source web browser developed by the Mozilla Foundation. It prioritizes privacy and security features while offering customizable options for users.

**Safari:** Safari is the default web browser for Apple devices, including Mac computers, iPhones, and iPads. It is known for its fast performance and integration with Apple's ecosystem.

**Microsoft Edge:** Edge is Microsoft's web browser that replaced Internet Explorer. It offers a streamlined user interface, fast browsing speeds, and tight integration with Windows 10 features.

**Opera:** Opera is a lesser-known browser that focuses on speed, security, and innovative features such as built-in ad-blocking and VPN (Virtual Private Network) capabilities.

By ensuring compatibility with these browsers, the Transcodes website can reach a broad audience of users across different devices and platforms, providing them with a consistent and reliable experience when accessing the website's content and features.

An **Integrated Development Environment**, or IDE, is a software application that provides comprehensive tools and features to facilitate software development. IDEs are primarily used by developers during the coding and debugging phases of a project. Here's a more detailed explanation:

1. **Code Writing and Editing:** IDEs offer advanced text editing capabilities, including syntax highlighting, auto-completion, and code snippets. These features help developers write code more efficiently and with fewer errors.
2. **Debugging:** IDEs come with built-in debugging tools that allow developers to identify and fix errors in their code. Debuggers provide features such as breakpoints, watch lists, and step-through execution, enabling developers to pinpoint and resolve issues quickly.
3. **Project Management:** IDEs provide project management features to organize code files, resources, and dependencies. Developers can create, open, and manage projects within the IDE, making it easier to navigate and work with large codebases.
4. **Version Control Integration:** Many IDEs integrate with version control systems like Git, allowing developers to track changes, collaborate with team members, and manage code repositories directly from the IDE.
5. **Build and Deployment:** IDEs often include tools for building and deploying applications. Developers can configure build settings, compile code, and package applications for distribution directly within the IDE.
6. **Plugin Ecosystem:** IDEs support a wide range of plugins and extensions that extend their functionality. Developers can customize their IDEs with plugins for specific languages, frameworks, or tools, enhancing productivity and workflow efficiency.

Popular examples of IDEs include:

* **Visual Studio Code:** Developed by Microsoft, Visual Studio Code is a lightweight and versatile IDE known for its extensive plugin ecosystem, support for multiple programming languages, and built-in Git integration.
* **Sublime Text:** Sublime Text is a powerful and customizable text editor with a minimalist interface. It offers features like multiple cursors, command palette, and customizable key bindings, making it popular among developers for its speed and simplicity.

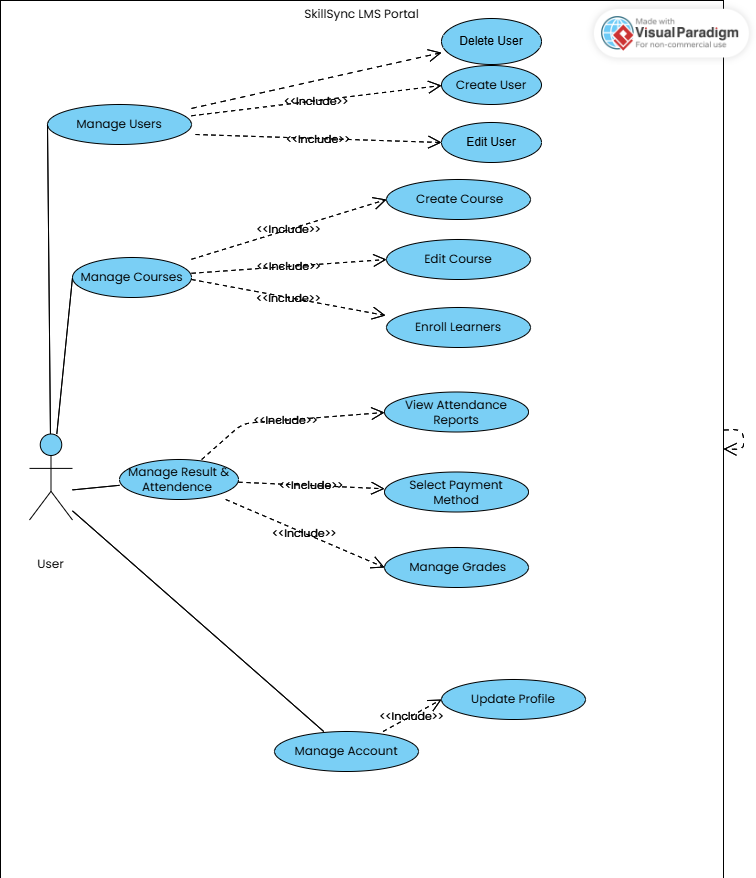
**Libraries and Frameworks:** The website is built using libraries and frameworks such as React.js and Node.js. React.js is a JavaScript library for building user interfaces, while Node.js is a JavaScript runtime environment used for server-side development. These tools facilitate efficient development and help create dynamic and interactive web applications.

