

**MSYADD1 Final Paper for  
DOST STARBOOKS: Whiz Challenge**

Project Documentation Submitted to the Faculty of the  
School of Computing and Information Technologies  
Asia Pacific College

In Partial Fulfillment of the Requirements for  
Systems Analysis and Detailed Design  
MSYADD1

Submitted by:  
**Dumbrique, Kelly (Team Leader)**  
**Gercan, Arcielle Marie (Member)**  
**Quianzon, Shandrae Lois (Member)**  
**Salipande, Janice Maxene (Member)**

G06 NEXUS – SF 231

Submitted to:  
**Mr. Jose Eugenio L. Quesada**  
Course Instructor

October 2025  
1<sup>st</sup> Term  
A.Y. 2025 – 2026

## I. PM DOCS CHAPTER 2

### 1.1 Project Charter

#### Purpose

The purpose of the STARBOOKS Whiz Challenge project is to develop an enhanced, gamified educational system for the Department of Science and Technology – Science and Technology Information Institute (DOST-STII) that transforms the traditional STARBOOKS experience into an engaging, interactive learning platform. The system aims to promote science and mathematics literacy through offline-capable educational mini-games, such as quizzes, puzzles, memory challenges, and 1v1 battles, while incorporating features like rewards, badges, and leaderboards to motivate learners. This project supports DOST-STII's mission to make digital learning accessible, enjoyable, and effective, especially in remote areas with limited internet connectivity.

#### High-level Project Description

The STARBOOKS Whiz Challenge is an interactive, offline-capable educational system designed to enhance the existing DOST-STII STARBOOKS platform through gamified learning experiences. The project aims to make Science and Mathematics more engaging for students by integrating a variety of educational mini-games such as Whiz Challenge (quiz), Whiz Memory, Whiz Puzzle, and Whiz Battle (1v1 quiz). Each game mode offers multiple difficulty levels and categories, allowing players to learn and have fun at their own pace. Players earn stars and badges as they progress, which are displayed on a leaderboard to encourage healthy competition and continuous learning.

On the administrative side, the system includes a web-based dashboard for managing game content, player data, and performance analytics. The dashboard provides visual insights such as player trends, popular game modes, and reward statistics to help DOST-STII monitor engagement and optimize educational content.

#### High-level Milestone Schedule

##### a. Requirements Gathering and Planning (April - August 2025)

- Meet with client and stakeholders
- Define scope and core features
- Gather system requirements
- Create use case and system diagrams
- Draft project timeline
- Design initial wireframes

**b. Development Phase 1 (September - October 2025)**

- Finalize system architecture
- Set up development environment
- Build registration and login
- Develop homepage and profile
- Implement leaderboard
- Create Memory Match and Puzzle games

**c. Development Phase 2 (November 2025 - March 2026)**

- Implement player-side modules (registration, game modes, rewards, leaderboard)
- Develop admin dashboard (game management, analytics dashboard, data visualization)
- Integrate offline capability and local data storage
- Conduct periodic internal testing of core functionalities

**d. Testing, Deployment, and Finalization (April - June 2026)**

- Perform system-wide testing
- Conduct user acceptance testing
- Fix bugs and optimize performance
- Finalize UI/UX polish
- Deploy system and submit deliverables

### **Rough Cost Estimate and Budget**

This project does not require direct financial funding, as it is being developed in close collaboration with DOST-STII, who serve as technical advisers throughout the process. The partnership focuses on knowledge sharing and guidance rather than monetary support. To ensure compatibility with existing STARBOOKS kiosk hardware, the development relies entirely on free and open-source tools, libraries, and platforms.

The main goal is to maintain efficiency and cost-effectiveness while upholding high standards of quality, security, and performance. Because the system will be deployed on the current DOST kiosk infrastructure, there are no anticipated additional expenses for hardware procurement or software licensing.

### **Stakeholders**

- **Project Sponsor:** DOST-STII
- **Target Users:** Grades 1 to 12 students, exhibit visitors

## **Development Team**

- **Project Manager:** Kelly Dumbrique
- **Documentation Lead:** Janice Maxene Salipande
- **UI/UX Designers:** Shandrae Lois Quianzon, Arcielle Marie Gercan, Janice Maxene Salipande
- **Frontend Developers:** Arcielle Marie Gercan, Janice Maxene Salipande
- **Backend Developers:** Shandrae Lois Quianzon, Kelly Dumbrique

## **Testing Team**

- **QA Lead:** Janice Maxene Salipande
- **Testers:** Arcielle Marie Gercan, Shandrae Lois Quianzon, Kelly Dumbrique

## **Project Manager**

Kelly Dumbrique – Project Leader, NEXUS Team

### **Project Manager's Responsibilities**

Kelly is in charge of managing the whole project, from planning all the way to deployment. As the project manager, she keeps the team organized, sets deadlines, tracks progress, and makes sure everyone stays aligned with the project goals. She also acts as the main communicator between our team and DOST-STII, making sure everything we develop meets their expectations and stays on schedule.

Work the Project Manager Will Perform on the Project:

- Lead and coordinate team meetings
- Communicate with DOST-STII consultants for technical feedback and approvals
- Track development milestones and deliverables
- Support planning, documentation, and quality assurance
- Help resolve issues or blockers faced by the team

### **Project Manager's Authority**

Kelly takes the lead when it comes to making key decisions for the team, such as assigning tasks, setting the project direction, and deciding which features to focus on first. As the project manager, she also handles the scheduling, coordinates with stakeholders, and ensures the team stays on track. If any issues or conflicts come up, she steps in to help resolve them and ensures everything gets done on time.

## **Formal Declaration of Sponsor's Support**

**Sponsor Name:** Ms. Marievic V. Narquita

**Position:** Information Services Section (ISS) Head, Department of Science and Technology – Science and Technology Information Institute (DOST-STII)

**Authority:** As the Section Head of ISS, where STARBOOKS is managed, Ms. Narquita has the authority to provide strategic guidance, ensure alignment with STARBOOKS goals, and serve as the project's main sponsor and liaison during development.

### **Statement of Support:**

As the Section Head and official sponsor of the STARBOOKS Whiz Challenge project, I express full support for this collaborative initiative between DOST-STII and the student development team from Asia Pacific College. This project embodies our shared vision of making science and technology education more engaging, interactive, and accessible-especially for students and learners in areas with limited internet connectivity.

The proposed Whiz Challenge platform will enhance the STARBOOKS experience through gamified learning, performance tracking, and offline-first technology. We believe this innovation will significantly contribute to the educational impact of our existing kiosks and support our ongoing efforts to reach a broader audience.

As sponsor, I am committed to providing the necessary guidance, feedback, and support to ensure the project's success from development to deployment. We are confident that this partnership will result in a valuable educational tool aligned with our mission.

Signed,



MARIEVIC V. NARQUITA

Supervising Science Research Specialist

Section Head, Information Services Section

Information Resources and Analysis Division, DOST-STII

11 August 2025

## 1.2 Objectives

The primary objective is to increase student engagement and replayability by introducing multiple game modes. This will reduce repetition and encourage students to spend more time on the app during events.

Another goal is to support different learning approaches. By offering solo and multiplayer challenges, as well as visual-based games, the system can cater to diverse learners and make STEM content more accessible.

The project also aims to enable performance tracking and player insights through analytics and leaderboards, helping DOST-STII measure impact and refine outreach strategies.

To address deployment challenges, the system will adopt an offline-first but sync-capable design. This ensures accessibility in low-connectivity areas while still allowing admins to update content and gather results across kiosks.

Finally, the team plans to enhance visual design and user experience. A modern UI with engaging layouts and animations will attract younger learners and keep STARBOOKS competitive with other educational apps.

By meeting these objectives, STARBOOKS will become a more engaging, scalable, and data-driven platform for promoting STEM learning nationwide.

### Categorized Objectives

Objective	Category	Measure
Increase student engagement and replayability	Engagement	Add at least 3 new game modes; increase average play sessions per student by 50% during events
Support different learning styles	Learning Experience	Add at least 3 new game modes; increase average play sessions per student by 50% during events
Enable performance tracking and player insights	Analytics and Impact	Track player demographics, badges earned, rewards distribution, and most played game mode; generate overall game reports
Improve system scalability and manageability	Deployment and Operations	Implement offline-first design with sync capability;

		allow admins to update content across kiosks within 1 hour
Enhance visual appeal and user experience	Design and Usability	Redesign interface with modern UI and animations; achieve at least 80% positive feedback in post event surveys

### 1.3 Scope

#### Within Scope:

- User registration and login system for tracking individual player progress
- Four game modes available:
  - Whiz Challenge
  - Whiz Memory Game
  - Whiz Puzzle
  - Whiz Battle (1v1 Quiz Showdown)
- Science and Math subject categories with three difficulty levels (Easy, Average, Difficult)
- Badge and reward system to recognize player achievements and boost motivation
- Leaderboard for Whiz Challenge and Whiz Battle
- Admin panel with functionalities to:
  - Manage player and admin lists
  - Manage quiz questions
  - Set number of questions, stars, and timer for each difficulty level
  - Manage badges and reward claiming
  - View and export leaderboard rankings
  - View and export game analytics
    - Total Registered Players
    - Most Played Game Mode
    - Average Session Duration
    - Male vs Female Registered Players
    - Age Distribution
    - Players by Region
    - Male vs Female Players per Game Mode
    - Rewards Distribution by Gender and Level
    - Most Played Game Mode by Age

## **Out of Scope:**

- No integration with the main STARBOOKS digital library system
- No mobile or tablet version; limited to kiosk deployment only
- No topic-specific categories for Whiz Memory Match; only default STARBOOKS-related icons are used
- Only jigsaw-type puzzles are supported in Whiz Puzzle; other puzzle formats (e.g., word games) are not included
- No image customization for memory and puzzle games; game assets and settings are fixed and cannot be modified by admins
- Whiz Battle is limited to 1v1 public matchmaking via game code; private rooms and other multiplayer formats are not yet supported
- No player-specific analytics; the admin dashboard displays only overall trends and summary statistics across all users

## **Project Deliverables**

- Fully functional STARBOOKS Whiz Challenge system with four educational game modes (Quiz, Memory, Puzzle, and 1v1 Battle)
- Offline-capable player-side application for use in STARBOOKS kiosks
- Admin dashboard for managing users, game content, analytics, and leaderboard
- Data visualization dashboard showing player trends, popular game modes, and rewards distribution
- User manuals and technical documentation (installation guide, admin guide, system design document)
- Test reports, UAT results, and final project documentation
- Deployment package for integration into existing STARBOOKS kiosks

## **Success Criteria**

- The system operates smoothly on existing STARBOOKS kiosk hardware without major performance issues
- All core game modes function correctly and provide engaging, educational gameplay
- Admin dashboard successfully records and visualizes analytics data
- System performs reliably in offline mode and syncs data properly
- Positive feedback from test users and DOST-STII representatives during UAT
- Deployment completed on schedule and ready for integration by June 2026

## Project Assumptions

- DOST-STII will provide access to relevant content, references, and technical guidance
- Existing STARBOOKS kiosk hardware meets the system's minimum requirements
- Development team will have consistent collaboration and communication with DOST-STII
- Internet connectivity will not be required for normal operation of the player module
- Open-source tools and libraries used will remain stable and compatible throughout development

## Project Constraints

- Limited development timeframe leading up to the June 2026 deployment target
- Dependence on DOST-STII's schedule for content provision and system testing
- Restricted hardware specifications of existing STARBOOKS kiosks
- Limited manpower and resources within the student development team
- Must function entirely offline, except for optional admin updates or analytics export

## 1.4 Stakeholder Analysis

Name	Position	Objectives, Requirements, Interests	Influence	Project Contribution	Resistance
Ms. Marievic V. Narquita	STARBOOKS Unit Head / Main Project Sponsor	Ensure the system aligns with DOST-STII's STARBOOKS goals, supports educational outreach, and remains compatible with existing kiosks.	High	Provides project direction, approves requirements, validates deliverables.	Possible resistance if the system deviates from STARBOOKS standards or exceeds hardware limitations.
Mr. McGyver C. Basaya	STARBOOKS Programmer / Technical Adviser	Ensure technical feasibility, provide guidance on system integration and compatibility with STARBOOKS infrastructure.	Medium	Offers technical advice, reviews code implementation, ensures system compatibility.	Minor resistance if technical recommendations are not followed or system causes performance issues.

