

6CS007- Project and Professionalism Assessment

Artifact Design and Test plan

Full Name : Anita Waiba

Student Number : 2331409

Class Group : L6CG8

University Email : [**A.Waiba@wlv.ac.uk**](mailto:A.Waiba@wlv.ac.uk)

Course : BSc (Hons) Computer Science

Supervisor : Niraj Singh

Reader : Yogesh Bikram Shah

Table of Contents

[Table of Figures 3](#_Toc189417638)

[Table of Tables 4](#_Toc189417639)

[1. Artefact 1](#_Toc189417640)

[1.1. Development Methodology: Scrum 1](#_Toc189417641)

[1.2. Wireframing 2](#_Toc189417642)

[1.3. Testing Strategy 2](#_Toc189417643)

[2. Product Backlog 3](#_Toc189417644)

[3. Functional Decomposition Diagram (FDD) 4](#_Toc189417645)

[4. User Management System (UMS) 6](#_Toc189417646)

[Diagrams 7](#_Toc189417647)

[5. Content Delivery System (CDS) 9](#_Toc189417648)

[Diagram 9](#_Toc189417649)

[6. Quiz System (QS) 11](#_Toc189417650)

[Diagrams 11](#_Toc189417651)

[7. Progress Tracking System (PTS) 13](#_Toc189417652)

[Diagrams 13](#_Toc189417653)

[8. Gamification System (GS) 15](#_Toc189417654)

[Diagrams 15](#_Toc189417655)

[9. Notification System (NS) 16](#_Toc189417656)

[Diagrams 17](#_Toc189417657)

[10. Class Diagram 18](#_Toc189417658)

[11. Test Case 18](#_Toc189417659)

[12. Wireframe 19](#_Toc189417660)

[References 24](#_Toc189417661)

# Table of Figures

[Figure 1: Product Backlog (1) 3](#_Toc189416856)

[Figure 2: Product Backlog (2) 4](#_Toc189416857)

[Figure 3: FDD 4](#_Toc189416858)

[Figure 4: UMS Activity (1) 7](#_Toc189416859)

[Figure 5: UMS- Forgot Password Activity (2) 7](#_Toc189416860)

[Figure 6: UMS Sequential 8](#_Toc189416861)

[Figure 7: UMS Use-Case 8](#_Toc189416862)

[Figure 8: CDS Activity (1) 9](#_Toc189416863)

[Figure 9: CDS Activity (2) 10](#_Toc189416864)

[Figure 10: CDS Sequential 10](#_Toc189416865)

[Figure 11: CDS Use-Case 11](#_Toc189416866)

[Figure 12: QS Activity 12](#_Toc189416867)

[Figure 13: QS Sequential 12](#_Toc189416868)

[Figure 14: QS Use-Case 13](#_Toc189416869)

[Figure 15: PTS Activity 14](#_Toc189416870)

[Figure 16: PTS Sequential 14](#_Toc189416871)

[Figure 17: PTR Use-Case 14](#_Toc189416872)

[Figure 18: GS Activity 15](#_Toc189416873)

[Figure 19: GS Sequential 16](#_Toc189416874)

[Figure 20: GS Use-Case 16](#_Toc189416875)

[Figure 21: NS Activity 17](#_Toc189416876)

[Figure 22: NS Sequential 17](#_Toc189416877)

[Figure 23: NS Use-Case 18](#_Toc189416878)

[Figure 24: Class Diagram of LearnIT 18](#_Toc189416879)

[Figure 25: A test case for LearnIT 19](#_Toc189416880)

[Figure 26: Sign Up page 20](#_Toc189416881)

[Figure 27: Profile Page 20](#_Toc189416882)

[Figure 28: Quiz Page 20](#_Toc189416883)

[Figure 29: Quests 21](#_Toc189416884)

[Figure 30: Opening Page 21](#_Toc189416885)

[Figure 31: Leaderboard 21](#_Toc189416886)

[Figure 32: Knowledge Enquiry Page 22](#_Toc189416887)

[Figure 33: Home Page 22](#_Toc189416888)

[Figure 34: Progress 22](#_Toc189416889)

[Figure 35: Login 23](#_Toc189416890)

# Table of Tables

[Table 1: UMS 6](#_Toc189416926)

[Table 2: CDS 9](#_Toc189416927)

[Table 3: QS 11](#_Toc189416928)

[Table 4: PTS 13](#_Toc189416929)

[Table 5: GS 15](#_Toc189416930)

[Table 6: NS 16](#_Toc189416931)

# Artefact

The term "artifact" in software development describes essential delivery products that comprise diagrams and scripts along with models and documents that guide software development toward structured completion. The artifacts function to deliver precision between planned objectives and actual development. A development project executed without artifacts would become equivalent to building a structure with no architectural plans. The project guides multiple team members including developers and stakeholders as well as designers throughout the development phases. (Méndez Fernández, 2019)

## Development Methodology: Scrum

This project development follows Scrum as its chosen subset within the Agile methodology. Scrum operates as a flexible change-based framework that helps teams achieve better results through cooperative work methods and continuous adjustment systems. Scrum has been chosen as the development methodology over Waterfall and DevOps and Rapid Application Development (RAD) because it offers flexible adaptability with incremental working software releases through its sprint cycles. (Morandini, 2021)

Scrum consists of:

* **Sprints:** Sprints' strategy incorporates brief development periods extending from one week up to four weeks to perform targeted task completion.
* Product Backlog maintains a prioritized schedule containing planned features, enhancement work, and corrective measures for development.
* Sprint Planning includes a team conference that selects backlog items for work within the sprint.
* Daily meetings run for brief periods to maintain progress status and identify upcoming obstacles and upcoming moves.
* Sprint Review: A demonstration of completed work at the end of each sprint.
* During Sprint Retrospective the team conducts a review which helps identify achievements and requirements for enhancement during the upcoming sprint. (Morandini, 2021)

Using Scrum ensures that:

* The development process operates iteratively to allow continuous flexibility and numerous adjustments.
* Team members get consistent feedback from all users and stakeholders.
* Regular testing combined with repeated development cycles helps applications maintain their high-quality standards.
* The synergy improves between developers and testers as well as product owners.

## Wireframing

A wireframe stands as an important fundamental component for designing a user interface (UI) before starting any development process. Wireframes offer developers and designers an artistic design structure to help them establish:

* Navigation flow for seamless user interaction.
* A strategic arrangement of components exists for maximizing user experience (UX).
* The system distributes content in suitable arrangements that build natural learning sequences.

The preliminary development of wireframes at the beginning of a project enables quick alteration opportunities to streamline the process moving toward actual code creation which lowers both development expenses and time requirements for modifications made in late design phases.

## Testing Strategy

A testing strategy with Black Box Testing and White Box Testing methods will be used to guarantee proper function alongside user requirements fulfillment.

* Testing the application's functionality through Black Box Testing occurs without investigating its internal code structure. The testing approach confirms that the system operates according to user expectations. (I., 2003)
* White Box Testing examines the system's internal code to investigate logic structures while searching for hidden bugs and potential security flaws. Such measures guarantee that system performance reaches its optimal state at the code level. (I., 2003)

The development process will become efficient collaborative and adjustable to user needs when Scrum methodology is combined with wireframing and robust testing across all stages of development. These project artifacts will direct the delivery of an excellent interactive learning platform that engages users effectively.