**Firebase :** Firebase is a Cloud-hosted, NoSQL database that uses a document-model. It can be horizontally scaled while letting you store and synchronize data in real-time among users. This is great for applications that are used across multiple devices such as mobile applications. Firebase is optimized for offline use with strong user-based security that allows for server-less based apps as well. Firebase is built on the Google infrastructure and is built to scale automatically. In addition to standard NoSQL database functionality, Firebase includes analytics, authentication, performance monitoring, messaging, crash reporting and much more. Because it is a Google product, there is also integration into a lot of other products. This includes integration with Google Ads, AdMob, Google Marketing Platform, the Play Store, Data Studio, BigQuery, Slack, Jira, and more.

The Firebase APIs are packaged into a single SDK that can be expanded to multiple platforms and languages. This includes C++ and Unity, which are both popular for mobile development.

**3.4 Use Cases**

In system analysis, a use case serves as a crucial tool for identifying, organizing, and understanding system requirements. It consists of a series of potential interactions between users and the system within a specific context, all aimed at achieving a particular objective. This method involves documenting every step a user takes to accomplish a task, providing a clear outline of system functionality. Typically crafted by business analysts, use cases find application across various stages of software development, including requirement planning, design validation, software testing, and documentation creation. By detailing user-system interactions, a use case document helps the development team anticipate and address potential errors or issues that may arise during system transactions, ensuring a smoother development process and a more robust final product.



**Figure 3.1** Admin Use Case Diagram

**Admin Use case description**

The figure 3.1 shows the interactions between the system and the admin of the system. The stick figure at the left side of the rectangle represents the admin and the rectangle defining the scope of the system. The oval shape (use case) within the rectangle represents actions that accomplish some task within the system. There is also admin panel which is used by admin to manage all activities shown on user panel. Admin can add, update and delete data and manage all courses.

**Table 3.1: UC-01 - Login**

|  |  |
| --- | --- |
| Use case ID | UC-01 |
| Use case Name | Login |
| Actors | Admin |
| Purpose | Login to access admin panel |
| Priority | High |
| Pre-condition | LMS web portal must be accessed |
| Post-condition | Admin successfully logged in and verified password |
| Actor action | Click the login button |
| Main Scenario | Admin opens the LMS web portal and the login screen appears |
| Alternate Scenario | N/A |

**Table 3.2: UC-02 - Verify Password**

|  |  |
| --- | --- |
| Use case ID | UC-02 |
| Use case Name | Verify Password |
| Actors | Admin |
| Purpose | Verify password during login |
| Priority | High |
| Pre-condition | LMS web portal must be accessed |
| Post-condition | Admin successfully verified password |
| Actor action | Click the login button |
| Main Scenario | Admin opens the LMS web portal and the login screen appears |
| Alternate Scenario | N/A |

**Table 3.3: UC-03**

|  |  |
| --- | --- |
| Use case ID | UC-03 |
| Use case Name | Display Login Error |
| Actors | Admin |
| Purpose | Check if password is wrong |
| Priority | High |
| Pre-condition | LMS web portal must be accessed |
| Post-condition | Display error message on wrong password |
| Actor action | Click the login button |
| Main Scenario | Admin opens the LMS web portal and the login screen appears |
| Alternate Scenario | N/A |