Kelly Dumbrique	Project Manager / Product Owner / Backend Developer / Tester	Deliver a high-quality, functional, and secure system on time; ensure coordination across development and testing tasks.	High	Oversees project progress, manages backend logic, coordinates team efforts.	Possible resistance to major scope changes or unrealistic deadlines.
Arcielle Marie Gercan	UX/UI Designer / Frontend Developer / Tester	Create intuitive and engaging user interfaces for both player and admin sides.	High	Designs front-end visuals, ensures user-friendly UI/UX, assists in testing.	Possible resistance if design constraints limit creativity or usability.
Shandrae Lois Quianzon	UX/UI Designer / Backend Developer / Tester	Develop stable backend systems and support UI design consistency.	High	Builds backend features, contributes to design, performs functionality testing.	Minor resistance if workload or role overlap causes inefficiency.
Janice Maxene Salipande	UX/UI Designer / Frontend Developer / QA Lead / Documentation Lead	Ensure system quality, accurate documentation, and consistent UI/UX implementation.	High	Leads QA testing, prepares documentation, handles frontend development.	Possible resistance if documentation or QA requirements are deprioritized.
Mr, Jose Eugenio L. Quesada	Project Adviser	Ensure the project aligns with academic and SDLC standards; guide students through proper methodologies.	Medium	Provides academic supervision, reviews milestones, ensures project quality.	Minor resistance if academic requirements are overlooked or deliverables are delayed.
Grades 1–12 Students, Exhibit Visitors	End Users	Access engaging, educational Science and Math games in STARBOOKS kiosks; prefer intuitive, fun, and offline-capable gameplay.	Medium	Provide user feedback through testing and evaluation; help validate usability and engagement.	Possible resistance if system is too complex, slow, or unappealing to users.

## II. OPENPROJECT ACTIVITIES

### 2.1 Budgets

#### Initial Phase

##### 1. Initiation Phase Budget

Added by Shandrea Lois Quianzon 2 months ago. Updated about 2 months ago.

Cost type	Budget		
Fixed date	08/29/2025		
Spent (ratio)	<div style="width: 10px; height: 10px; background-color: #ccc;"></div> 7% Total progress		
Description	The Initiation Phase budget covers all essential costs required to formally start the STARBOOKS Whiz Challenge project for DOST-STII. This includes expenses for setting up the initial work environment, conducting the kick-off meeting, and preparing the necessary tools, materials, and resources to define the project scope and objectives.		
<b>UNITS</b>			
<b>Planned unit costs</b>			
UNITS	COST TYPE	COMMENT	BUDGET
1.00	Cloud Services Monthly Subscription	One Drive 100GB storage	PHP 100.00
1.00	Office Supplies Php100	Bond Papers for Printing of Forms	PHP 100.00
8.00	Transportation expenses Php 100	Travel to DOST-STII	PHP 800.00
			PHP 1,000.00
<b>Actual unit costs</b>			
WORK PACKAGE	UNITS	COST TYPE	COSTS
Task #365: 1.1 Design Thinking Stage 1: Empathize	2.00	Contingency Reserve	PHP 200.00
Task #370: 1.0.1 Prospective Client Discovery	2.00	Transportation expenses Php 100	PHP 200.00
Task #374: 1.0.2 Finalize choice of IP/PBL Client	2.00	Transportation expenses Php 100	PHP 200.00
			PHP 600.00
<b>LABOR</b>			
<b>Planned labor costs</b>			
HOURS	USER	COMMENT	BUDGET
100.00 hours	Kelly Dumbrigue	Contribution to Transportation Costs for DOST-STII Visits	PHP 10,000.00
100.00 hours	Arcielle Gercan	Contribution to Transportation Costs for DOST-STII Visits	PHP 10,000.00
100.00 hours	Shandrea Lois Quianzon	Contribution to Transportation Costs for DOST-STII Visits	PHP 10,000.00
100.00 hours	Janice Maxene Salipande	Contribution to Transportation Costs for DOST-STII Visits	PHP 10,000.00
			PHP 40,000.00
<b>Actual labor costs</b>			
WORK PACKAGE	HOURS	USER	COSTS
Task #365: 1.1 Design Thinking Stage 1: Empathize	20.00 hours	Janice Maxene Salipande	PHP 2,000.00
Task #376: 1.2.4 Create Stakeholder Analysis	2.00 hours	Arcielle Gercan	PHP 0.00
Task #395: 1.2.3 Create Scope	2.00 hours	Janice Maxene Salipande	PHP 0.00
Task #379: 1.0.1 Prospective Client Discovery	10.00 hours	Kelly Dumbrigue	PHP 0.00
Task #374: 1.0.2 Finalize choice of IP/PBL Client	5.00 hours	Kelly Dumbrigue	PHP 0.00
Task #373: 1.2.1 Create Charter	2.00 hours	Kelly Dumbrigue	PHP 200.00
Task #374: 1.2.2 Create Objectives	2.00 hours	Shandrea Lois Quianzon	PHP 0.00
Summary-Task #366: 1.3 Define Budgets	2.00 hours	Janice Maxene Salipande	PHP 200.00
			PHP 2,400.00

#### Planning Phase

##### 2. Planning Budget Phase

Added by Arcielle Gercan 2 months ago. Updated 2 months ago.

Cost type	Budget		
Fixed date	08/13/2025		
Spent (ratio)	<div style="width: 10px; height: 10px; background-color: #ccc;"></div> 10% Total progress		
Description	The Planning Phase budget provides the resources needed to design, organize, and finalize the development plan for the STARBOOKS Whiz Challenge project.		
<b>UNITS</b>			
<b>Planned unit costs</b>			
UNITS	COST TYPE	COMMENT	BUDGET
1.00	Cloud Services Monthly Subscription	One Drive 100GB storage	PHP 100.00
5.00	Food Allowance	Snacks & drinks for meetings	PHP 500.00
8.00	Transportation expenses Php 100	Travel to DOST-STII	PHP 800.00
4.00	Miscellaneous Php100	Mobile data load for devices	PHP 400.00
			PHP 1,800.00
<b>Actual unit costs</b>			
WORK PACKAGE	UNITS	COST TYPE	COSTS
Summary-Task #346: 2.1 Design Thinking Stage 2: Define	5.00 hours	Arcielle Gercan	PHP 500.00
Summary-Task #366: 2.2 Design Thinking Stage 3: Ideate	5.00 hours	Shandrea Lois Quianzon	PHP 500.00
Task #364: 2.3.1.2 Dataflow Diagram Level 1	3.00 hours	Janice Maxene Salipande	PHP 300.00
Task #363: 2.3.1.1 Context Diagram	3.00 hours	Shandrea Lois Quianzon	PHP 300.00
Task #360: 2.3.1.2.2 Dataflow Diagram Level 2.2 Play Game	3.00 hours	Arcielle Gercan	PHP 300.00
Task #109: 2.3.1.2.6 Dataflow Diagram Level 2.6 Manage Player List	3.00 hours	Janice Maxene Salipande	PHP 300.00
Task #550: 2.3.1.2.7 Dataflow Diagram Level 2.7 View Analytics Data...	3.00 hours	Janice Maxene Salipande	PHP 300.00
Task #554: 2.3.1.2.8 Dataflow Diagram Level 2.8 View Leaderboard	3.00 hours	Janice Maxene Salipande	PHP 300.00
Task #367: 2.3.1.2.1 Dataflow Diagram Level 2.1 Manage Player Acco...	3.00 hours	Shandrea Lois Quianzon	PHP 300.00
<b>LABOR</b>			
<b>Planned labor costs</b>			
HOURS	USER	COMMENT	BUDGET
100.00 hours	Arcielle Gercan		PHP 10,000.00
100.00 hours	Janice Maxene Salipande		PHP 10,000.00
100.00 hours	Kelly Dumbrigue		PHP 10,000.00
100.00 hours	Shandrea Lois Quianzon		PHP 10,000.00
100.00 hours			PHP 40,000.00
<b>Actual labor costs</b>			
WORK PACKAGE	HOURS	USER	COSTS
Summary-Task #346: 2.1 Design Thinking Stage 2: Define	5.00 hours	Arcielle Gercan	PHP 500.00
Summary-Task #366: 2.2 Design Thinking Stage 3: Ideate	5.00 hours	Shandrea Lois Quianzon	PHP 500.00
Task #364: 2.3.1.2 Dataflow Diagram Level 1	3.00 hours	Janice Maxene Salipande	PHP 300.00
Task #363: 2.3.1.1 Context Diagram	3.00 hours	Shandrea Lois Quianzon	PHP 300.00
Task #360: 2.3.1.2.2 Dataflow Diagram Level 2.2 Play Game	3.00 hours	Arcielle Gercan	PHP 300.00
Task #109: 2.3.1.2.6 Dataflow Diagram Level 2.6 Manage Player List	3.00 hours	Janice Maxene Salipande	PHP 300.00
Task #550: 2.3.1.2.7 Dataflow Diagram Level 2.7 View Analytics Data...	3.00 hours	Janice Maxene Salipande	PHP 300.00
Task #554: 2.3.1.2.8 Dataflow Diagram Level 2.8 View Leaderboard	3.00 hours	Janice Maxene Salipande	PHP 300.00
Task #367: 2.3.1.2.1 Dataflow Diagram Level 2.1 Manage Player Acco...	3.00 hours	Shandrea Lois Quianzon	PHP 300.00