# Product Backlog

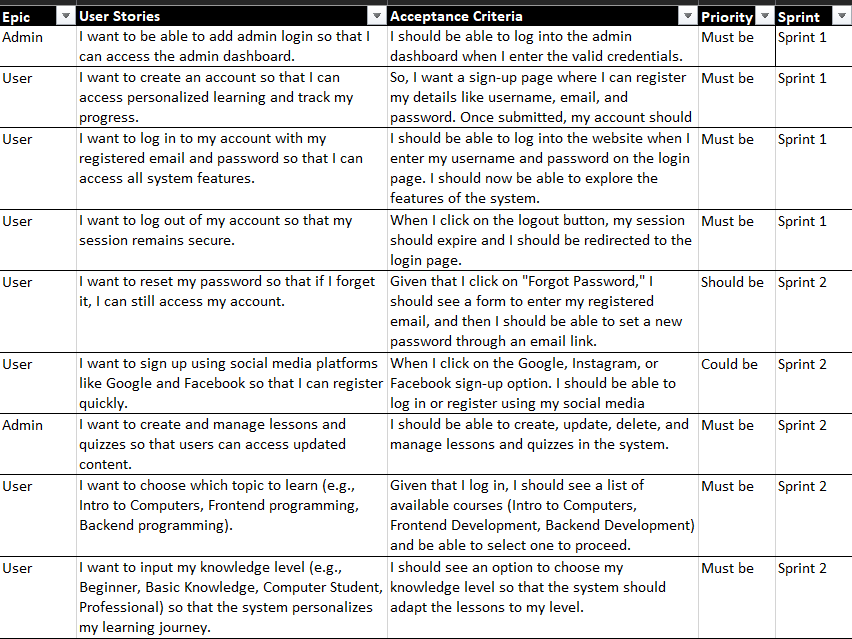


Figure 1: Product Backlog (1)

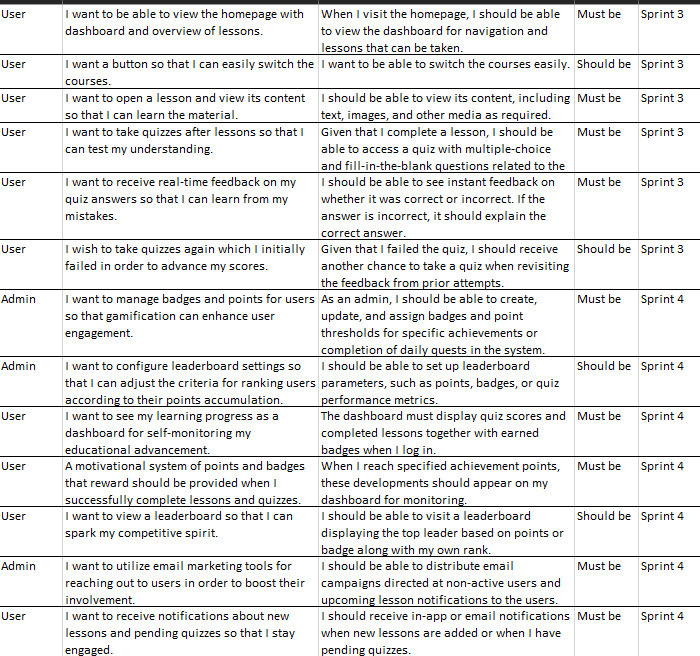


Figure 2: Product Backlog (2)

# Functional Decomposition Diagram (FDD)

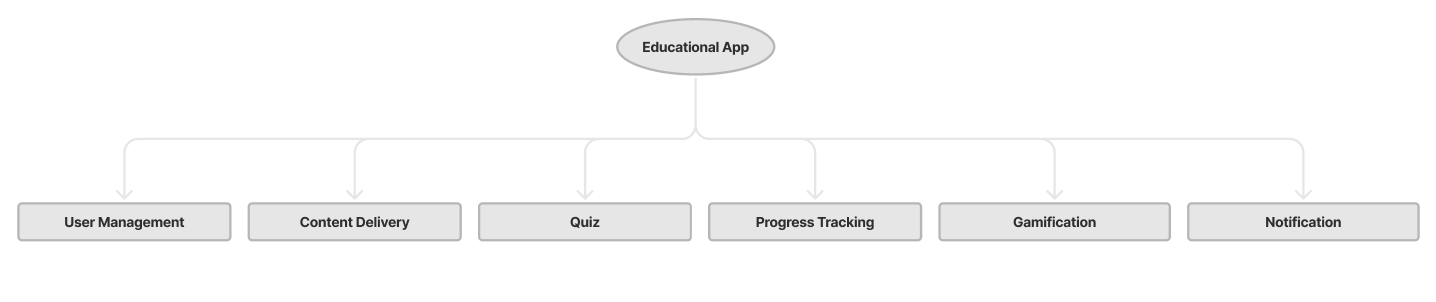


Figure 3: FDD

**Legend**

UMS – User Management System

CDS – Content Delivery System

GS – Gamification System

PTS – Progress Tracking System

QS – Quiz System

NS – Notifications System

**Requirement Types:**

F – Functional Requirement

NF – Non-Functional Requirement

U – Usability Requirement

# User Management System (UMS)

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. Code** | **Requirement Description** | **Use Case** | **Moscow Prioritization** |
| UMS-F-1.0 | The system allows the user to create a new account and log in. | User Login and Sign Up. | Must have |
| UMS-F-1.0.1 | The system allows the user to log in or sign up through a social media account (Facebook, Instagram, or Google account). | User Login | Could have |
| UMS-NF-1.1 | The sign-up form information should be filled in by the user and should be encrypted for security purposes. |  | Must have |
| UMS-UR1.2 | For a strong password, the user must enter a password that is a combination of alphabets, numbers, and special characters. |  | Could have |
| UMS-F-2.0 | The user should assist the user in resetting the password in case the user forgets their password. | Reset Password | Should have |
| UMS-F-2.0 | After the password has been change successfully, an alert message should be displayed. | Alert notification | Should have |
| UMS-F-3.0 | After logging in, the user should be able to view the home page and lessons. | View lessons | Must have |

Table 1: UMS

## Diagrams

1. Activity Diagrams:

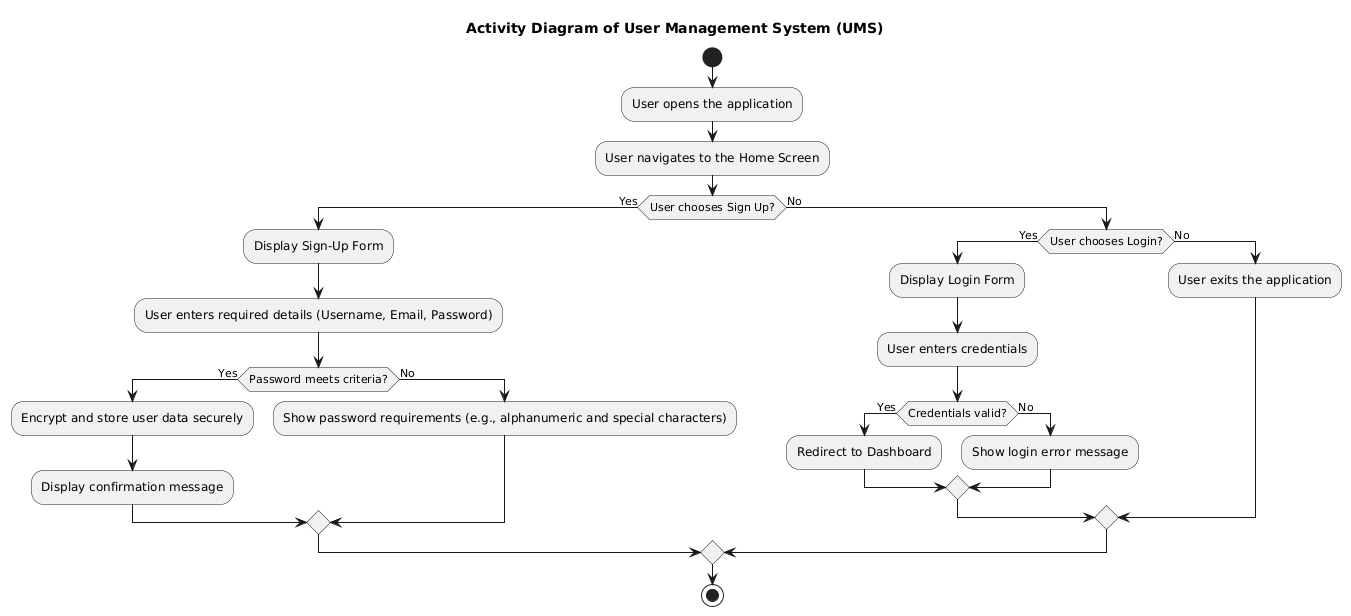


Figure 4: UMS Activity (1)