**Table 3.4: UC-04**

|  |  |
| --- | --- |
| Use case ID | UC-04 |
| Use case Name | Manage Attendance |
| Actors | Admin |
| Purpose | For manage Attendance |
| Priority | High |
| Pre-condition | User must be signup/login. |
| Post-condition | Admin can manage Attendance |
| Actor action | Click the manage Attendance button |
| Main Scenario | Admin opens the LMS web portal and the login screen appears |
| Alternate Scenario | N/A |

**Table 3.5: UC-05**

|  |  |
| --- | --- |
| Use case ID | UC-05 |
| Use case Name | Add Result |
| Actors | Admin |
| Purpose | Check if Result add |
| Priority | High |
| Pre-condition | Access to view Result |
| Post-condition | Task info is open successfully. |
| Actor action | Click the result button |
| Main Scenario | User click on any task the automatically open |
| Alternate Scenario | N/A |

**Table 3.6: UC-06**

|  |  |  |
| --- | --- | --- |
| Use case ID | UC-06 | |
| Use case Name | Manage Result | |
| Actors | Admin | |
| Purpose | Check the Result | |
| Priority | High | |
| Pre-condition | Access to manage Result | |
| Post-condition | Result info is open successfully. | |
| Actor action | Click the Manage Result button | |
| Main Scenario | Admin click on automatically open |
| Alternate Scenario | N/A |

**Table 3.7: UC-07**

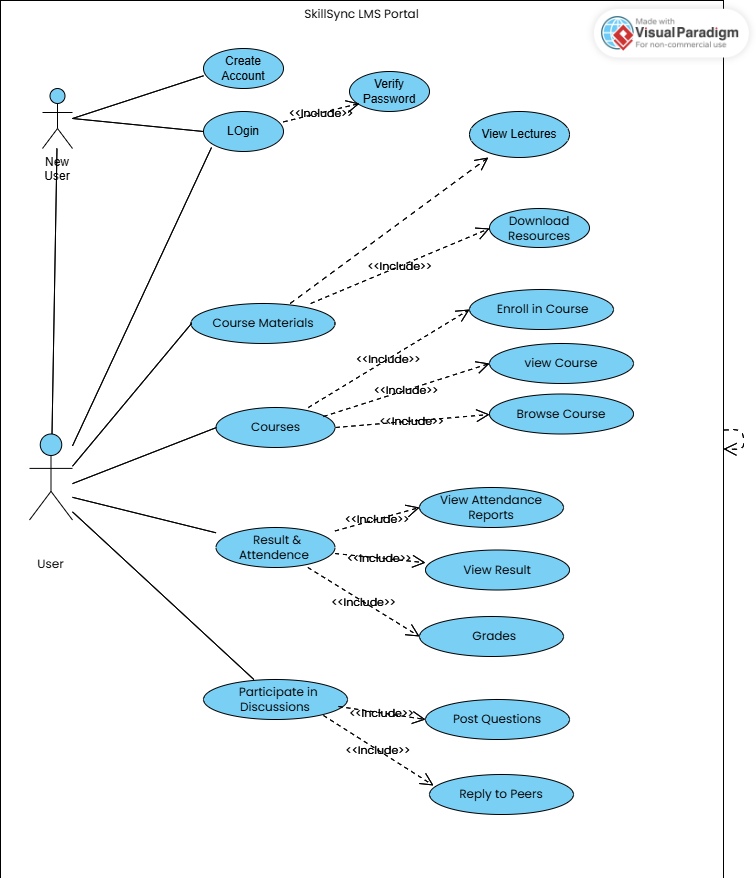
|  |  |
| --- | --- |
| Use case ID | UC-07 |
| Use case Name | Add/Delete Courses |
| Actors | Admin |
| Purpose | Change the number of Courses |
| Priority | High |
| Pre-condition | Access to manage Courses |
| Post-condition | Admin can add or delete Courses. |
| Actor action | Click the add/delete button |
| Main Scenario | Admin can update the Courses data |
| Alternate Scenario | N/A |

**Table 3.8: UC-08**

|  |  |
| --- | --- |
| Use case ID | UC-08 |
| Use case Name | Edit Courses |
| Actors | Admin |
| Purpose | Edit the Courses |
| Priority | High |
| Pre-condition | Access to manage Courses |
| Post-condition | Admin can edit Courses. |
| Actor action | Click the edit Courses button |
| Main Scenario | Admin can update the Courses data |
| Alternate Scenario | N/A |