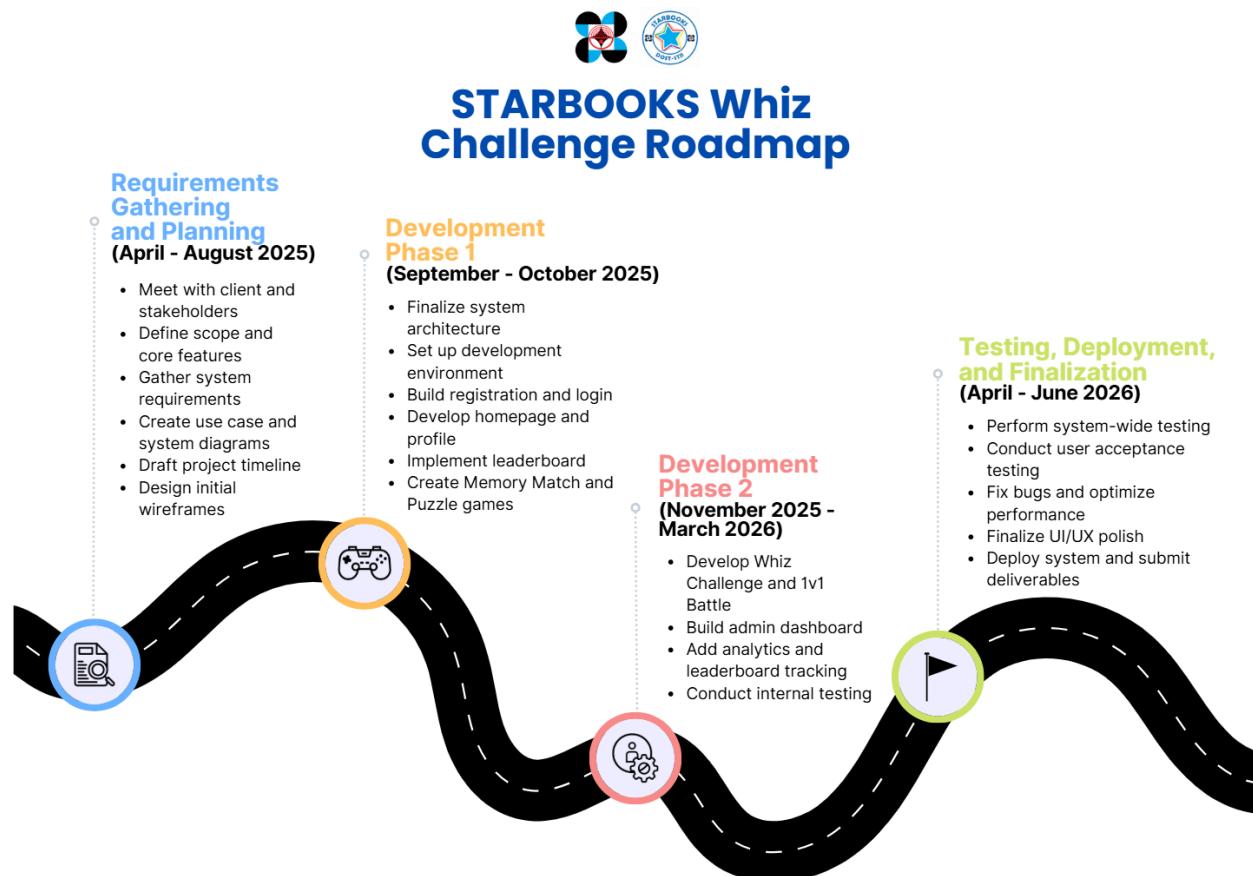
## Executing Phase

### 3. Executing Phase Budget

Added by Arcielle Gercan 2 months ago. Updated 2 months ago.

Cost type	Budget
Fixed date	08/13/2025
Spent (ratio)	<div style="width: 2%;"><div style="width: 2%;">2% Total progress</div></div>
Description	The Execution Phase budget supports the full-scale development of the STARBOOKS Whiz Challenge application, covering all resources needed to build, test, and refine the system.
UNITS	
Planned unit costs	Actual unit costs
UNITS COST TYPE COMMENT BUDGET	WORK PACKAGE UNITS COST TYPE COSTS
6.00 Cloud Services Monthly Subscription One Drive 100GB storage PHP 600.00	Task #367: 3.1 Design Thinking Stage 4: Prototype 2.00 Transportation expenses Php 100 PHP 400.00
8.00 Food Allowance Meals/snacks for dev sessions PHP 800.00	Task #368: 3.2 Design Thinking Stage 5: Test 2.00 Transportation expenses Php 100 PHP 400.00
8.00 Transportation expenses Php 100 Travel to DOST-STII for mid-development demos PHP 800.00	
4.00 Miscellaneous Php 100 Mobile data load for devices PHP 400.00	
	PHP 800.00
	PHP 2,600.00
LABOR	
Planned labor costs	Actual labor costs
HOURS USER COMMENT BUDGET	WORK PACKAGE HOURS USER COSTS
100.00 hours Arcielle Gercan PHP 10,000.00	Task #367: 3.1 Design Thinking Stage 4: Prototype 20.00 hours Shandrea Lois Quianzon PHP 0.00
100.00 hours Janice Maxene Salipande PHP 10,000.00	Task #368: 3.2 Design Thinking Stage 5: Test 3.00 hours Kelly Dumbrigue PHP 0.00
100.00 hours Kelly Dumbrigue PHP 10,000.00	
100.00 hours Shandrea Lois Quianzon PHP 10,000.00	
	PHP 0.00
	PHP 40,000.00

## 2.2 Roadmap



## 2.3 Work Packages

373	1.2.1 Create Charter	TASK	Closed	 Kelly Dumbrique	Normal		Initiating	1 - Planning and Design
374	1.2.2 Create Objectives	TASK	Closed	 Shandrae Lois Quianzon	Normal		Initiating	1 - Planning and Design
375	1.2.3 Create Scope	TASK	Closed	 Janice Maxene Salipande	Normal		Initiating	1 - Planning and Design
376	1.2.4 Create Stakeholder Analysis	TASK	Closed	 Arcielle Gercan	Normal		Initiating	1 - Planning and Design
377	1.2.5 PM Foundations Ch 2 Initiate a project Completed	MILESTONE	Closed	 Janice Maxene Salipande	Normal		Initiating	1 - Planning and Design
362	1.3 Define Budgets	SUMMARY TASK	Closed	 Arcielle Gercan	Normal		Initiating	1 - Planning and Design
363	1.4 Define Timelines	SUMMARY TASK	New	-	Normal		Initiating	1 - Planning and Design
344	▼ 2.0 Planning Phase	PHASE	In progress	 Janice Maxene Salipande	Normal		Planning	1 - Planning and Design
348	2.1 Design Thinking Stage 2: Define	SUMMARY TASK	Closed	 Arcielle Gercan	Normal		Planning	1 - Planning and Design
366	2.2 Design Thinking Stage 3: Ideate	SUMMARY TASK	Closed	 Shandrae Lois Quianzon	Normal		Planning	1 - Planning and Design
369	▼ 2.3 Requirements and Analysis Design Diagrams	SUMMARY TASK	Closed	 Shandrae Lois Quianzon	Normal		Planning	1 - Planning and Design
378	▼ 2.3.1 Dataflow Diagrams	SUMMARY TASK	Closed	 Kelly Dumbrique	Normal		Planning	1 - Planning and Design
383	2.3.1.1 Context Diagram	TASK	Closed	 Shandrae Lois Quianzon	Normal		Planning	1 - Planning and Design
384	▼ 2.3.1.2 Dataflow Diagram Level 1	TASK	Closed	 Janice Maxene Salipande	Normal		Planning	1 - Planning and Design
387	2.3.1.2.1 Dataflow Diagram Level 2.1 Manage Player Account	TASK	Closed	 Shandrae Lois Quianzon	Normal		Planning	1 - Planning and Design
388	2.3.1.2.2 Dataflow Diagram Level 2.2 Play Game	TASK	Closed	 Arcielle Gercan	Normal		Planning	1 - Planning and Design
1721	2.3.1.2.3 Dataflow Diagram Level 2.3 Reward Redemption	TASK	Closed	 Arcielle Gercan	Normal		Planning	1 - Planning and Design
1723	2.3.1.2.4 Dataflow Diagram Level 2.4 Manage Admin Account	TASK	Closed	 Kelly Dumbrique	Normal		Planning	1 - Planning and Design
344	▼ 2.0 Planning Phase	PHASE	In progress	 Janice Maxene Salipande	Normal		Planning	1 - Planning and Design
369	▼ 2.3 Requirements and Analysis Design Diagrams	SUMMARY TASK	Closed	 Shandrae Lois Quianzon	Normal		Planning	1 - Planning and Design
378	▼ 2.3.1 Dataflow Diagrams	SUMMARY TASK	Closed	 Kelly Dumbrique	Normal		Planning	1 - Planning and Design
384	▼ 2.3.1.2 Dataflow Diagram Level 1	TASK	Closed	 Janice Maxene Salipande	Normal		Planning	1 - Planning and Design
1812	2.3.1.2.5 Dataflow Diagram Level 2.5 Manage Quiz Settings	TASK	Closed	 Kelly Dumbrique	Normal		Planning	1 - Planning and Design
1813	2.3.1.2.6 Dataflow Diagram Level 2.6 Manage Player List	TASK	Closed	 Shandrae Lois Quianzon	Normal		Planning	1 - Planning and Design
2520	2.3.1.2.7 Dataflow Diagram Level 2.7 View Analytics Dashboard	TASK	Closed	 Janice Maxene Salipande	Normal		Planning	1 - Planning and Design
2521	2.3.1.2.8 Dataflow Diagram Level 2.8 View Leaderboard	TASK	Closed	 Janice Maxene Salipande	Normal		Planning	1 - Planning and Design
379	▼ 2.3.2 Use Cases	SUMMARY TASK	Closed	 Janice Maxene Salipande	Normal		Planning	1 - Planning and Design
385	▼ 2.3.2.1 Use Case Diagram	SUMMARY TASK	Closed	 Janice Maxene Salipande	Normal		Planning	1 - Planning and Design
389	▼ 2.3.2.1.1 Fully Dressed Use Cases for Manage Player Account	SUMMARY TASK	Closed	 Shandrae Lois Quianzon	Normal		Planning	1 - Planning and Design
393	2.3.2.1.1.0 Activity Diagram with Swimlanes for Manage Player Account	TASK	Closed	 Shandrae Lois Quianzon	Normal		Planning	1 - Planning and Design
391	2.3.2.1.1.1 Test Cases for Manage Player Account	TASK	Closed	 Shandrae Lois Quianzon	Normal		Planning	1 - Planning and Design
2522	▼ 2.3.2.1.2 Fully Dressed Use Cases for Play Game	SUMMARY TASK	Closed	 Arcielle Gercan	Normal		Planning	1 - Planning and Design
2876	2.3.2.1.2.0 Activity Diagram with Swimlanes for Play Game	TASK	Closed	 Arcielle Gercan	Normal		Planning	1 - Planning and Design
2906	2.3.2.1.2.1 Test Cases for Play Game	TASK	Closed	 Arcielle Gercan	Normal		Planning	1 - Planning and Design
2523	▼ 2.3.2.1.3 Fully Dressed Use Cases for Reward Redemption	SUMMARY TASK	Closed	 Arcielle Gercan	Normal		Planning	1 - Planning and Design
2877	2.3.2.1.3.0 Activity Diagram with Swimlanes for Reward Redemption	TASK	Closed	 Arcielle Gercan	Normal		Planning	1 - Planning and Design
2907	2.3.2.1.3.1 Test Cases for Reward Redemption	TASK	Closed	 Arcielle Gercan	Normal		Planning	1 - Planning and Design
2524	▼ 2.3.2.1.4 Fully Dressed Use Cases for Manage Admin Account	SUMMARY TASK	Closed	 Kelly Dumbrique	Normal		Planning	1 - Planning and Design
2878	2.3.2.1.4.0 Activity Diagram with Swimlanes for Manage Admin Account	TASK	Closed	 Kelly Dumbrique	Normal		Planning	1 - Planning and Design
2908	2.3.2.1.4.1 Test Cases for Manage Admin Account	TASK	Closed	 Kelly Dumbrique	Normal		Planning	1 - Planning and Design
2525	▼ 2.3.2.1.5 Fully Dressed Use Cases for Manage Quiz Settings	SUMMARY TASK	Closed	 Kelly Dumbrique	Normal		Planning	1 - Planning and Design
2879	2.3.2.1.5.0 Activity Diagram with Swimlanes for Manage Quiz Settings	TASK	Closed	 Kelly Dumbrique	Normal		Planning	1 - Planning and Design
2909	2.3.2.1.5.1 Test Cases for Manage Quiz Settings	TASK	Closed	 Kelly Dumbrique	Normal		Planning	1 - Planning and Design