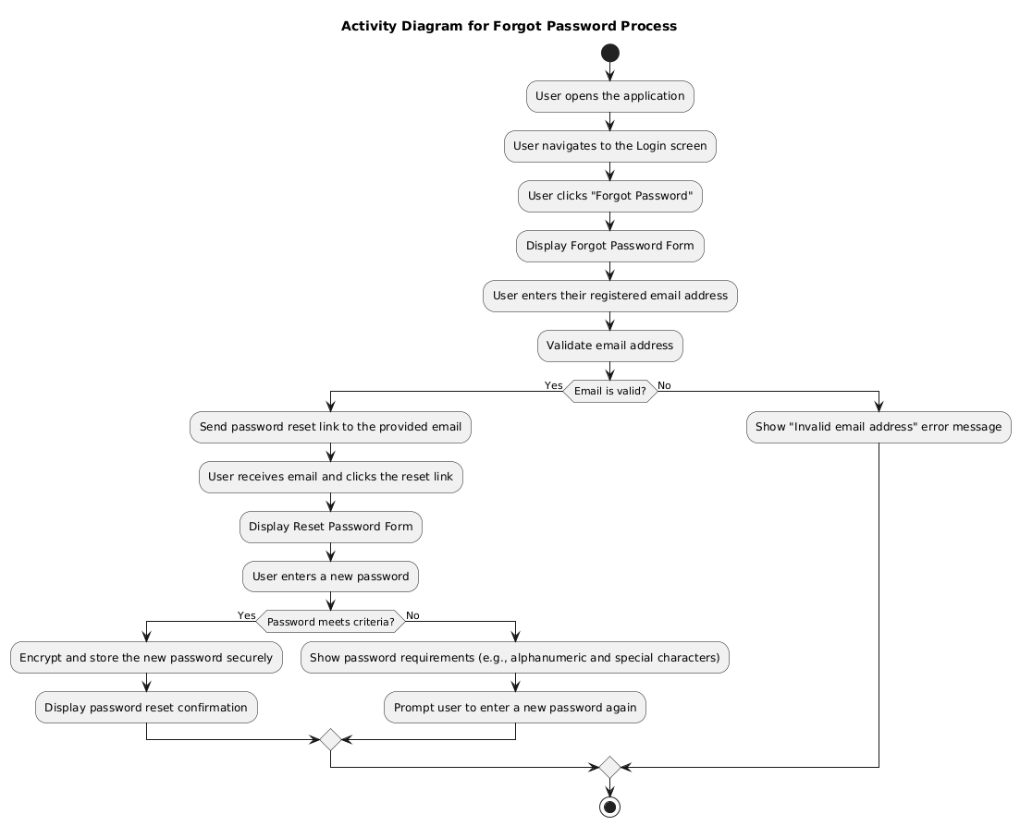


Figure 5: UMS- Forgot Password Activity (2)

1. Sequential Diagram

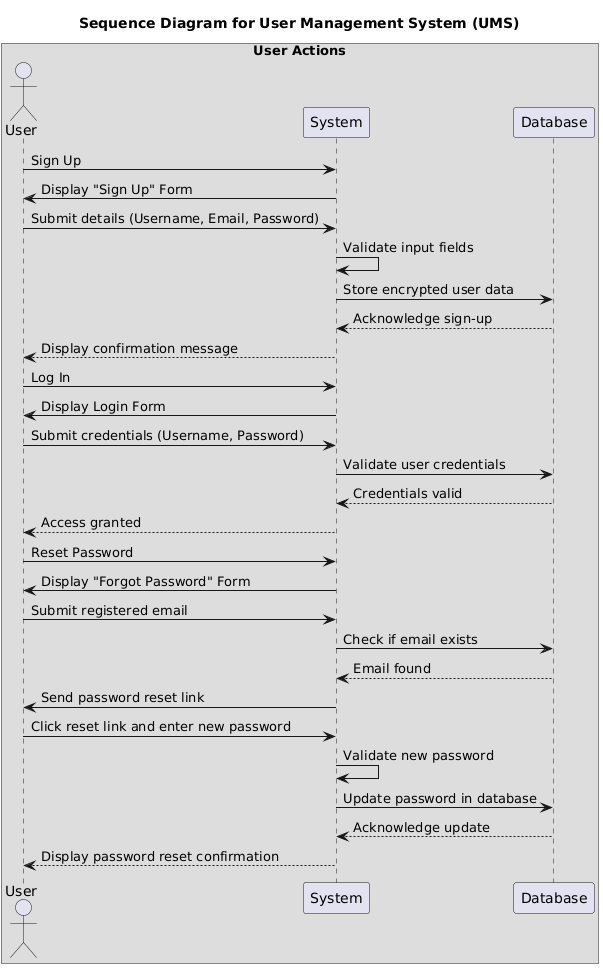


Figure 6: UMS Sequential

1. Use-case Diagram

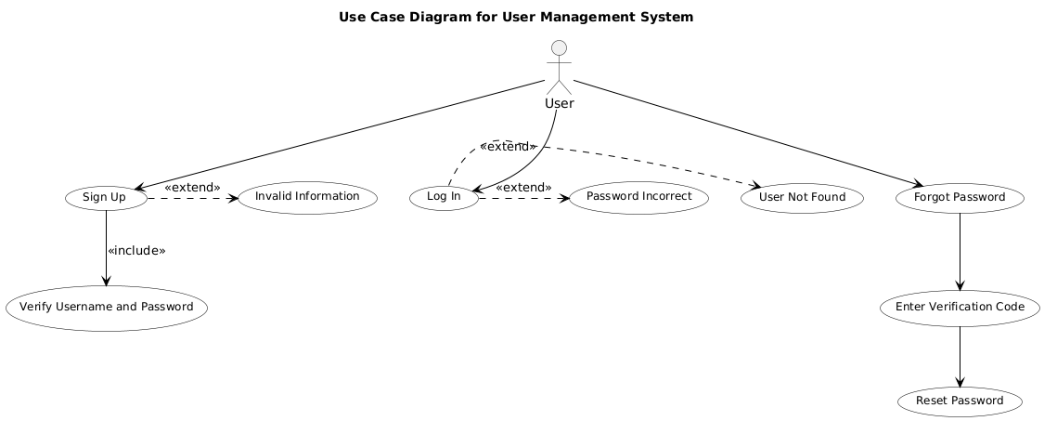


Figure 7: UMS Use-Case

# Content Delivery System (CDS)

|  |  |  |  |
| --- | --- | --- | --- |
| Req. Code | Requirement Description | Use Case | Moscow Prioritization |
| CDS-F-1.0 | The system allows the user to view and choose from the available topics (e.g., Intro to Computers, Frontend, Backend). | View and Choose Courses | Must have |
| CDS-F-1.1 | User can input their knowledge for personalized lessons (Beginner, Basic, Student, or Professional). | Personalize Lessons | Must have |
| CDS-F-1.2 | The system allows users to open a lesson and view its content (text, images, media). | View Lessons Content | Must have |

Table 2: CDS

## Diagram

1. Activity Diagram

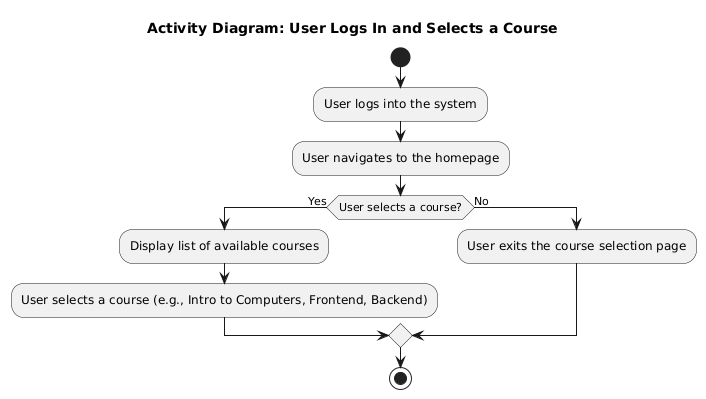


Figure 8: CDS Activity (1)