**Table 3.9: UC-09**

|  |  |
| --- | --- |
| Use case ID | UC-09 |
| Use case Name | View Students Progress |
| Actors | Admin |
| Purpose | Check Students Progress |
| Priority | High |
| Pre-condition | Access to view Students |
| Post-condition | Admin can access Students Progress |
| Actor action | Click the view button |
| Main Scenario | Admin can analyze Students Progress |
| Alternate Scenario | N/A |



**Figure 3.2** LMS Portal Use Case Diagram

**SkillSync Portal use case description**

Figure 3.2 shows the user account management process within the Learning Management System (LMS) portal entails several key interactions. New users initiate their journey by registering for an account, providing essential details such as name, email, and password. Once validated, the system creates their account, granting access to the platform. For existing users, accessing the system involves a straightforward login process, where they input their credentials and are authenticated to enter the portal. Upon successful login, users are directed to the home page, where they navigate through various services and functionalities offered by the LMS. Here, they can explore courses, access resources, and interact with different features. When a user wishes to gather information about a specific course or service, they select the relevant option, and the system provides comprehensive details. Furthermore, users may initiate a quote request for a course by filling out a form with pertinent details such as course name, duration, and participant count. The system then generates a quote based on the provided information for the user's consideration. It's important to note that once logged in, users remain authenticated until they choose to log out, ensuring a seamless experience across interactions within the LMS portal.

**Table 3.10: UC-10**

|  |  |
| --- | --- |
| Use case ID | UC-10 |
| Use case Name | Create Account |
| Actors | New User |
| Purpose | New user have to create account |
| Priority | High |
| Pre-condition | App must be Installed. |
| Post-condition | User just login, and go to the home screen. |
| Actor action | Click the create account button |
| Main Scenario | User open web Home screen appears |
| Alternate Scenario | N/A |

**Table 3.11: UC-11**

|  |  |
| --- | --- |
| Use case ID | UC-11 |
| Use case Name | Login |
| Actors | User |
| Purpose | To use the facilities of of web |
| Priority | High |
| Pre-condition | App must be Installed. |
| Post-condition | User just login, and go to the home screen. |
| Actor action | Click the Login button |
| Main Scenario | User open website Home screen appears |
| Alternate Scenario | N/A |

**Table 3.12: UC-12**

|  |  |
| --- | --- |
| Use case ID | UC-12 |
| Use case Name | Select Courses |
| Actors | User |
| Purpose | Enroll for Courses |
| Priority | High |
| Pre-condition | User must be login to use the web |
| Post-condition | User can submit his request for enrollment |
| Actor action | Click the enroll Courses button |
| Main Scenario | User can select any Courses from the from list |
| Alternate Scenario | N/A |

**Table 3.13: UC-13**

|  |  |
| --- | --- |
| Use case ID | UC-13 |
| Use case Name | Browse Courses |
| Actors | User |
| Purpose | Requesting for Courses for specific category of Courses |
| Priority | High |
| Pre-condition | User must view Courses |
| Post-condition | User can submit his request |
| Actor action | Click the select Courses categories button |
| Main Scenario | User can select any Courses from the from list |
| Alternate Scenario | N/A |

**Table 3.14: UC-14**

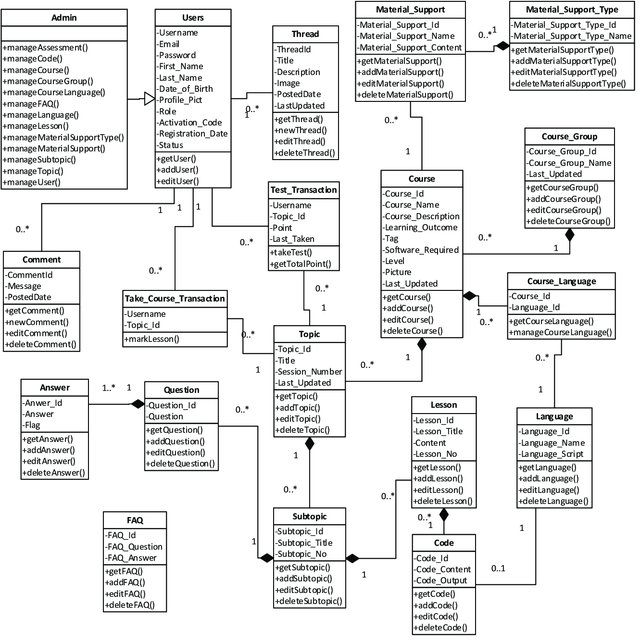
|  |  |
| --- | --- |
| Use case ID | UC-14 |
| Use case Name | Download Recourses |
| Actors | User |
| Purpose | Check if the Recourses downloading is done or not |
| Priority | High |
| Pre-condition | User must enroll in cours |
| Post-condition | User can avail the Resources downloaded |
| Actor action | Click the download button |
| Main Scenario | User is Resources downloaded |
| Alternate Scenario | N/A |