344	2.0 Planning Phase	PHASE	In progress	Janice Maxene Salipande	Normal	Planning	1 - Planning and Design
369	2.3 Requirements and Analysis Design Diagrams	SUMMARY TASK	Closed	Shandrae Lois Quianzon	Normal	Planning	1 - Planning and Design
379	2.3.2 Use Cases	SUMMARY TASK	Closed	Janice Maxene Salipande	Normal	Planning	1 - Planning and Design
385	2.3.2.1 Use Case Diagram	SUMMARY TASK	Closed	Janice Maxene Salipande	Normal	Planning	1 - Planning and Design
2525	2.3.2.1.5 Fully Dressed Use Cases for Manage Quiz Settings	SUMMARY TASK	Closed	Kelly Dumbrigue	Normal	Planning	1 - Planning and Design
2526	2.3.2.1.6 Fully Dressed Use Cases for Manage Player List	SUMMARY TASK	Closed	Shandrae Lois Quianzon	Normal	Planning	1 - Planning and Design
2880	2.3.2.1.6.0 Activity Diagram with Swimlanes for Manage Player List	TASK	Closed	Shandrae Lois Quianzon	Normal	Planning	1 - Planning and Design
2910	2.3.2.1.6.1 Test Cases for Manage Player List	TASK	Closed	Shandrae Lois Quianzon	Normal	Planning	1 - Planning and Design
2527	2.3.2.1.7 Fully Dressed Use Cases for View Analytics Dashboard	SUMMARY TASK	Closed	Janice Maxene Salipande	Normal	Planning	1 - Planning and Design
2881	2.3.2.1.7.0 Activity Diagram with Swimlanes for View Analytics Dashboard	TASK	Closed	Janice Maxene Salipande	Normal	Planning	1 - Planning and Design
2911	2.3.2.1.7.1 Test Cases for View Analytics Dashboard	TASK	Closed	Janice Maxene Salipande	Normal	Planning	1 - Planning and Design
2528	2.3.2.1.8 Fully Dressed Use Cases for View Leaderboard	SUMMARY TASK	Closed	Janice Maxene Salipande	Normal	Planning	1 - Planning and Design
2882	2.3.2.1.8.0 Activity Diagram with Swimlanes for View Leaderboard	TASK	Closed	Janice Maxene Salipande	Normal	Planning	1 - Planning and Design
2912	2.3.2.1.8.1 Test Cases for View Leaderboard	TASK	Closed	Janice Maxene Salipande	Normal	Planning	1 - Planning and Design
380	2.3.3 Entity Relationship Diagrams	SUMMARY TASK	Closed	Kelly Dumbrigue	Normal	Planning	1 - Planning and Design
1666	3.3 Development	EPIC	New	-	Normal	Executing	2 - Development
1667	3.3.1 Player Account	USER STORY	New	-	Normal	Executing	2 - Development
1668	3.3.1.1 Registration Page	FEATURE	In testing	Arcielle Gercan	Normal	Executing	2 - Development
1669	3.3.1.1.1 Create registration page layout and input fields	TASK	In progress	-	Normal	Executing	2 - Development
2927	3.3.1.1.2 Build form with validation (required fields, email format, password rules)	TASK	In progress	-	Normal	Executing	2 - Development
2928	3.3.1.1.3 Connect to API, validate inputs, hash password, save to database	TASK	In progress	-	Normal	Executing	2 - Development
2929	3.3.1.1.4 Display messages for success, invalid input, or duplicates	TASK	In progress	-	Normal	Executing	2 - Development
2930	3.3.1.1.5 Verify registration works (valid/invalid cases) and check responsiveness	TASK	In progress	-	Normal	Executing	2 - Development
2913	3.3.1.2 Login Page	FEATURE	In testing	Arcielle Gercan	Normal	Executing	2 - Development
2931	3.3.1.2.1 Create login page with username and password fields	TASK	In progress	-	Normal	Executing	2 - Development
2932	3.3.1.2.2 Build form with basic validation (required fields, invalid input handling)	TASK	In progress	-	Normal	Executing	2 - Development
2933	3.3.1.2.3 Connect to authentication API, verify credentials, manage sessions/tokens	TASK	In progress	-	Normal	Executing	2 - Development
2934	3.3.1.2.4 Show messages for wrong credentials, account not found, or successful login	TASK	In progress	-	Normal	Executing	2 - Development
2935	3.3.1.2.5 Verify login works (valid/invalid cases, empty fields, incorrect password)	TASK	In progress	-	Normal	Executing	2 - Development
1673	3.3.1.3 Homepage with Player Profile Section	FEATURE	In testing	Shandrae Lois Quianzon	Normal	Executing	2 - Development
2936	3.3.1.3.1 Create homepage layout (profile section, badges display, edit profile button, 4 game mode cards)	TASK	In progress	-	Normal	Executing	2 - Development
2937	3.3.1.3.2 Fetch and display player data (username, earned badges) from database	TASK	In progress	-	Normal	Executing	2 - Development
2938	3.3.1.3.3 Link each game mode button to its respective game screen	TASK	In progress	-	Normal	Executing	2 - Development
2939	3.3.1.3.4 Verify profile info loads correctly, badges display properly, and navigation to game modes works	TASK	In progress	-	Normal	Executing	2 - Development

## 2.4 Backlogs

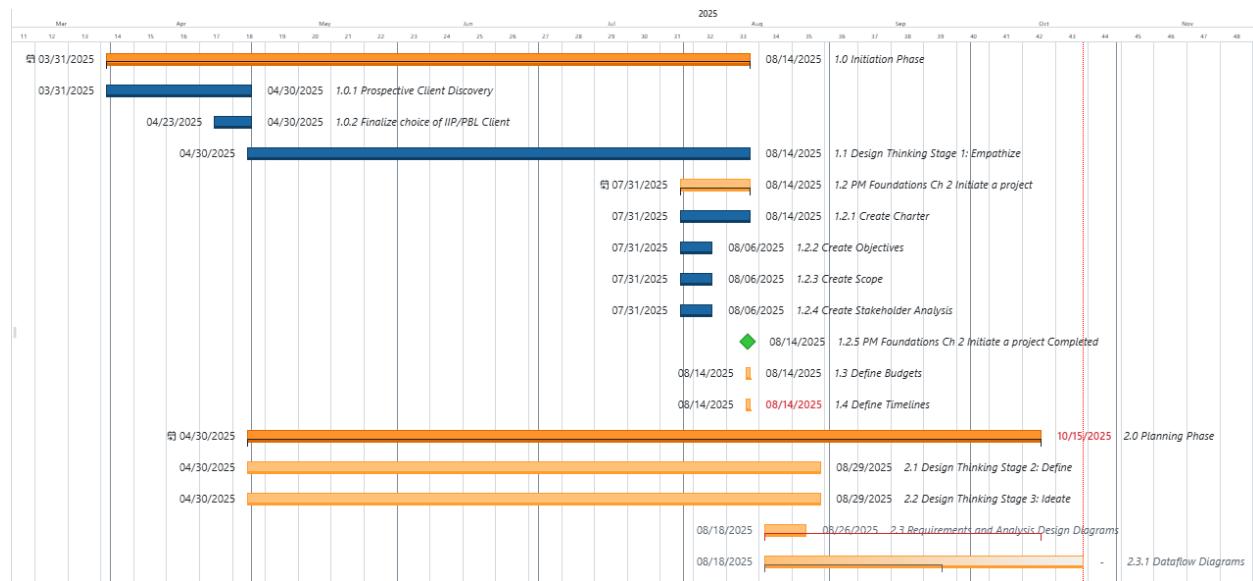
2 - Development		0
1666	Epic: 3.3 Development	New
1667	User story: 3.3.1 Player Account	New
1668	Feature: 3.3.1.1 Registration Page	In testing
2999	Feature: 3.3.1.2 Login Page	In progr...
2913	Feature: 3.3.1.2 Login Page	In testing
1673	Feature: 3.3.1.3 Homepage with Player Profile Section	In testing
2926	Feature: 3.3.1.4 Leaderboard	New
1677	User story: 3.3.2 Four Games	New
2914	Feature: 3.3.2.1 Whiz Challenge	New
2915	Feature: 3.3.2.2 Whiz Memory Match	New
2916	Feature: 3.3.2.3 Whiz Puzzle	New
2917	Feature: 3.3.2.4 Whiz Battle	New
2918	User story: 3.3.3 Admin Dashboard	New
2925	Feature: 3.3.3.1 Login Page	New
2919	Feature: 3.3.3.2 Analytics Dashboard	New
2920	Feature: 3.3.3.3 Leaderboard	New
2921	Feature: 3.3.3.4 List of Players	New
2922	Feature: 3.3.3.5 List of Admins	New
2923	Feature: 3.3.3.6 List of Questions	New
2924	Feature: 3.3.3.7 Difficulty Level Settings	New

