**Live Project – Part 2 – Learning Management System (Waterfall model)**

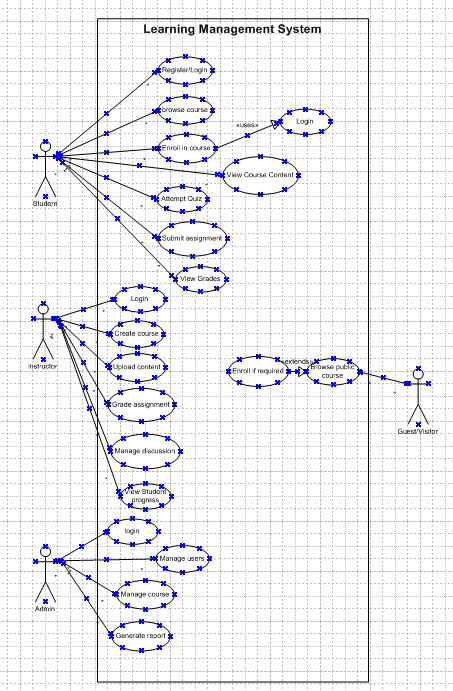
**Document 6- Please prepare a use case diagram, activity diagram and a use case specification document.**

A **Use Case Diagram** is a visual representation used in software engineering and system design to show how users interact with a system to achieve specific goals. It helps in understanding the functional requirements of a system and how different users will use it.

Two Types of use case

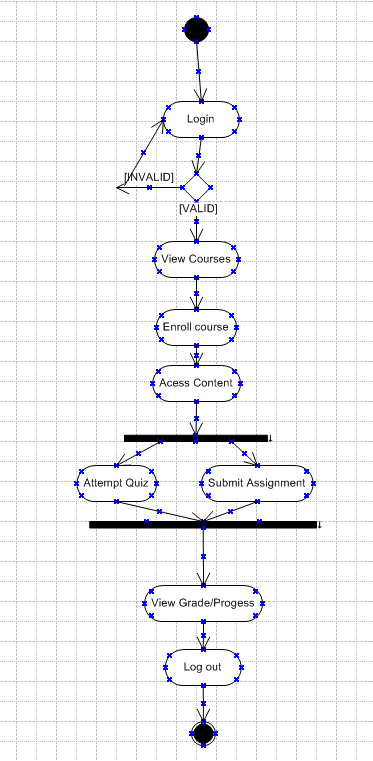
Essential use case: Describes the *core goals* that the system must support — the main actions users take to achieve their primary objectives.

Supporting use case: Describes additional, helpful, or background actions that *support* the essential use case but are not the core goal



Activity diagram

An **Activity Diagram** is a type of diagram used in **Unified Modelling Language (UML)** to represent the flow of activities or actions in a system, process, or workflow. It shows how the system behaves when performing a task and the sequence in which these tasks or activities occur.



Use case Specification Document

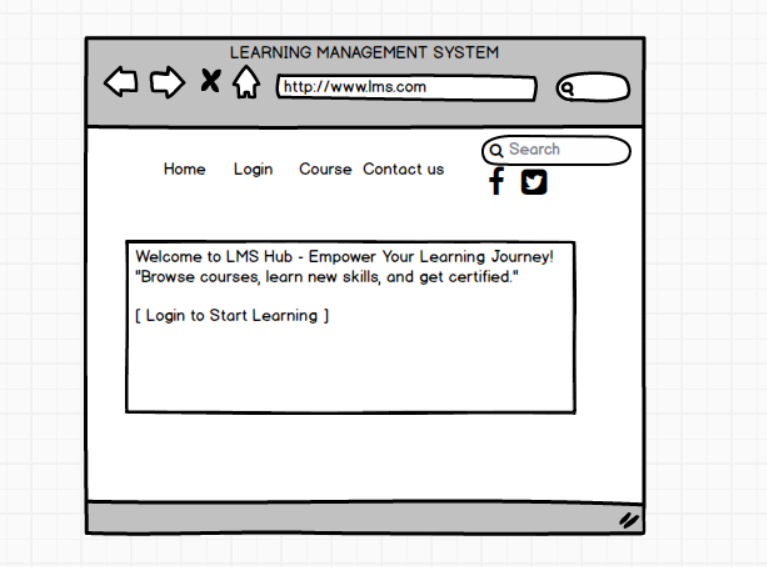
A **Use Case Document** is a detailed description of how **users (actors)** interact with a **system** to achieve a specific goal. It outlines the **steps**, **conditions**, and **interactions** involved in completing a task in the system.