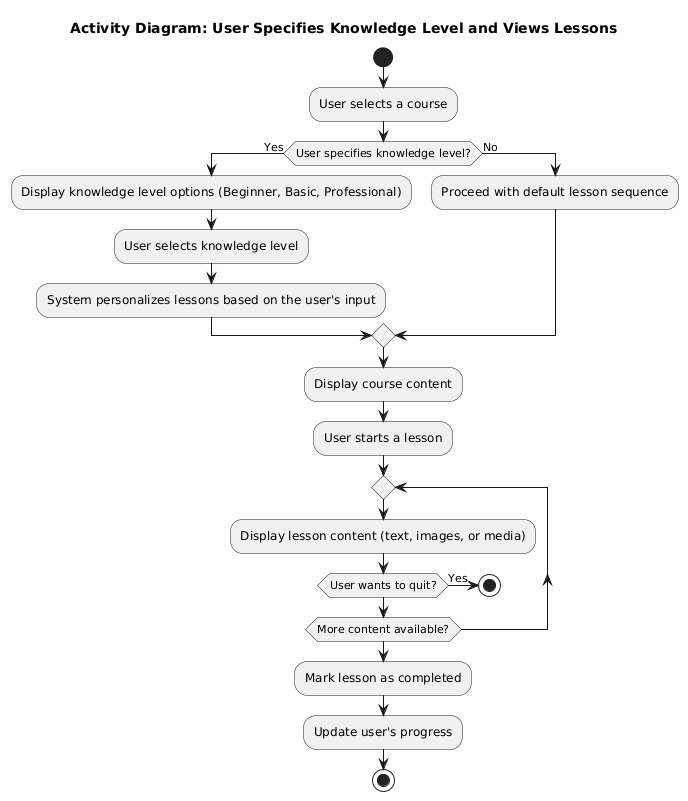


Figure 9: CDS Activity (2)

1. Sequential Diagram

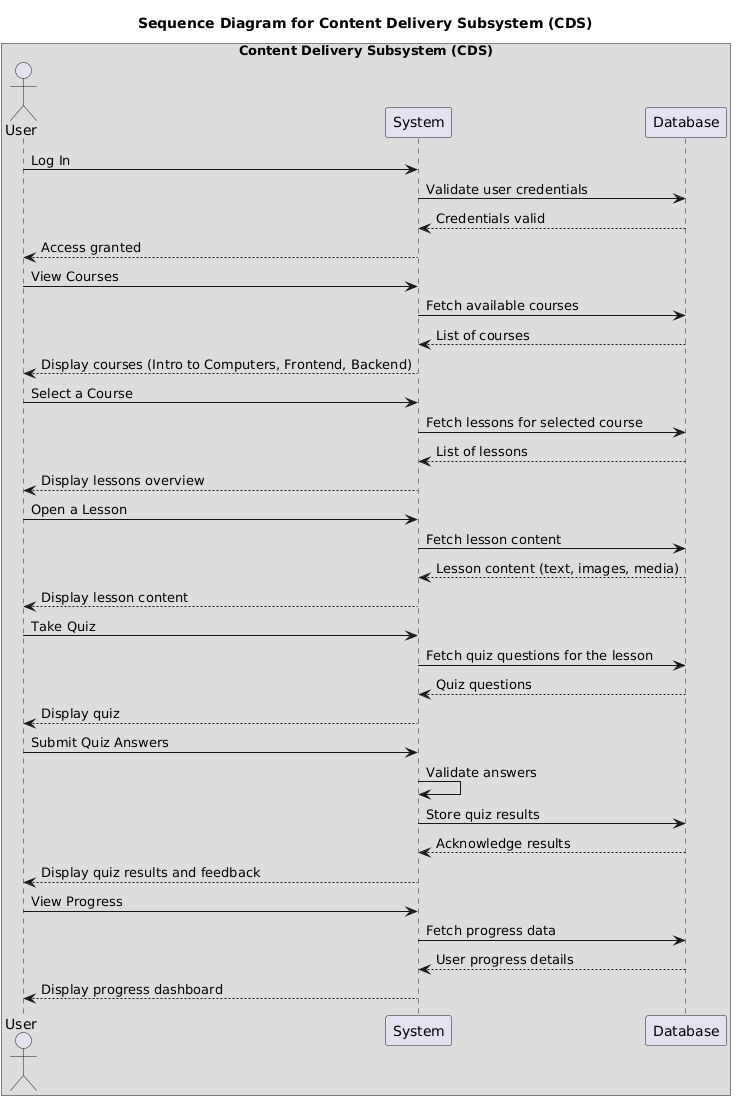


Figure 10: CDS Sequential

1. Use-Case Diagram

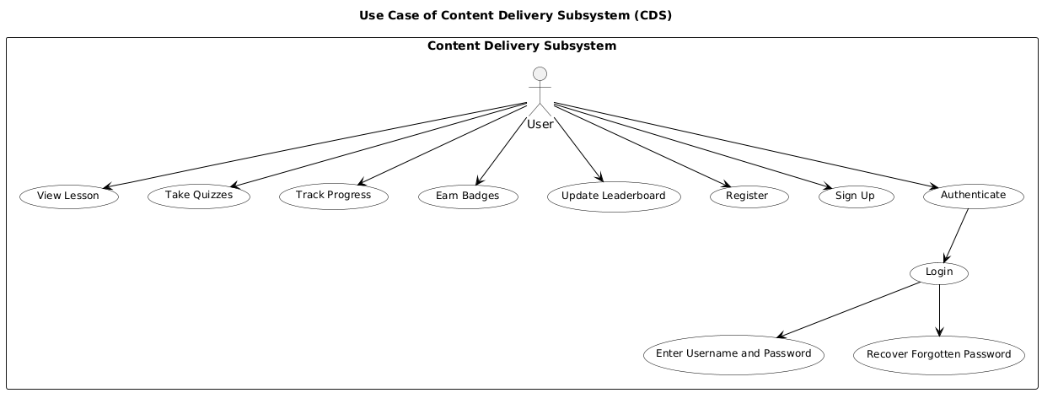


Figure 11: CDS Use-Case

# Quiz System (QS)

|  |  |  |  |
| --- | --- | --- | --- |
| Req. Code | Requirement Description | Use Case | Moscow Prioritization |
| QS-F-1.0 | |  | | --- | |  |  |  | | --- | | The system allows users to give quizzes after lessons to test understanding. | | Take Quizzes | Must have |
| QS-F-1.1 | The system provides real-time feedback on quiz answers to help the user improve their understanding and learn from mistakes. | Real-Time Feedbacks | Must have |
| QS-F-1.2 | Users can retake the failed quiz to correct their mistakes and improve more. | Retake Failed Quiz | Must have |