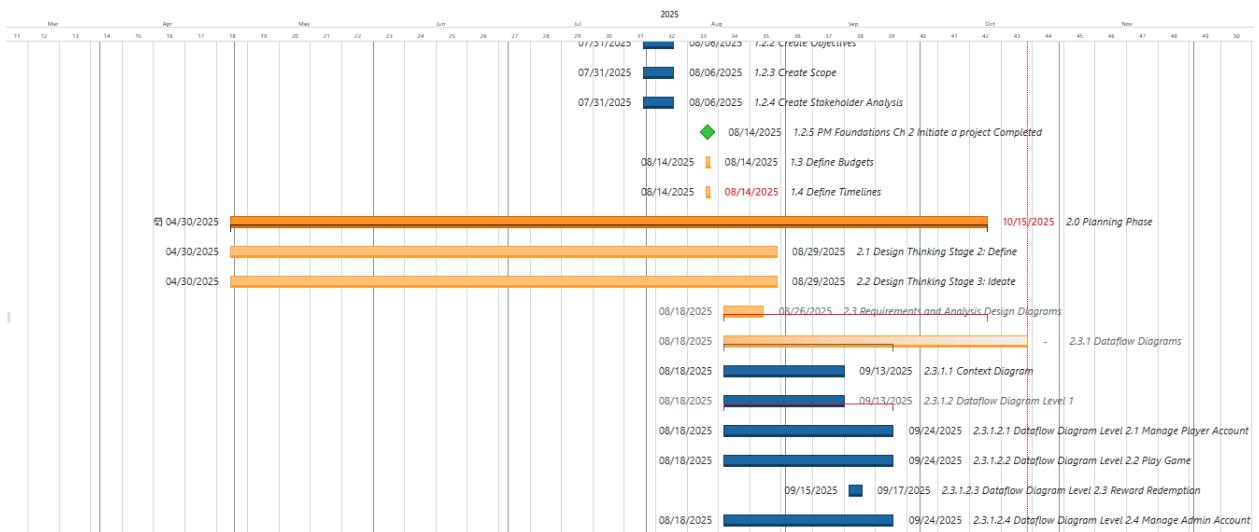
## 2.5 AY 2025-2026 T1 Sprint 01 Board

AY 2025-2026 T1 Sprint 01

Committed Backlog Items	Tasks Assigned (Not Started)	In Progress	For Review/Testing	Reviewed/Tested	Done
+ Closed #385 - APC_2025_2026_T1... <b>SUMMARY TASK</b> 2.3.2.1 Use Case Diagram Sep 11, 2025 - Sep 23, 2025	+ Closed #389 - APC_2025_2026_T1... <b>SUMMARY TASK</b> 2.3.2.1.1 Fully Dressed Use Cases for Manage Player Account Sep 11, 2025 - Oct 15, 2025	+ Closed #400 - APC_2025_2026_T1... TASK 2.3.2.1.1.0 Activity Diagram with Swimlanes for Manage Player Account Sep 15, 2025 - Oct 25, 2025	+ In Progress #1673 - APC_2025_2026... FEATURE 3.3.1.3 Homepage with Player Profile Section Sep 17, 2025	+ In Progress #1668 - APC_2025_2026... FEATURE 3.3.1.1 Registration Page Sep 17, 2025	+ Closed #400 - APC_2025_2026_T1... SUMMARY TASK 2.3.2.1 Entity Relationship Diagrams Sep 08, 2025 - Oct 15, 2025
			+ In Progress #1673 - APC_2025_2026... FEATURE 3.3.1.3 Homepage with Player Profile Section Sep 17, 2025		+ Closed #388 - APC_2025_2026_T1... TASK 2.3.2.1.8.1 Test Cases for View Leaderboard Sep 15, 2025 - Oct 15, 2025
				+ Closed #388 - APC_2025_2026_T1... TASK 2.3.2.1.8.0 Activity Diagram with Swimlanes for View Leaderboard Sep 15, 2025 - Oct 15, 2025	
					+ Closed #388 - APC_2025_2026_T1... SUMMARY TASK 2.3.2.1.8 Fully Dressed Use Cases for View Leaderboard Sep 11, 2025 - Oct 15, 2025
					+ Closed #388 - APC_2025_2026_T1... TASK 2.3.2.1.7.1 Test Cases for View Analytics Dashboard Sep 18, 2025 - Oct 15, 2025
					+ Closed #388 - APC_2025_2026_T1... TASK 2.3.2.1.7.0 Activity Diagram with Swimlanes for View Analytics Dashboard Sep 15, 2025 - Oct 15, 2025
					+ Closed #387 - APC_2025_2026_T1... <b>SUMMARY TASK</b> 2.3.2.1.7 Fully Dressed Use Cases for Play Game Sep 15, 2025 - Oct 15, 2025
					+ Closed #387 - APC_2025_2026_T1... TASK 2.3.2.1.7.0 Activity Diagram with Swimlanes for Play Game Sep 15, 2025 - Oct 15, 2025
					+ Closed #387 - APC_2025_2026_T1... <b>SUMMARY TASK</b> 2.3.2.1.7.1 Test Cases for Play Game Sep 15, 2025 - Oct 15, 2025
					+ Closed #387 - APC_2025_2026_T1... TASK 2.3.2.1.7.1.1 Test Cases for Play Game Sep 15, 2025 - Oct 15, 2025
					+ Closed #387 - APC_2025_2026_T1... <b>SUMMARY TASK</b> 2.3.2.1.7.1.2 Fully Dressed Use Cases for Play Game Sep 15, 2025 - Oct 15, 2025
					+ Closed #387 - APC_2025_2026_T1... TASK 2.3.2.1.7.1.2.0 Activity Diagram with Swimlanes for Play Game Sep 15, 2025 - Oct 15, 2025
					+ Closed #387 - APC_2025_2026_T1... <b>SUMMARY TASK</b> 2.3.2.1.7.1.2.1 Test Cases for Play Game Sep 15, 2025 - Oct 15, 2025

## 2.6 Gantt Chart





### **III. DESIGN THINKING OUTPUT**

#### **3.1 Stage 1: Empathize**

The Empathize stage focuses on understanding the needs, motivations, pain points, and experiences of the stakeholders involved with the STARBOOKS: Whiz Challenge application. The goal of this stage is to immerse the team in the users' perspectives and the context in which the product is used, allowing us to identify both explicit requirements and latent needs.

For this project, the team sought to understand the perspectives of three main stakeholder groups:

1. **The STARBOOKS project team** – responsible for the vision, development, and maintenance of the application.
2. **Potential administrators** – such as teachers or librarians from STARBOOKS-partnered schools who may manage the application or assist players.
3. **End-users (players)** – primarily students from elementary to senior high school who interact with the application during events, fairs or in partner institutions.

Due to project constraints and limited access to partner schools, direct interviews with administrators and student players were not possible during this stage. Instead, insights were gathered from:

- **Ms. Marievic V. Narquita** – STARBOOKS Unit Head
- **Mr. McGyver C. Basaya** – STARBOOKS Lead Programmer, with access to both the player and admin perspectives

These interviews provided a baseline understanding of the application's current goals, user demographics, and functional scope.

## Interview with the Clients



On April 30, 2025, the team went to the DOST–Science and Technology Information Institute (STII) building in Taguig City for our first face-to-face meeting with the STARBOOKS team. This was our chance to ask open-ended questions and really dig deeper into the “why” and “how” of the Whiz Challenge app. The team came prepared with a list of questions focused on its purpose, audience, and limitations, as well as ideas for improvements.

For this section, the table below shows the summarized questions and answers from our interview with Ms. Marievic V. Narquita, the unit head of STARBOOKS.

#	Question	Answer
1	Why did DOST-STII create the STARBOOKS: Whiz Challenge app?	The app was made mainly to entertain students and visitors during DOST events and science fairs. It's also a way to get them interested in Science and Math through a fun quiz game.
2	In what contexts or events is the app usually played?	It's mostly played during exhibits, science fairs, and DOST outreach programs. It's also available in some schools and libraries that have STARBOOKS kiosks.
3	Who are the main players of the app?	Mostly students from elementary, high school, and senior high school. They usually get to play it during events or when they visit places with a STARBOOKS kiosk.
4	How do players usually access the app?	The app is available offline in kiosks prepared by the organizers at events or installed in schools.
5	What are the app's main strengths or appealing features for players?	Students like the competitive quiz style and the scoring system. It's also easy to use, so players can start right away without needing a lot of instructions.
6	What challenges or limitations have you observed in the current app?	Right now, it only has a solo quiz mode with badges. There aren't other game modes to keep players engaged for long. It also can't track scores or player progress over time.
7	What kind of feedback have you received from students or teachers about the app?	There's no formal way to gather feedback yet, but during events, students seem to enjoy playing. Some have casually mentioned they'd like more variety in the games.
8	What improvements or new features would you like to see in future versions?	More game modes, especially player battles, a better-looking interface, a reward system that makes students want to keep playing, and features for tracking performance and scores.
9	How does the STARBOOKS team measure the success of the Whiz Challenge app?	Mostly by observing how many students play and/or claim rewards during events and how engaged they look.

10	How does the Whiz Challenge app align with STARBOOKS' mission?	STARBOOKS wants to make science and technology resources more accessible, especially in areas with fewer opportunities. The app supports this by making learning more interactive and enjoyable for students.
----	--	---

During the same meeting, the team also got the chance to talk to Mr. McGyver C. Basaya, lead programmer for the Whiz Challenge app. Since he has direct experience working on the code and has access to the admin panel, his answers gave us both a technical view and an idea of how the app is managed from an admin's perspective.

The questions for him covered the current features, technical setup, limitations, and what it's like for possible admin users such as teachers or librarians. Table below summarizes his responses.

#	Question	Answer
1	What is your role in the development of the Whiz Challenge app?	I worked on programming the game, integrating the quiz logic, and setting up the admin panel features.
2	Can you describe the main features available in the current version?	The current app has a solo quiz mode with Science and Math categories. It awards badges based on performance. The admin panel allows adding and editing quiz questions.
3	How is the app usually set up during events?	It's installed on desktop PCs and can run offline to avoid internet issues.
4	Does the app collect any player data or statistics?	Yes, the admin can see the whole list of players and their badges on each kiosk.
5	How are quiz questions managed in the current system?	They're stored in a local database that's part of the app's offline package. Admins can add or edit questions through the admin panel.
6	What technical limitations have you observed with the current app?	Since it's offline, any updates to the questions need to be done manually on each kiosk. Also, the players can only access their profiles on the same kiosks they logged into because syncing is not possible.
7	Are there any security or stability issues that need to be addressed?	Since it's a local setup, the quiz content can be edited directly if

		someone has access to the admin panel.
8	What improvements would you like to see from a development perspective?	Adding different game modes, improve interface, track performance, view analytics, and a way to sync data across different kiosks.
9	How long does it usually take to update or prepare the app for an event?	If it's just changing questions, it can be done in a few minutes. But if there are new features or bug fixes, it depends on the complexity.
10	What do you think admins like teachers or librarians would find challenging in using the current system?	Since there's no analytics or tracking, admins can't see the overall performance of users. Also, the interface is simple but not very modern, so it might not be as engaging for them to use regularly.