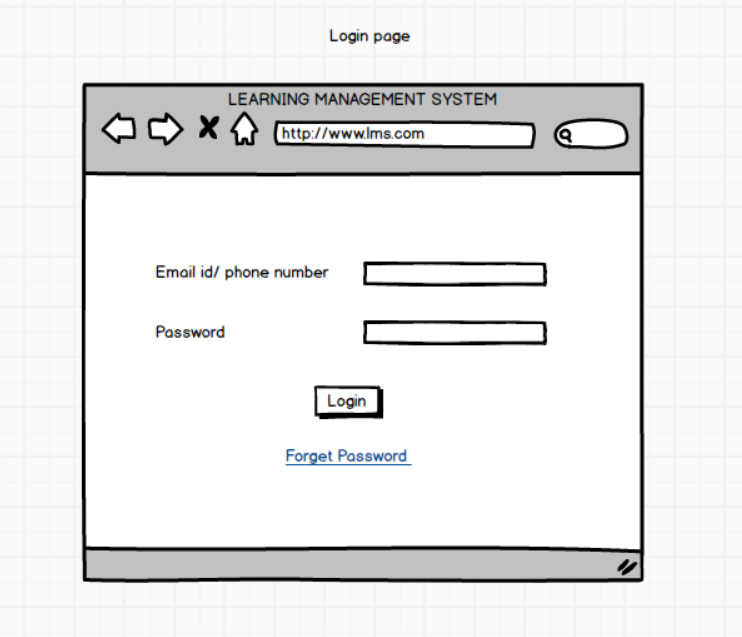
**Purpose of a Use Case Document:**

* To clearly define the functional requirements of the system
* To help developers, testers, and stakeholders understand what the system should do
* Acts as a communication bridge between Business Analysts, Developers, Testers, and Clients