Table 3: QS

## Diagrams

1. Activity Diagrams

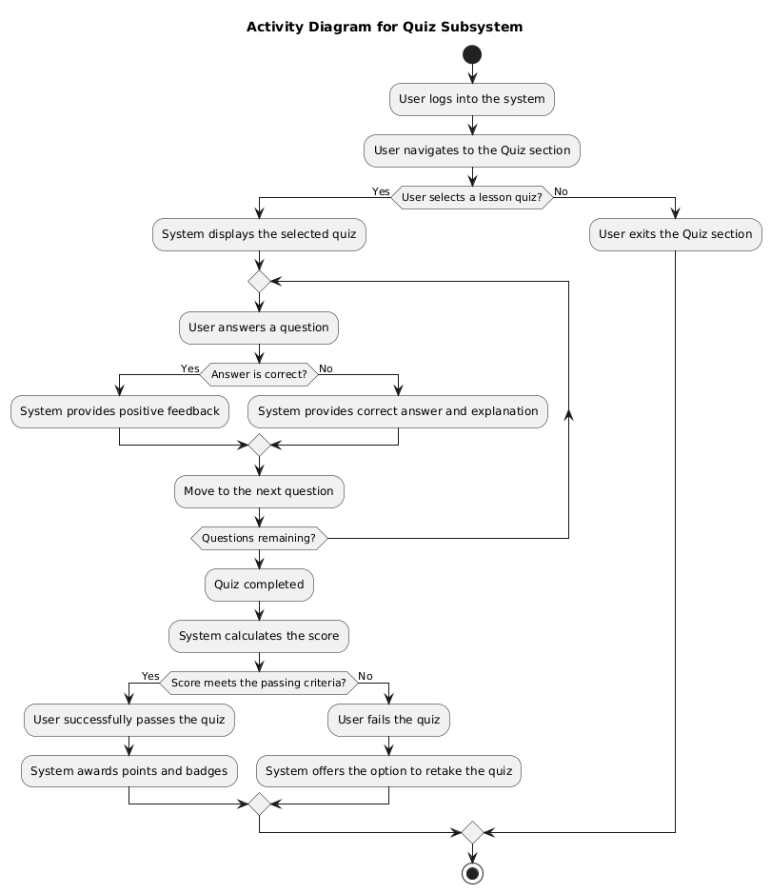


Figure 12: QS Activity

1. Sequential Diagram

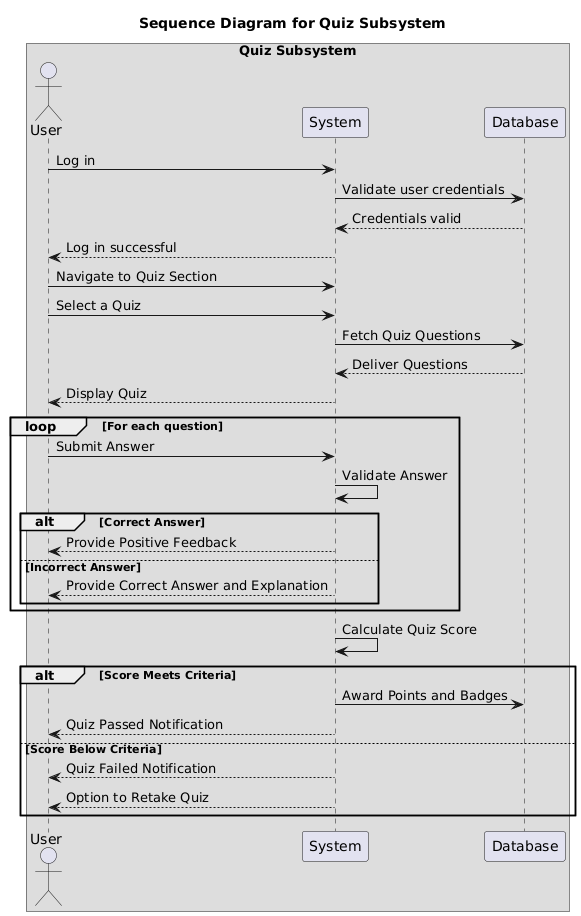


Figure 13: QS Sequential

1. Use-Case Diagram

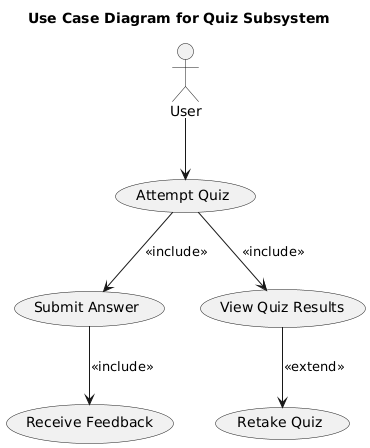


Figure 14: QS Use-Case

# Progress Tracking System (PTS)

|  |  |  |  |
| --- | --- | --- | --- |
| Req. Code | Requirement Description | Use Case | Moscow Prioritization |
| PTS-F-1.0 | |  | | --- | |  |  |  | | --- | | The system allows the user to track their progress in the dashboard. As the user completes a lesson a tick mark appears. | | Track Progress | Must have |
| PTS-F-1.1 | The system generates reports showing completed lessons and quiz results. | Generate Progress Reports | Should have |

Table 4: PTS

## Diagrams

1. Activity Diagram

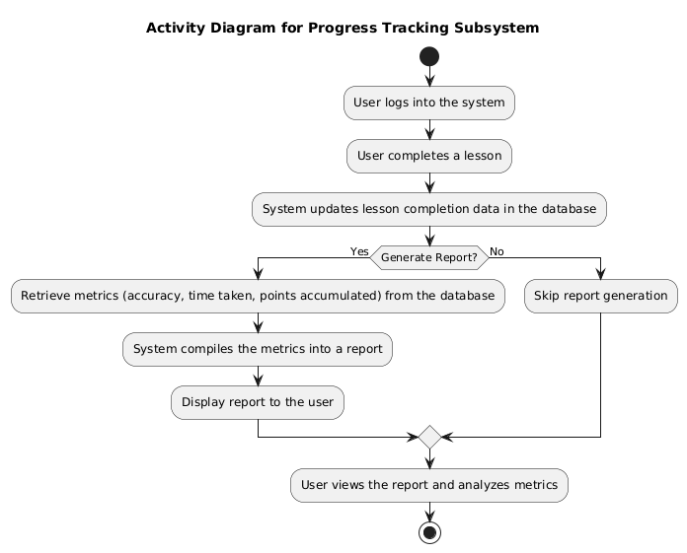


Figure 15: PTS Activity

1. Sequential Diagram

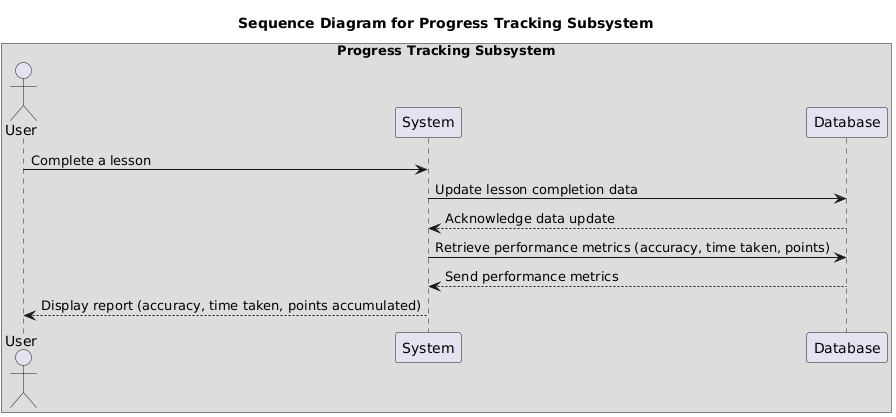


Figure 16: PTS Sequential

1. Use-case Diagram

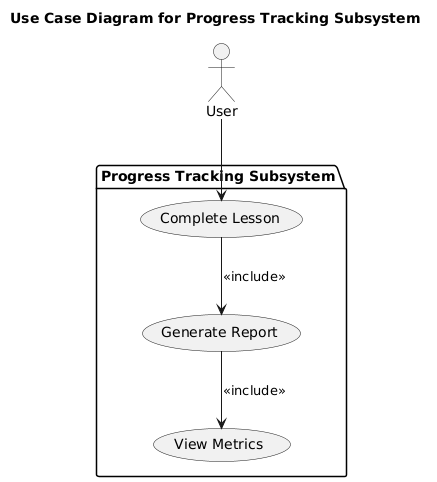


Figure 17: PTR Use-Case

# Gamification System (GS)

|  |  |  |  |
| --- | --- | --- | --- |
| Req. Code | Requirement Description | Use Case | Moscow Prioritization |
| GS-F-1.0 | The system gives the user badges and points when they complete the lessons. | Award Badges and Points | Must have |
| GS-F-1.1 | The system allows the users to view a leaderboard to track their ranking and motivate them through competition. | Personalize Lessons | Should have |