**Table 3.15: UC-15**

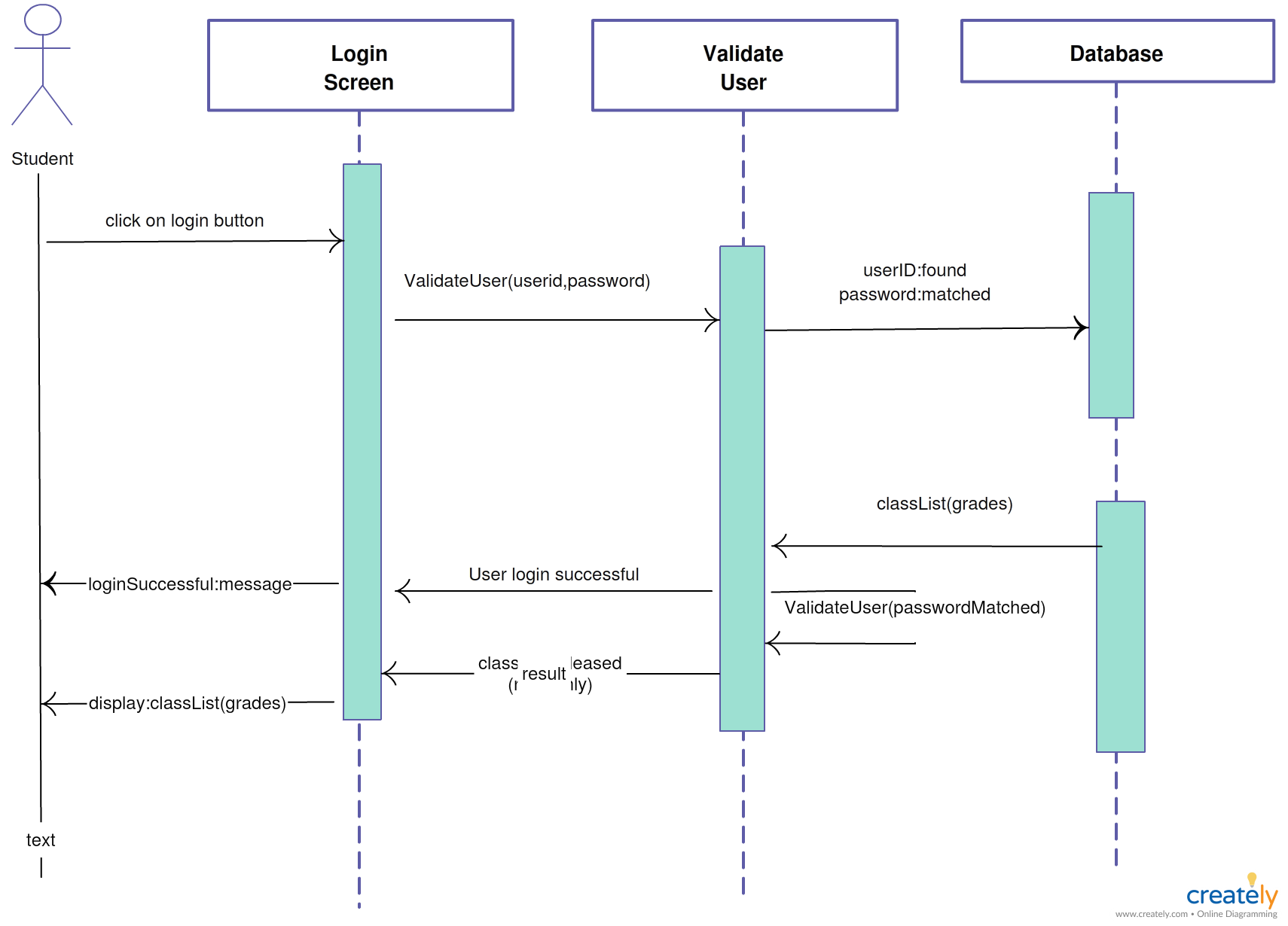
|  |  |
| --- | --- |
| Use case ID | UC-15 |
| Use case Name | Check Result |
| Actors | User |
| Purpose | View the result |
| Priority | High |
| Pre-condition | User must enrolled in the courses |
| Post-condition | User is satisfied |
| Actor action | Click the result button |
| Alternate Scenario | N/A |