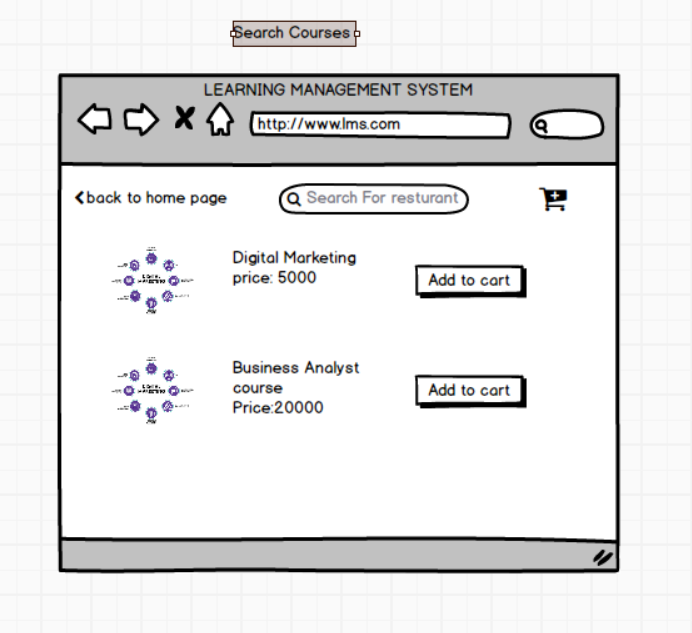
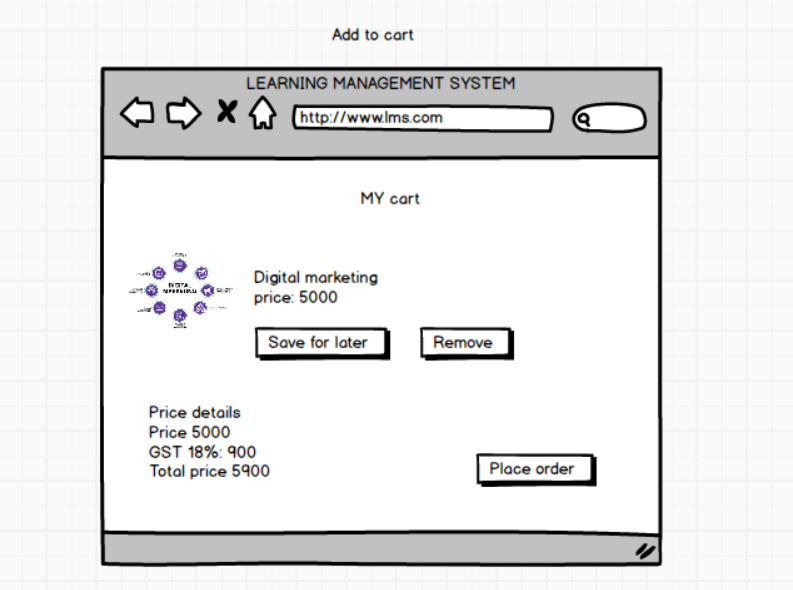
|  |
| --- |
| **Use Case:** Student Course Enrollment |
| **Use Case Description:** This use case describes how a student searches for available courses and enrols in a selected course within the Learning Management System. |
| **Primary Actor:** Student |
| **Secondary Actor:** Admin (For course approval or modification) |
| **Goal:** Allow student to login in the LMS portal |
| **Preconditions:**  The student must be a registered user of the LMS.  Courses must be created and available in the system. |
| **Main Success Scenario:**   1. Student logs into the LMS. 2. The system authenticates the student. 3. Student navigates to the "Courses" section. 4. System displays the list of available courses. 5. Student selects a desired course. 6. System displays the course details. 7. Student clicks "Enroll." 8. System confirms the enrolment and updates the student profile. |
| **Postconditions:**  The student's course list is updated with the newly enrolled course.  System logs the enrolment action |
| **Exceptions:**   * Invalid login credentials—System displays an error. * System error while enrolling—Displays a message and logs the error. |
| **Assumption**  The system is available and operational.  The student meets the enrolment criteria. |
| **Constraint**   * Enrolment limits per course. * Course availability depends on the academic calendar. |
| **Dependencies**  Course modules and content should be uploaded by instructors beforehand.  Payment (if applicable) should be processed before final enrolment. |
| **Inputs and Outputs**   * **Inputs:** Student login credentials, course selection * **Outputs:** Enrolment confirmation, course access granted |
| **Business Rules**   * Students can enrol in courses only during the enrolment period. * Prerequisites must be fulfilled before enrolment. * The number of courses per student may be limited. |
| **Miscellaneous Information**   * Email/SMS confirmation may be sent after successful enrolment. |

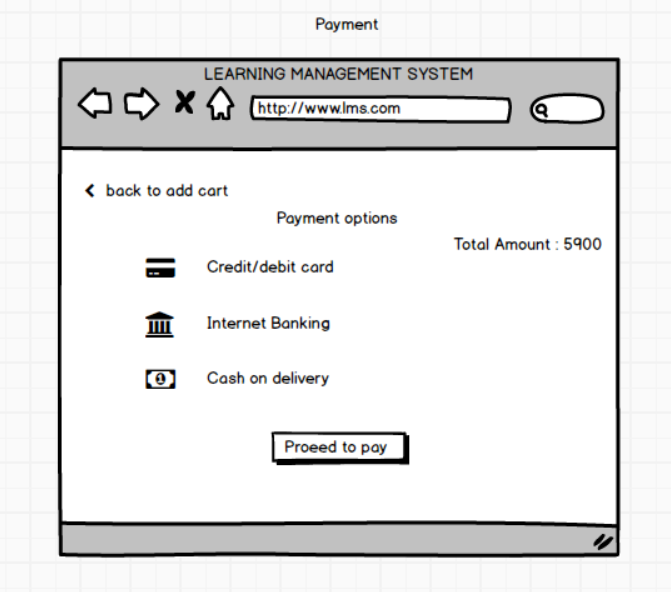
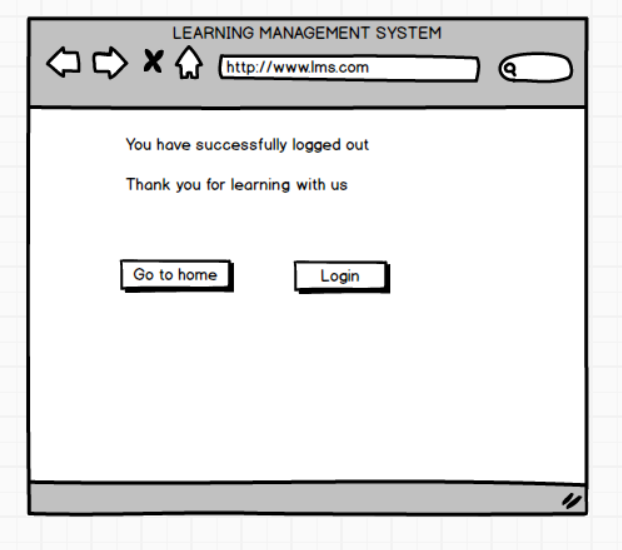
**Document 7- Screens and pages**