Table 5: GS

## Diagrams

1. Activity Diagram

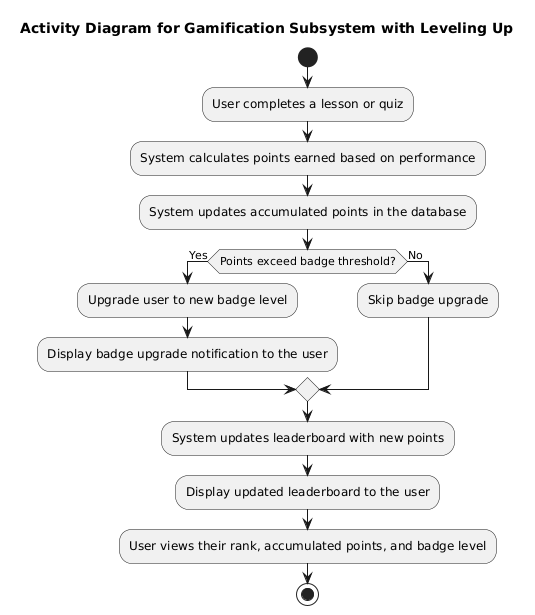


Figure 18: GS Activity

1. Sequential Diagram

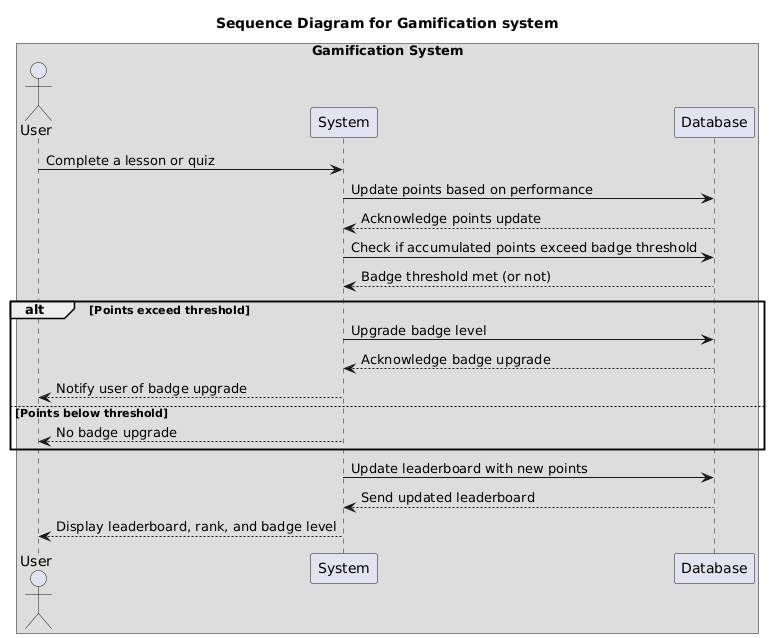


Figure 19: GS Sequential

1. Use-case Diagram

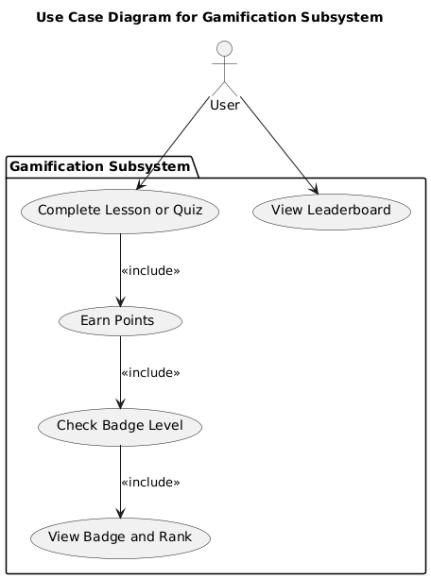


Figure 20: GS Use-Case

# Notification System (NS)

|  |  |  |  |
| --- | --- | --- | --- |
| Req. Code | Requirement Description | Use Case | Moscow Prioritization |
| NS-F-1.0 | The system sends notifications to the user about the new lessons or pending quizzes to stay engaged. | Notifications | Must have |