## Personas

Personas are fictional yet realistic representations of key users, built from real data, interviews, and observations. They help the design team visualize who they are designing for, not just in terms of demographics, but also behaviors, goals, frustrations, and motivations. By putting a “face” and “story” to the users, personas make it easier to keep their needs at the center of every design decision.

For the STARBOOKS: Whiz Challenge project, the team developed two types of personas:

- **Actual Personas** – These were based on real interviews we conducted with members of the STARBOOKS team. For this stage, we created profiles for Ms. Marievic V. Narquita, the STARBOOKS Unit Head, and Mr. McGyver C. Basaya, the Lead Programmer. Both play important roles in the app’s creation and maintenance, giving us valuable insights into the purpose of the application, its current challenges, and their vision for its improvement.
- **Theoretical Personas** – These were created to represent the types of users we couldn’t directly interview due to access limitations. These include possible admins such as teachers or librarians, and players who typically interact with the app during DOST exhibit fairs. The goal of these personas is to anticipate their needs, frustrations, and expectations based on secondhand information from our interviewees, combined with our own user research and assumptions.

By working with both actual and theoretical personas, we ensured that our design process addresses both the confirmed needs of stakeholders we spoke to and the likely

needs of those we couldn't directly reach but will ultimately use or interact with the application.

## Persona 1: Ms. Marievic V. Narquita (Actual)



### Ms. Marievic V. Narquita

#### Role:

Unit Head – STARBOOKS,  
DOST-STII

#### BACKGROUND

- oversees the STARBOOKS program nationwide
- ensures that STARBOOKS reaches communities in remote areas
- coordinates with local government units, school heads, and librarians to facilitate deployment and training

#### GOALS AND MOTIVATIONS

- Ensure that STARBOOKS remains accessible and useful, even in offline settings
- Increase engagement among younger audiences
- Maintain the quality and accuracy of STEM content

#### FRUSTRATIONS / PAIN POINTS:

- Difficulty in monitoring user engagement
- Limited ways to measure whether students are actually learning from STARBOOKS

#### TECH SKILLS & HABITS:

- Not directly involved in coding or programming
- Uses common project management tools like document, email, and messaging apps

#### STARBOOKS WHIZ CHALLENGE NEEDS:

- Additional game modes to increase user plays
- Implement player battle features for better user engagement
- A simple way to check overall game usage, to see if the app is achieving its learning goals

## Persona 2: Mr. McGyver C. Basaya (Actual)



### Mr. McGyver C. Basaya

Role:  
Lead Programmer –  
STARBOOKS, DOST-STII

#### BACKGROUND

- lead developer of the current STARBOOKS app
- ensures the system is stable, bug-free, and compatible with the hardware used in kiosks

#### GOALS AND MOTIVATIONS

- Ensure that STARBOOKS runs smoothly across all kiosks
- Incorporate new features that keep the system modern while staying resource-efficient
- Reduce downtime and improve ease of maintenance

#### FRUSTRATIONS / PAIN POINTS:

- Hardware limitations in some deployments
- Variability in technical skills among librarians and teachers
- Challenges in updating and syncing content in offline environments

#### TECH SKILLS & HABITS:

- Strong programming and troubleshooting background

#### STARBOOKS WHIZ CHALLENGE NEEDS:

- A game system that is lightweight but engaging
- Modular design so new games or quizzes can be added without heavy rework
- Data syncing across different kiosks for easy admin management

## Persona 3: Chloe Ramirez (Theoretical)



### Chloe Ramirez

Role:  
Grade 5 Student –  
First-time STARBOOKS  
Visitor at Science Fair

#### BACKGROUND

- a 10-year-old elementary student visiting the science fair with her classmates
- curious with the booths and activities
- this is her first time seeing the STARBOOKS kiosks

#### GOALS AND MOTIVATIONS

- Try fun activities she's never done before
- Win tokens or prizes
- Play something she can easily understand in a few minutes

#### FRUSTRATIONS / PAIN POINTS:

- Gets bored quickly if there's too much reading
- May lose interest if the game is too hard right away
- Easily distracted by other booths

#### TECH SKILLS & HABITS:

- Knows how to use tablets and mobile games
- Prefers games with instant rewards or effects

#### STARBOOKS WHIZ CHALLENGE NEEDS:

- Eye-catching visuals
- Short game sessions
- Fun sound effects or animations that make the experience memorable

## Persona 4: John Ray Santos (Theoretical)



**John Ray  
Santos**

**Role:**  
Grade 8 Student –  
Public High School  
(STARBOOKS User)

### BACKGROUND

- a 14-year-old student in a public high school
- his school has a STARBOOKS kiosk in the library
- enjoys interactive learning

### GOALS AND MOTIVATIONS

- Understand Science and Math topics in a way that feels fun and less stressful
- Compete with classmates and earn badges

### FRUSTRATIONS / PAIN POINTS:

- Reading long text on the kiosk feels tiring
- STARBOOKS kiosk is too "serious" and doesn't feel exciting
- Limited time in the library because of other activities and class schedules

### TECH SKILLS & HABITS:

- Very familiar with using gadgets and social media
- Can quickly learn new game mechanics if they're similar to ones he already plays

### STARBOOKS WHIZ CHALLENGE NEEDS:

- Mini-games that make Science and Math feel like a challenge, not a lecture
- A way to track progress, earn rewards, and compare scores with classmates
- Colorful, animated interface that doesn't look like a plain educational app

## Persona 5: Mr. Carlo Mendoza (Theoretical)



**Mr. Carlo  
Mendoza**

**Role:**  
Teacher at the  
Science Fair

### BACKGROUND

- accompanies a group of students from his class to the science fair
- He wants them to explore as many educational booths as possible
- He appreciates tools that can make his subject more engaging in class

### GOALS AND MOTIVATIONS

- See if STARBOOKS can be integrated into his teaching
- Find ways to keep students engaged
- Discover free or low-cost educational platforms

### FRUSTRATIONS / PAIN POINTS:

- Limited time at each booth due to tight schedules
- Students tend to skip the academic booths
- Hard to monitor all students at once during the fair

### TECH SKILLS & HABITS:

- Comfortable with computers
- Open to trying new teaching tools if they're easy to adopt
- Often looks for free digital learning platforms

### STARBOOKS WHIZ CHALLENGE NEEDS:

- Quick demo mode showcasing features without long setup
- Clear handouts or flyers explaining how schools can get STARBOOKS
- Option to show student leaderboards or instant quiz results for group engagement

## Persona 6: Ms. Liza de Vera (Theoretical)



**Ms. Liza de  
Vera**

**Role:**  
School Librarian –  
Public High School

### BACKGROUND

- manages the school library and is also in charge of the STARBOOKS kiosks
- assists students in using the app
- reports technical issues to DOST-STII

### GOALS AND MOTIVATIONS

- Increase student engagement with STARBOOKS
- Make the library not just a place for research but also for interactive learning
- Reduce the number of times she has to explain basic kiosk navigation to students

### FRUSTRATIONS / PAIN POINTS:

- Students quickly lose interest if the content looks too academic
- Some students are shy to ask for help and end up not using STARBOOKS at all
- Limited training on troubleshooting

### TECH SKILLS & HABITS:

- Comfortable using basic computer applications
- Relies on IT staff or DOST-STII when technical problems are too complex
- Uses social media to promote library activities

### STARBOOKS WHIZ CHALLENGE NEEDS:

- Simple, attractive game interface that doesn't require long explanations
- Easy system maintenance with minimal technical steps
- Features that encourage friendly competition among students to boost kiosk visits

## Empathy Maps

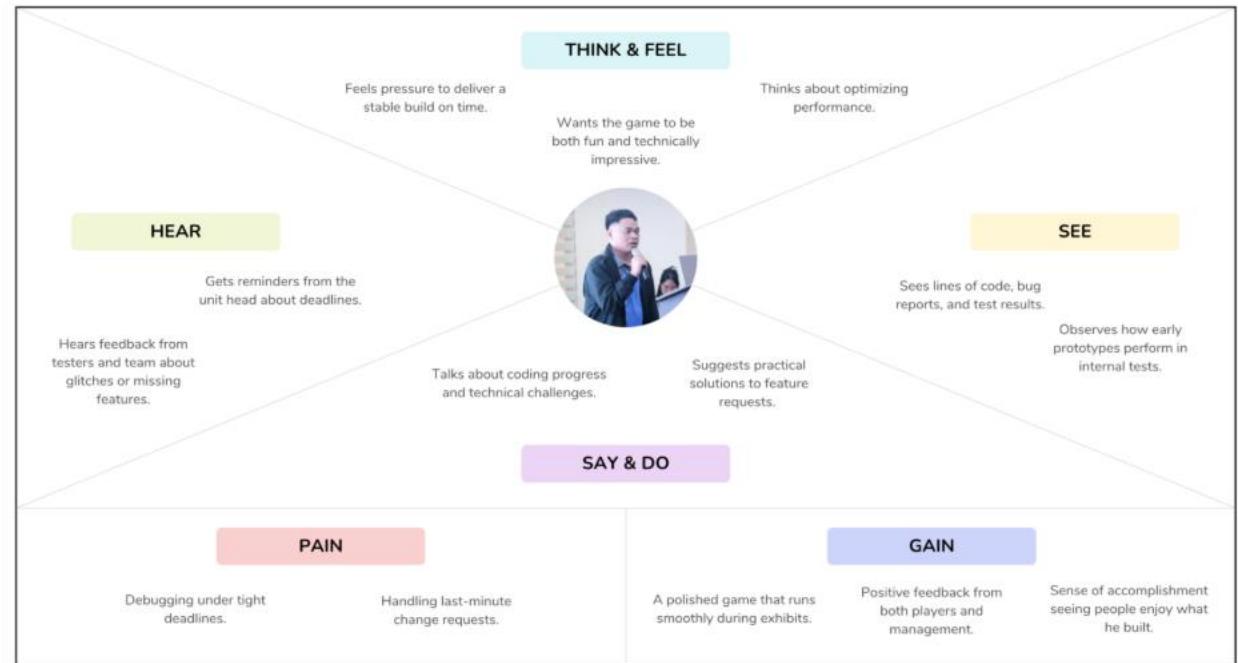
Empathy maps are visual tools that help the team step into the shoes of each persona and see the system from their perspective. An empathy map captures what a persona might say, think, do, and feel in a given context, along with their goals and pain points.