****

****

**44**

****

****

**Document 8- Tools-Visio and Axure**

**Visio Experience:**Using Microsoft Visio for the Learning Management System (LMS) project was a valuable experience in creating detailed flowcharts, process diagrams, and system architecture designs. Visio's user-friendly interface made it easy to map out the LMS structure, including the user roles, workflows, and data flow. It helped visualize complex processes such as course enrolment, user authentication, and content management in a clear and professional format. The drag-and-drop functionality and variety of templates allowed me to efficiently create diagrams that improved both communication and understanding of the system design among the project team.

**Axure Experience:**Axure proved to be an excellent tool for developing interactive prototypes for the Learning Management System project. It allowed me to design high-fidelity wireframes that demonstrated the user interface and key functionalities, such as user login, course browsing, and assignment submission. The ability to add dynamic elements and interactive features helped simulate the actual user experience, which was useful during stakeholder presentations and user testing. Axure’s support for conditional logic and adaptive views made it easier to design responsive interfaces, ensuring the LMS prototype was both functional and user-friendly.

**Document 9- BA experience**

**My experience as BA in following phases:**

**Requirement Gathering:**

* Utilized the MOSCOW technique to prioritize requirements as Must have, should have, Could have, and Won’t have.
* Faced client unavailability issues, proactively identified alternate Points of Contact (POCs) to ensure continuous information flow.
* Applied the FURPS model (Functionality, Usability, Reliability, Performance, and Supportability) to validate requirements.
* Identified and removed duplicate or redundant requirements to maintain clarity.
* Used Prototyping techniques to elicit specific and refined requirements from stakeholders.

**Requirement Analysis:**

* Created UML diagrams to visually represent the system's functionalities.
* Designed Activity Diagrams to explain process flows clearly.
* Facilitated communication of diagrams to the team, handled feedback, and incorporated necessary changes.
* Prepared Business Requirements Specification (BRS) and Software Requirements Specification (SRS) documents ensuring clear traceability.

**Design Phase:**

* Derived Test Cases directly from Use Case Diagrams to ensure coverage.
* Maintained constant communication with the client regarding the design and solution approaches.
* Developed both positive and negative test cases to cover all possible scenarios.
* Focused on ensuring no missing test cases to prevent potential risks during development.
* Prepared Test Data for various testing phases.
* Updated the Requirements Traceability Matrix (RTM) ensuring every requirement is mapped to a test case.

**Development Phase:**

* Organized Joint Application Development (JAD) sessions to enhance collaboration between teams.
* Addressed technical team queries promptly during coding.
* Managed conflicts diplomatically during JAD sessions, holding one-on-one discussions to resolve disagreements and maintain a healthy team environment.
* Used diagrams and specifications as references for unit development.
* Regularly coordinated meetings with the technical team and client, ensuring documentation and session recordings for absent members, followed by individual clarifications.

**Testing Phase:**

* Drafted test cases based on Use Cases to ensure functional validation.
* Executed High-Level Testing (HLD) and coordinated with clients for test data needs.
* Updated the RTM post each testing iteration.
* Collected client sign-offs after each phase and prepared the client for User Acceptance Testing (UAT).

**Deployment Phase:**

* Shared the updated RTM as part of the Project Closure Document.
* Facilitated end-user manual preparation and ensured distribution.
* Organized training sessions for end-users and ensured full participation.
* Managed knowledge transfer and proper project hand-off.