**3.4.3 Class Diagram**

A class diagram, a fundamental component of Unified Modeling Language (UML) in software engineering, serves to depict the static structure of a system. This diagram outlines the system's classes, including their attributes and operations (also known as methods), as well as the relationships existing among them.

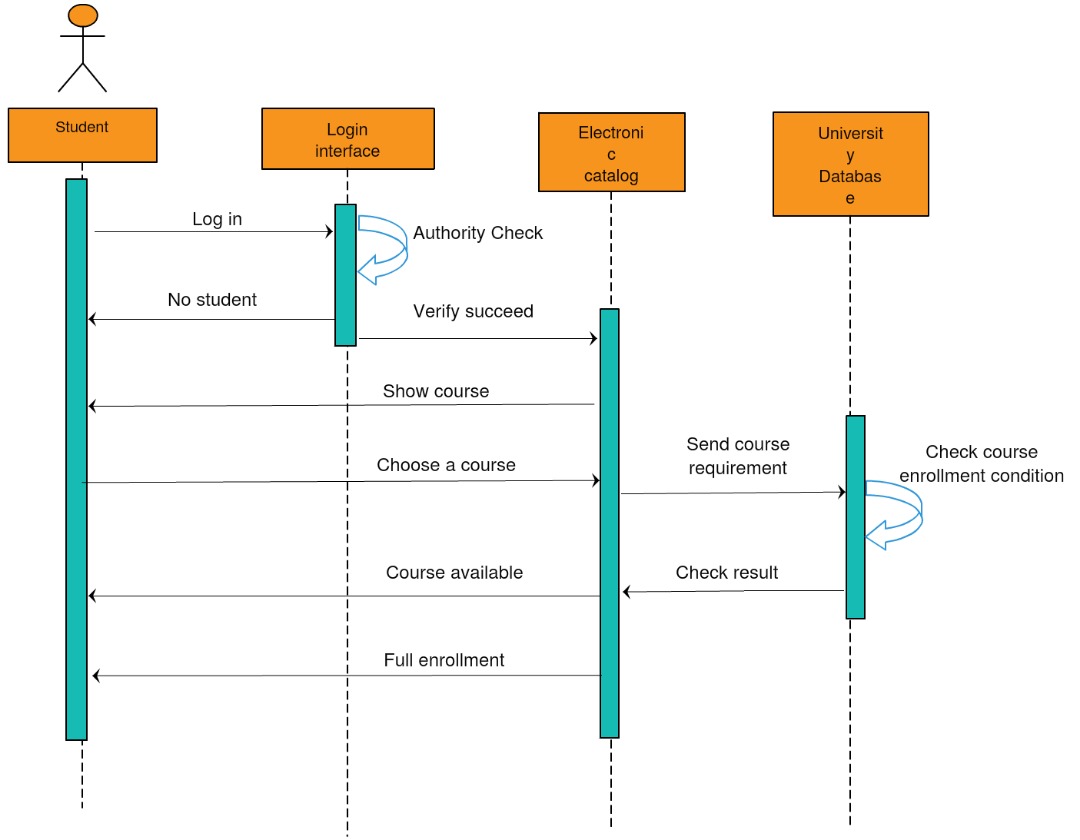


**Figure 3.3** Class diagram



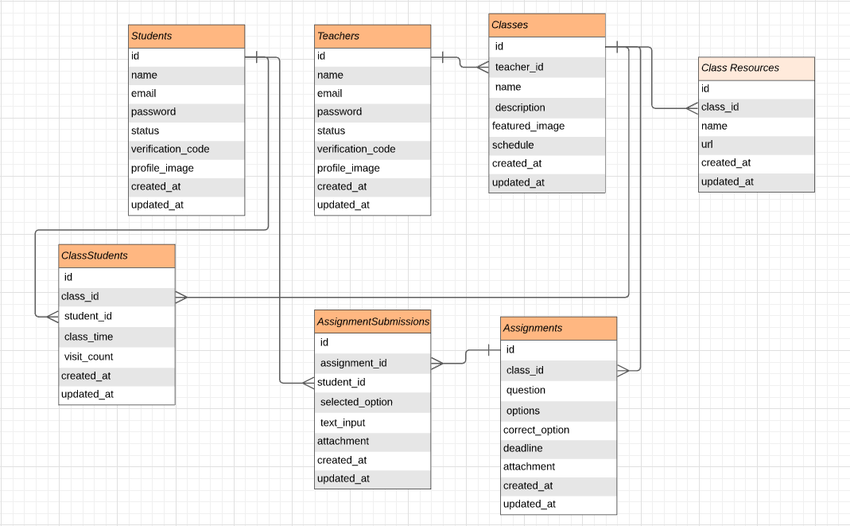
**Figure 3.4** Login Sequence Diagram

The sequence diagram in figure 3.3 illustrates the login process, while here we focus on the sequence of pages within the application. Initially, users log in to the application; if unregistered, they can click on the "register my account" button. This action prompts the appearance of a new registration form. After completing all required fields and clicking the "OK" button, a toast message confirms that "your account is registered," and the data is saved in the database. Subsequently, when attempting to log in again with the same registered account, the database compares the provided data with its stored records. If the data matches successfully, the user gains access to the app; otherwise, a login error is displayed.



**Figure 3.5** Repair Sequence Diagram

The sequence diagram depicted in Figure 3.5 illustrates the process of selecting courses within the application. The sequence of pages begins with the user logging into the website, after which they are directed to the course categories page. Here, users can browse and select the courses they are interested in. Upon selecting a course, the next page allows users to specify additional details such as course duration, format, or instructor preferences. After configuring their course preferences, users can enroll the selected course. Subsequently, the request for quote page appears, presenting a summary of the chosen course(s) along with any specified preferences. When the user submits the request for a quote, the system saves the details into the database. Additionally, all course details, including the selected courses, are stored in the user's course history for future reference.



**Figure 3.5** Entity-Relationship Diagram (ERD)

The figure 3.5 shows The Learning Management System ERD. The ERD shows the entities, its attributes and relationship between them.

**Chapter 5: System Implementation**

Implementation signifies the transformation of conceptual ideas into tangible reality. System implementation, in the context of software development, entails the realization of technical specifications or algorithms as executable programs, software components, or integrated computer systems through the processes of programming and deployment.

The implementation phase encompasses a series of procedures aimed at finalizing the design outlined in the approved systems design document and executing, testing, installing, and commencing the use of the new or revised Information System. The goals of system implementation are as follows:

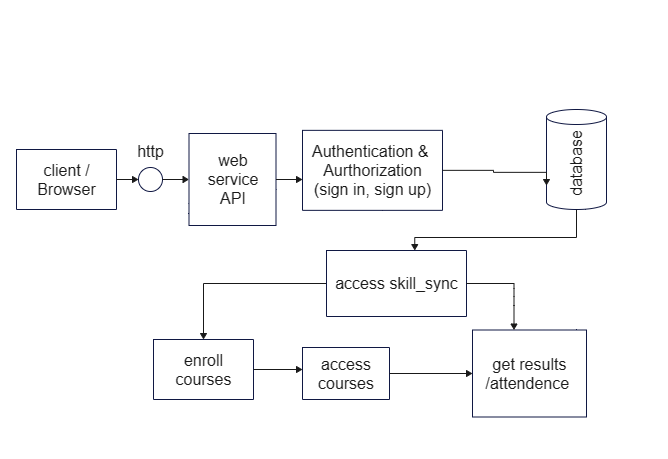
1. **Completing Design Elements:** The implementation process involves finalizing the design elements outlined in the approved systems design document. This includes creating detailed documents, defining computer screen layouts, and establishing database structures.
2. **Programming, Testing, and Documentation:** System implementation requires the development, testing, and documentation of programs and procedures as specified in the approved systems design document. This ensures that the implemented system functions as intended.
3. **User Training and Documentation:** To ensure successful adoption of the new system, comprehensive user manuals and other documentation are prepared. Additionally, personnel within the organization undergo training to operate the new system effectively.
4. **User Acceptance Testing:** Thorough testing of the implemented system with end-users is essential to verify that it meets their requirements and expectations. This ensures that the system aligns with user needs and preferences.
5. **Conversion Planning and Execution:** The implementation phase involves meticulous planning, control, and execution of the conversion process. This includes transitioning from the old system to the new one in an orderly manner, minimizing disruptions and ensuring data integrity throughout the conversion process.

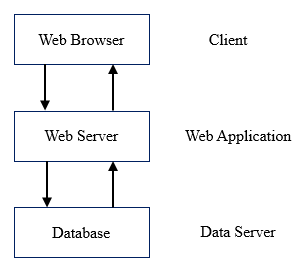
**5.1 System Architecture**

The systems architecture process is where the concepts that will be the backbone of the actual system are developed. It is a conceptual model that describes the structure and behavior of the proposed system or of an existing system. The model could include the technical framework, end user requirements, and a list of system components (hardware and software).

The key decisions that need to be made during the systems architecture process are:

* The attributes of the new system
* The style of architecture
* Type of software used (custom or off-the-shelf)
* Types of technologies used
* How the system will be deployed





**Figure 5.32** System Architecture design

Flowcharts are straightforward diagrams illustrating how the steps of a process are interconnected. Frank Gilbreth, an American engineer, is credited with introducing the concept of a "Process Chart" to the American Society of Mechanical Engineers in 1921, considered the precursor to modern flowcharts. Their simplicity makes flowcharts effective tools for communicating and documenting processes efficiently, increasing the likelihood of correct and consistent application. They aid in estimating the timescale of a process by providing a clear understanding of the time required for each task. Additionally, flowcharts assist in identifying stakeholders to involve at various stages, such as senior management or compliance authorities.

Creating a flowchart offers benefits beyond simply documenting a process. As one constructs it step by step, they can focus on the details of each stage without feeling overwhelmed by the entire process. This allows them to "zoom out" periodically to grasp the broader picture. While individuals often begin by hand-drawing flowcharts, utilizing diagramming applications becomes more convenient for saving, editing, and sharing charts. These applications range from simple and free options like draw.io, creately, and Pencil Project to more sophisticated paid options such as glorify, lucidchart, smartDraw, and Visio.

**Figure 5.33** Basic Flow chart

* We have tried to make our application as simple as we can so users can easily get use of it. Website is very user-friendly with several features.
* It will fulfill all the basic requirements of Students & Teachers.
* It is secure and easily maintainable.

The system shall provide correct results more than 90% of the times. The website offers reliability, easy usability and easy control. Usability is that assesses how easy user interfaces are to use.

**Chapter 7: Conclusions**

The conclusion drawn from this study indicates that expertise in a specific domain enhances overall outcomes. This project was executed on the platform of Visual studio. Additionally, several features were incorporated into the project to enhance its effectiveness. The outlined requirements and specifications guided the project development process. The implementation utilized Visual Studio and Firebase database.

This study aimed to devise a solution for managing course materials by closely observing and interviewing users of the system. An analysis of the existing manual system was conducted to identify the necessary conditions for the proposed system. The resulting system offers a user-friendly interface for various functionalities including course enrollment, check result, view attendance, download resources etc.

The developed system was built using web technologies suitable for an LMS web portal website. Firebase database facilitated the design and management of the database. The coding was primarily done using relevant web development languages such as HTML, CSS, and JavaScript. The system has achieved a stable state with all identified bugs rectified.

Moving forward, the system is poised to operate at a high level of efficiency, benefiting users who are familiar with its features and advantages.