By exploring these dimensions, we can identify not only the functional needs of our users, but also their motivations, frustrations, and expectations. This helps guide design decisions that are user-centered and grounded in real scenarios.

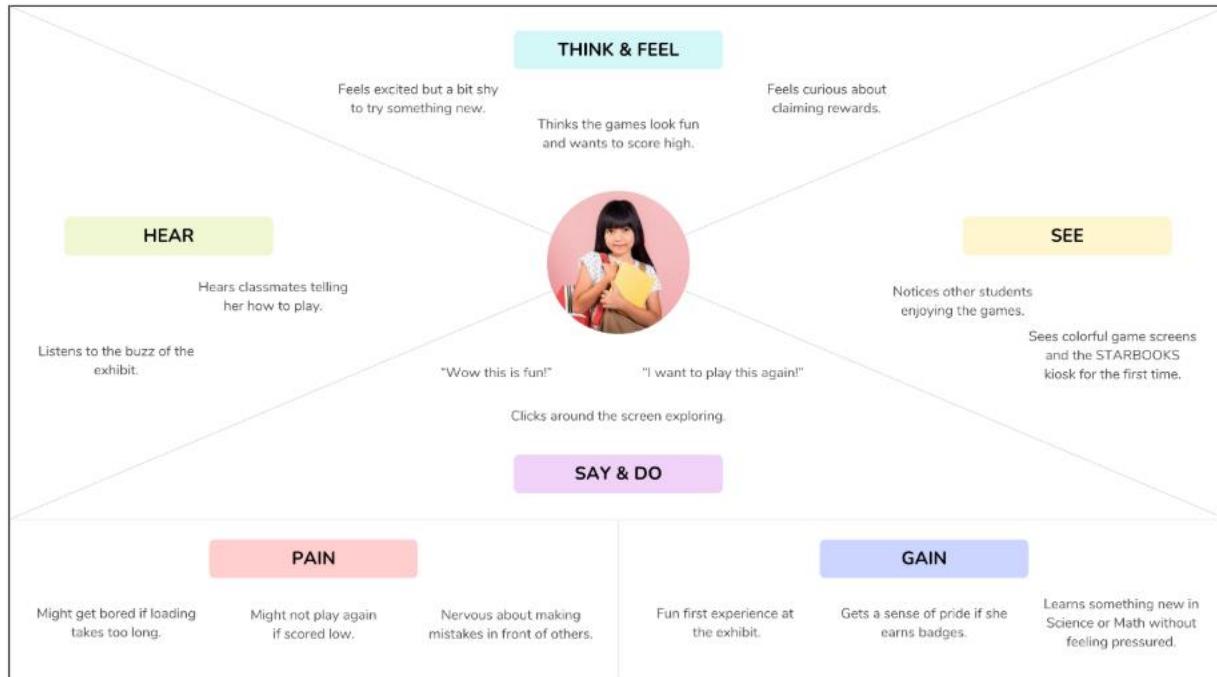
## Empathy Map 1: Ms. Marievic V. Narquita (STARBOOKS Unit Head)



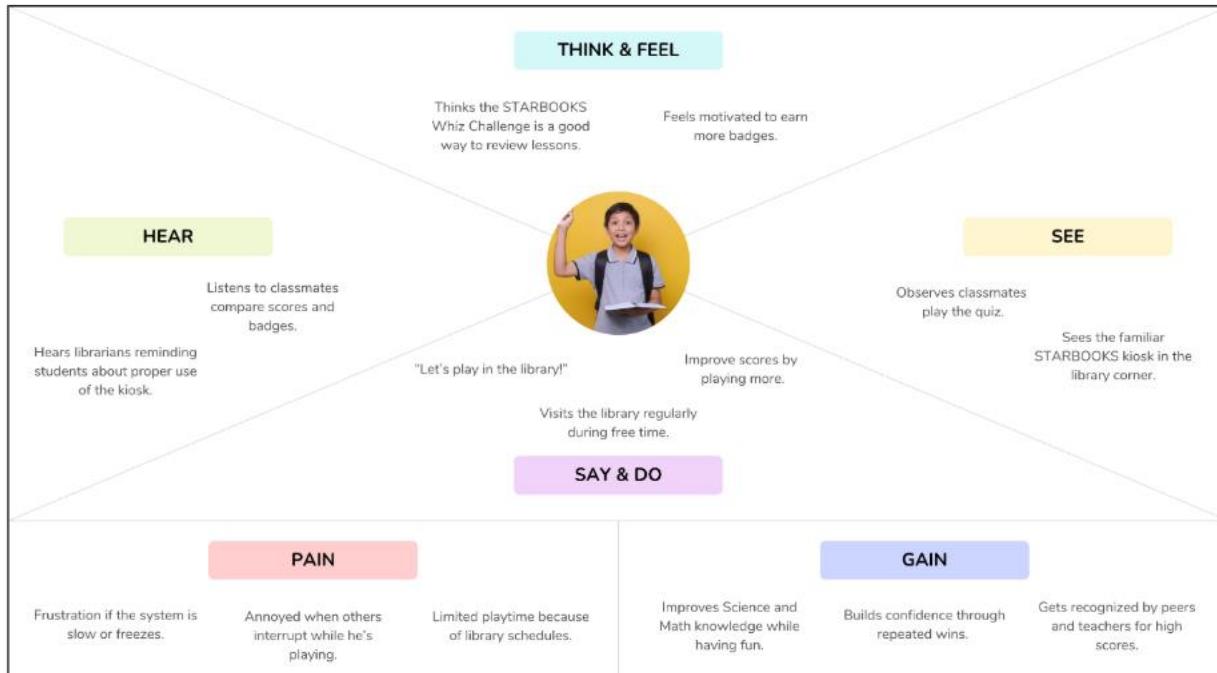
## Empathy Map 2: Mr. McGyver C. Basaya (STARBOOKS Lead Programmer)



## Empathy Map 3: Chloe Ramirez (First-time Exhibit Visitor)



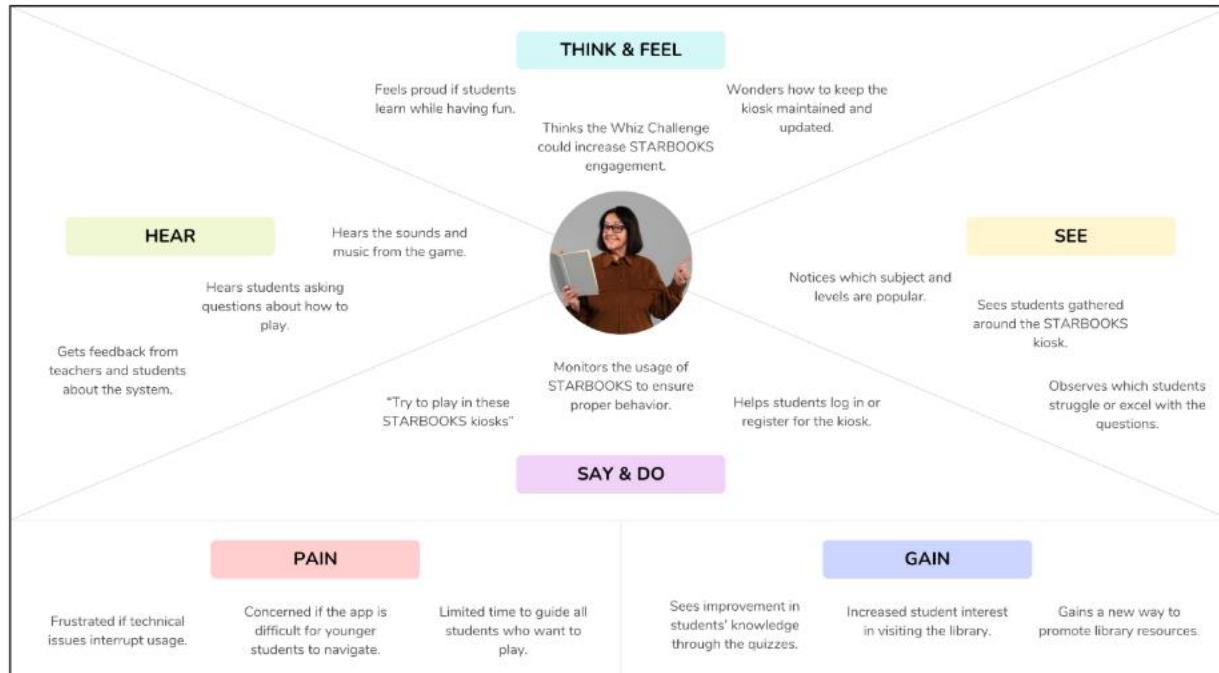
## Empathy Map 4: John Ray Santos (STARBOOKS User)



## Empathy Map 5: Mr. Carlo Mendoza (Teacher at the Science Fair)



## Empathy Map 6: Ms. Liza de Vera (School Librarian)



## Pain-Gain Analysis

Pain–Gain Analysis is a simple, yet effective tool used to identify the challenges (pains) that users face and the benefits (gains) they expect or experience when interacting with a system. By mapping out these factors for each persona, the team can better understand user needs, frustrations, and motivations. This helps in prioritizing features, improving user experience, and ensuring that the final product delivers maximum value while minimizing obstacles. Table below shows the summarized pain-gain analysis with the six personas.

Persona	Pains	Gains
Ms. Marievic V. Narquita (STARBOOKS Unit Head)	Balancing educational goals with engaging gameplay; Time constraints for managing project; Limited budget/resources	Delivering an educational tool that works offline; Showcasing innovation during exhibits; Enhancing STARBOOKS appeal
Mr. McGyver C. Basaya (Lead Programmer)	Risk of technical issues during exhibits; Time pressure for development; Ensuring smooth performance offline	Creating a polished, bug-free system; Positive feedback from players
Chloe Ramirez (First-time Exhibit Visitor)	Difficulty understanding controls at first; May feel intimidated by questions	Fun, memorable exhibit experience; Learning without realizing it; Earning badges
John Ray Santos (STARBOOKS User)	Limited time to play during breaks; Difficulty answering harder questions	Enjoyable learning during free time; Comparing scores and badges with friends; Improving Science and Math skills
Mr. Carlo Mendoza (Teacher)	Keeping students engaged in large groups; Limited time for each student to try the system	A ready-made learning tool; Students excited about Science & Math; Easy integration into class visits
Ms. Liza de Vera (School Librarian)	Technical issues disrupt usage; Younger students struggle with navigation; Limited ability to guide all players	Higher student interest in library visits; Students improve knowledge while having fun; New way to promote library resources

## **Insights:**

Looking at all the pain–gain results from the six personas, we noticed a few patterns that kept coming up:

### **1. Common Pains Across Personas**

- **Getting started and figuring things out** – First-time users often have a hard time figuring out how to play right away without someone guiding them.
- **Limited time** – Teachers and students usually only have limited time to try the games, whether it's during a class trip to the fair or quick breaks in the library.
- **Possible tech issues** – Both the project lead and lead programmer are concerned about how the system will handle heavy use during exhibit days.