Table 6: NS

## Diagrams

1. Activity Diagram

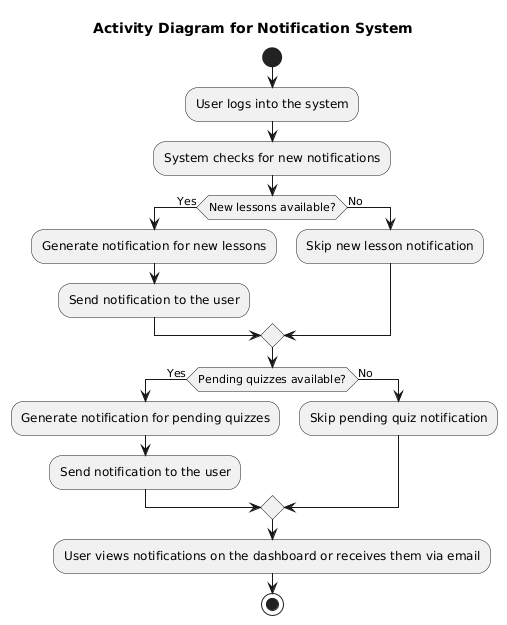


Figure 21: NS Activity

1. Sequential Diagram

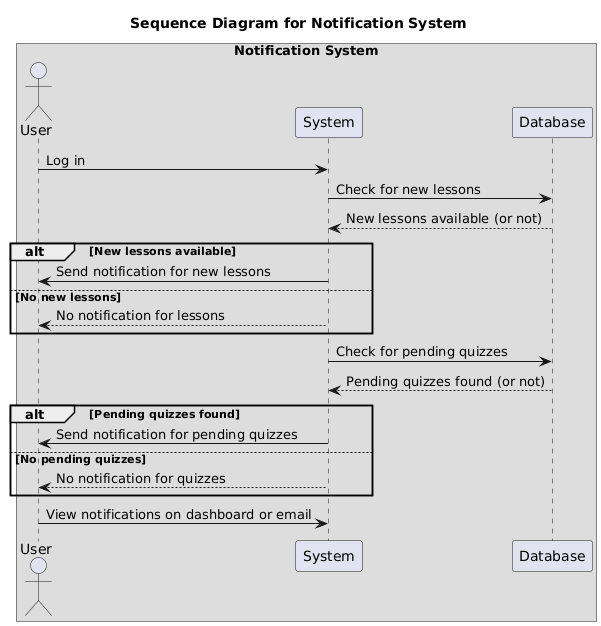


Figure 22: NS Sequential

1. Use-case Diagram

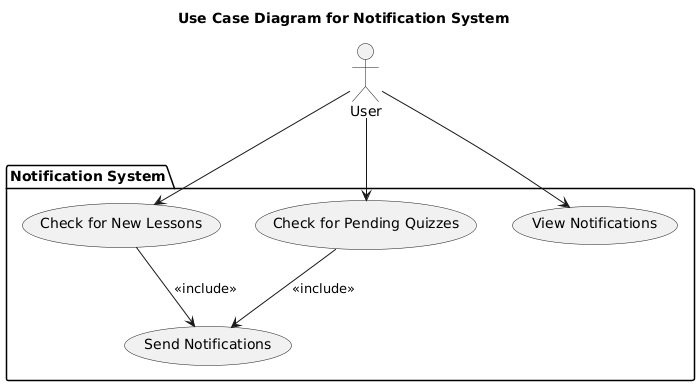


Figure 23: NS Use-Case

# Class Diagram

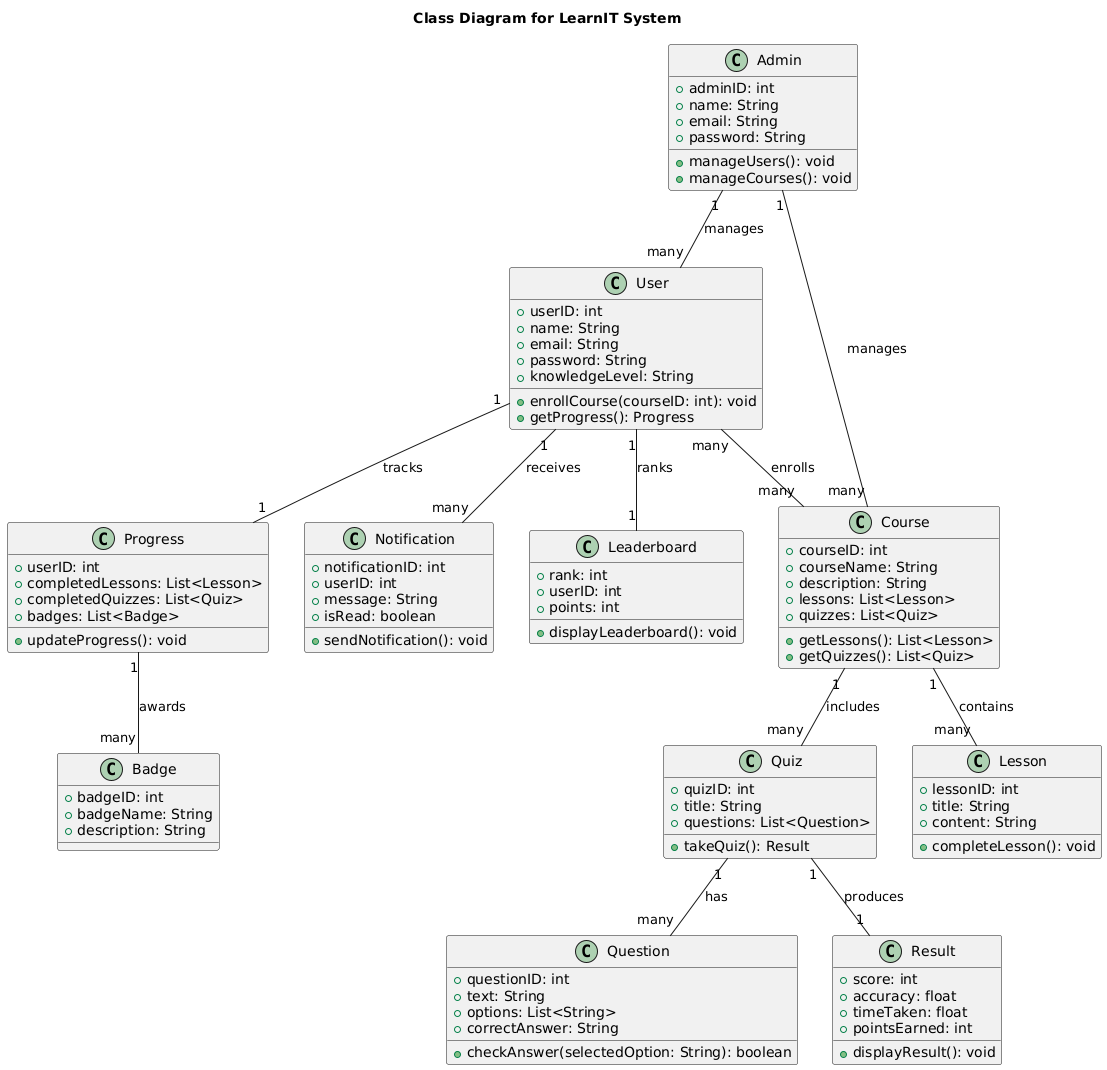


Figure 24: Class Diagram of LearnIT

# Test Case

1. Black Box Testing

The testing methodology does not reveal the system's logical structure or code design information to testers. During testing activities, testers evaluate the application inputs and outputs as they neglect to explore its underlying process. (Myers, 2020)

1. White Box Testing

The tester gains complete access to understand internal system design elements when following this testing approach. During testing the internal program operations and logical code structure are checked for verification. (I., 2003)

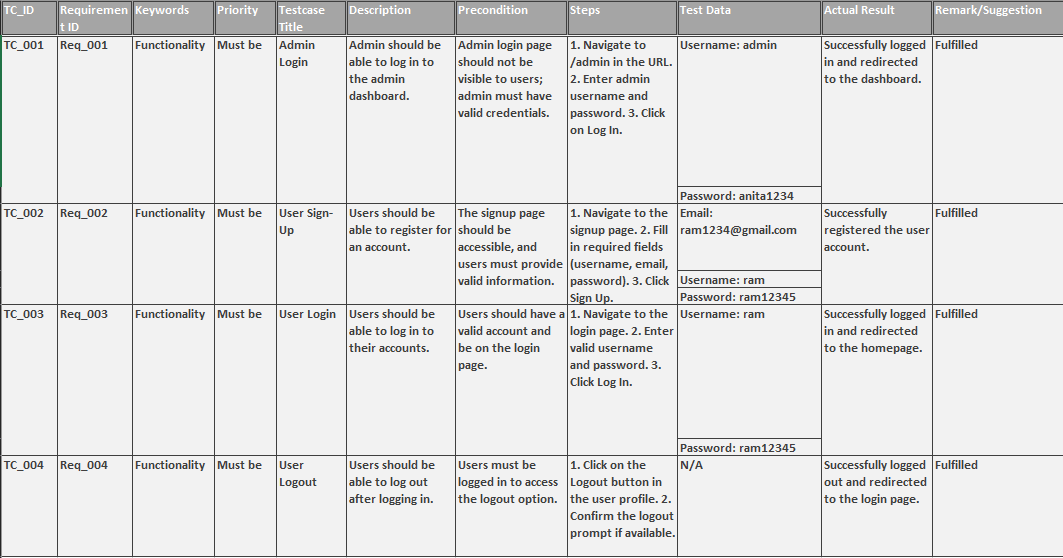


Figure 25: Test case for LearnIT

# Wireframe

Link:<https://www.figma.com/design/rAvGxRU4zXgAZ1utATdpjW/LearnIT?t=gYbnhp0t6Y5tHmaF-1>

Pictures:

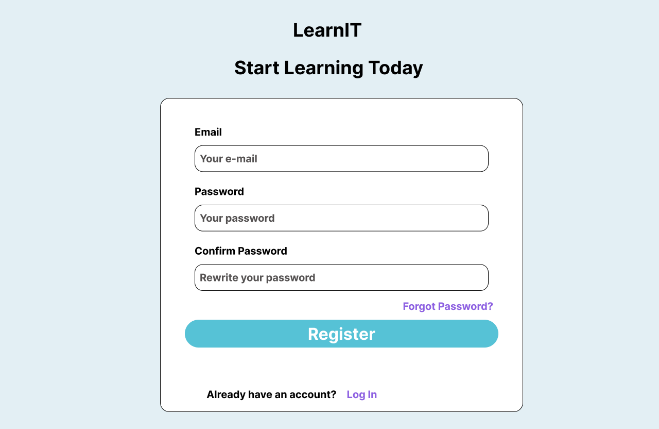


Figure 26: Sign Up page

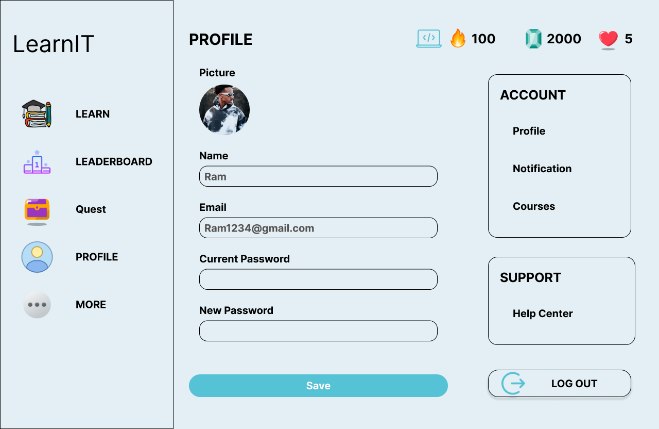


Figure 27: Profile Page

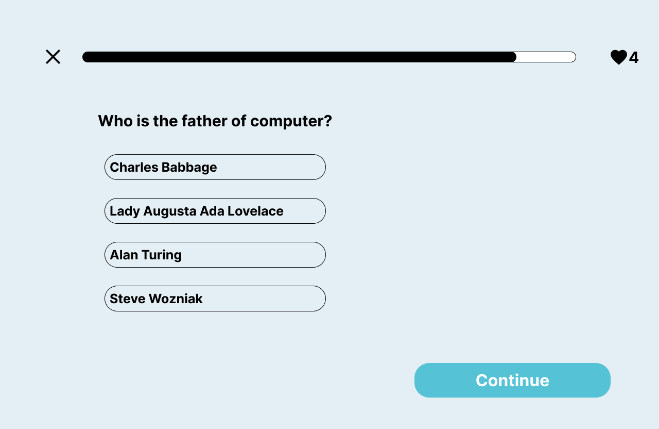


Figure 28: Quiz Page

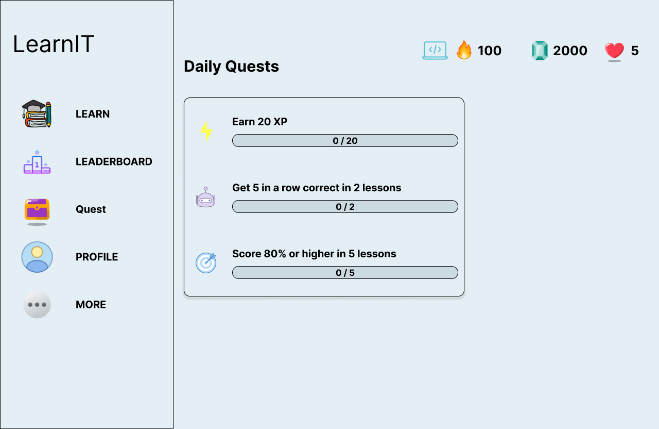


Figure 29: Quests

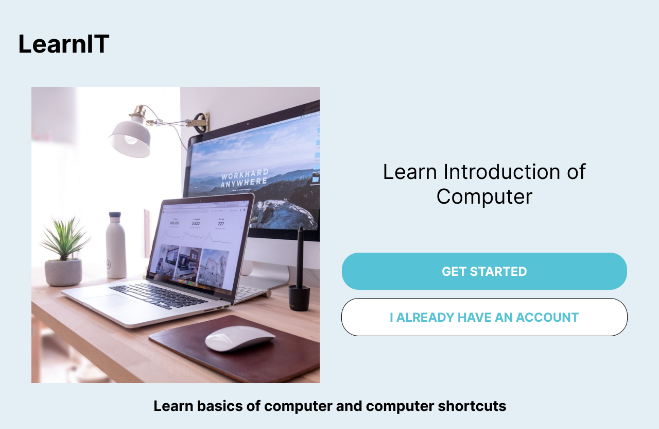


Figure 30: Opening Page

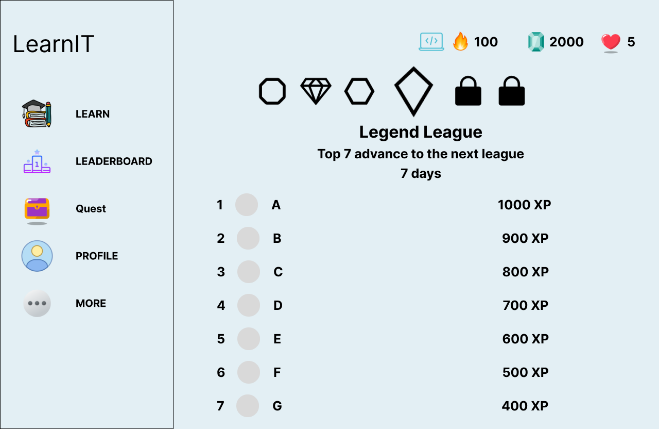


Figure 31: Leaderboard

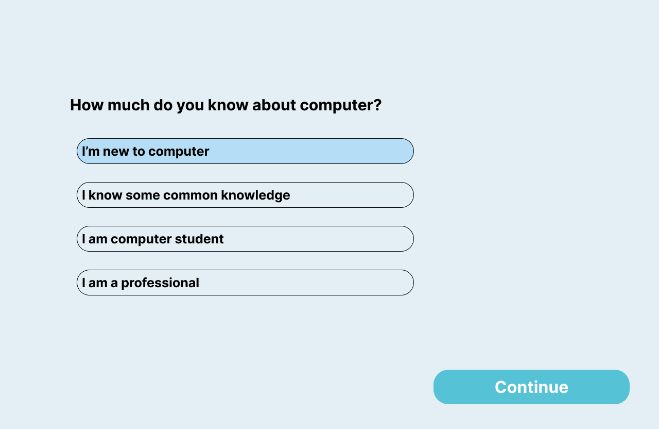


Figure 32: Knowledge Enquiry Page

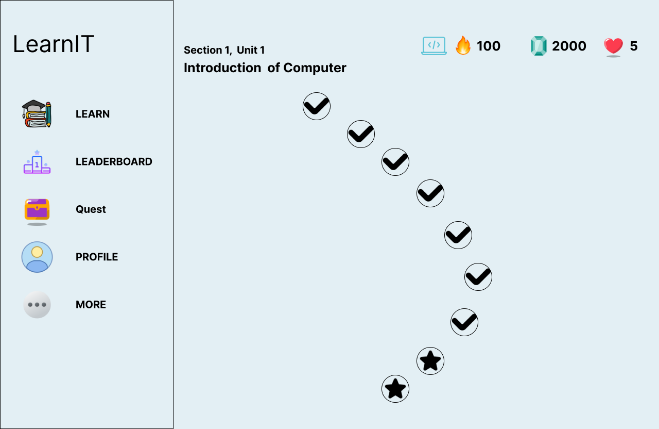


Figure 33: Home Page



Figure 34: Progress

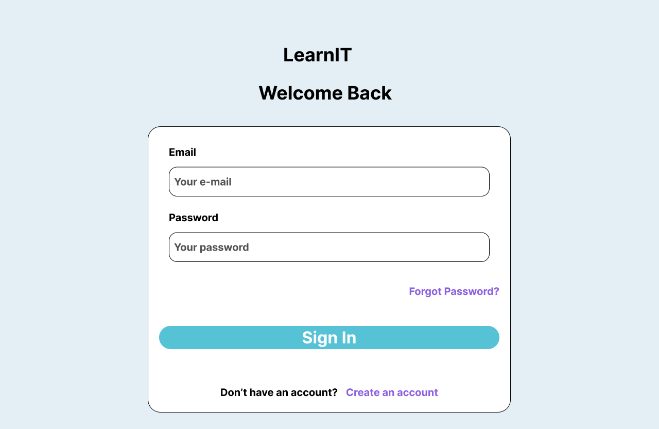


Figure 35: Login

# References

I., B., 2003. *Practical Software Testing.* s.l.:s.n.

Méndez Fernández, D. B. W. V. A. M. J. B. M. K. M. &. W. T., 2019. Artefacts in software engineering: a fundamental positioning. *Software & Systems Modeling,* 18(2019), p. 2777–2786.

Morandini, M. C. T. O. J. E. &. C. P., 2021. Considerations about the efficiency and sufficiency of the utilization of the Scrum methodology: A survey for analyzing results for development teams. Volume 39, p. 100314 .

Myers, G. J. S. C. &. B. T., 2020. *The Art of Software Testing.* s.l.:s.n.