### **2. Common Gains Across Personas**

- **Fun while learning** – Everyone appreciates that the games mix enjoyment with Science and Math lessons, making the subjects feel less intimidating.
- **Rewards and competition as motivation** – Badges and prizes keep students and visitors coming back for more tries.
- **Good image for schools and exhibits** – Teachers, librarians, and project leads see the system as something that can help promote their school or exhibit as modern and engaging.

### **3. Implications for System Design**

- Make the onboarding process very simple so that even first-time or younger users can try without much help.
- Design games to be short and easy to replay, so visitors can enjoy them even with limited time.
- Focus on strong offline performance and have quick ways to fix issues during busy exhibit days.

### **3.2 Stage 2: Define**

The Define stage is about making sense of all the information the team gathered during the Empathize phase. Here, we sort through the insights, group related problems together, and identify the core challenges that our STARBOOKS: Whiz Challenge application needs to address. This step helps us move from raw observations to clear, actionable problem statements.

#### **Clustering Problems**

The team scheduled an online meeting to consolidate the findings. Using a collaborative Canva whiteboard, each member listed the problems identified from the empathy maps and pain–gain analysis. The board quickly filled with sticky notes, each representing a specific user concern or frustration. Once all problems were listed, the team began grouping them into clusters based on common themes. This visual grouping shown allowed everyone to see overlapping issues across different personas and pinpoint the most critical problem areas to address moving forward.

First-time users confused about game mechanics and controls. <b>(QUIANZON)</b>	Students can't play for long because of short recess or limited exhibit time. <b>(DUMBRIQUE)</b>	Lag or slow loading during peak usage causes frustration. <b>(SALIPANDE)</b>	Some players lose interest after a few rounds because it lacks other game modes. <b>(QUIANZON)</b>	
Younger players need clearer visual cues and instructions. <b>(SALIPANDE)</b>	Games with longer question sets can't always be finished in one sitting. <b>(GERCAN)</b>	No easy way for on-site staff to reset or restart the system if a problem occurs. <b>(DUMBRIQUE)</b>	Limited variety in question formats can make quiz feel monotonous. <b>(GERCAN)</b>	
Difficulty in locating specific game levels. <b>(GERCAN)</b>	No quick-play or mini-challenge option for visitors with limited time. <b>(SALIPANDE)</b>	Badges and rewards don't always feel significant or personalized. <b>(SALIPANDE)</b>		
Events or fairs have many attractions, so the app has to capture attention fast. <b>(QUIANZON)</b>			No feature to encourage friendly team play or group competition. <b>(DUMBRIQUE)</b>	

## “How Might We” Questions

From the problem clusters, the team reframed challenges into open-ended “How Might We” questions to spark solution ideas:

1. How might we design the game interface so that even first-time or younger users can start playing without needing detailed explanations?
2. How might we simplify navigation so that key features are easy to find in just a few clicks?
3. How might we use universally familiar icons or visuals to help players understand actions without reading long text?
4. How might we structure game sessions so players can enjoy and learn from them even with only a few minutes to spare?
5. How might we let players pause and resume games without losing progress?
6. How might we ensure smooth, stable, and offline-ready gameplay during high-traffic days or in areas with limited internet access?
7. How might we make it easy for on-site staff to troubleshoot without technical training?

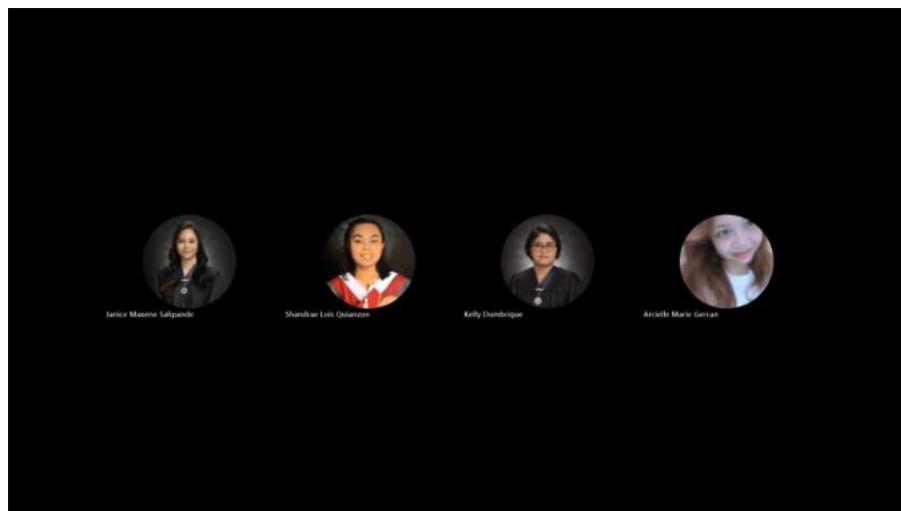
8. How might we keep players coming back by making the experience rewarding, competitive, and varied over time?
9. How might we balance educational content with fun gameplay mechanics so learning feels natural?
10. How might we make achievements and badges feel more meaningful to players?

### **3.3 Stage 3: Ideate**

The Ideate stage focuses on generating a wide range of creative solutions based on the problems and opportunities identified in the previous stages. At this point, the team shifts from problem-definition to idea-generation, aiming to explore multiple possibilities without immediately judging or narrowing them down. The goal is to encourage free thinking, build on each other's suggestions, and combine different perspectives to come up with innovative yet feasible concepts for the improved STARBOOKS: Whiz Challenge application.

#### **Brainstorming Session**

The team gathered once again in an online meeting to exchange ideas and propose creative yet practical features for the application. Using a collaborative whiteboard, everyone contributed freely without filtering or judging the ideas at this stage.



## Crazy 8's

Following the brainstorming session, the team proceeded with a Crazy 8's exercise to rapidly explore a wide range of design possibilities for the application. Each member was given eight minutes to create eight quick sketches, one per panel, capturing potential features, interface layouts, or user experience improvements.

**Crazy 8's of Kelly Dumbrique**



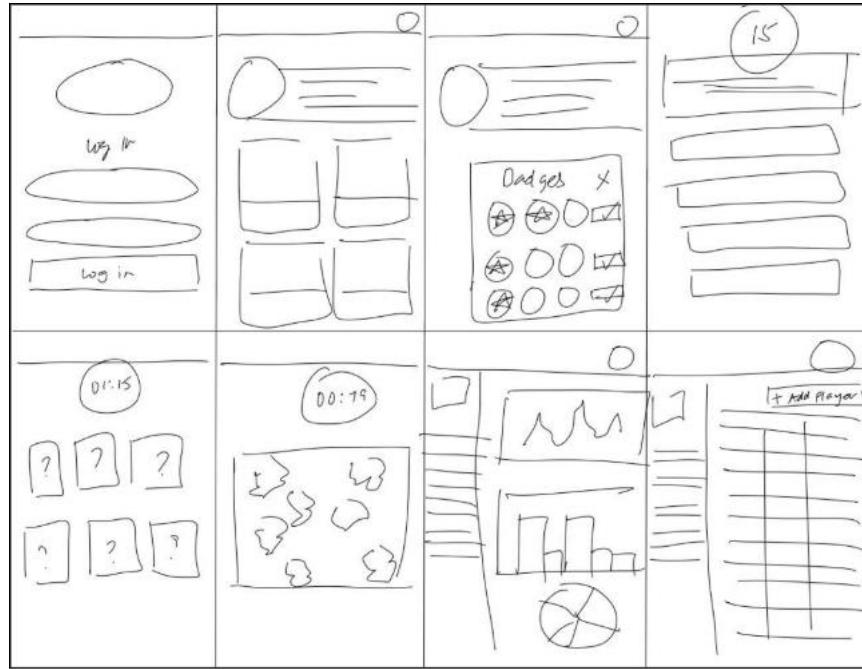
## Crazy 8's of Arcielle Marie Gercan



## Crazy 8's of Shandrea Lois Quianzon

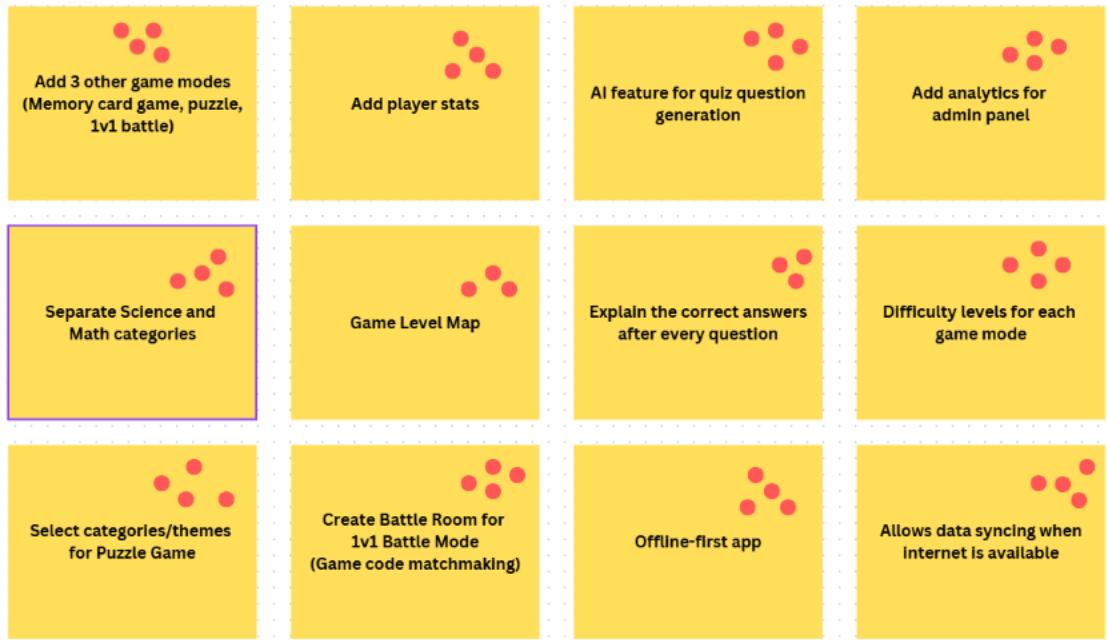
 Welcome page	Edit Name <input type="text"/> board  Short tutorials for new users b4 registration 1 full at a time	Choose Game <input type="checkbox"/> offline <input type="checkbox"/> online  	Profil V/Stat  Badges <input type="button" value="claim button"/>
Score 9/10  Time 5m 65s  Quiz ref. Summary	P1 - L  P2 - V   Whiz Battle Progress  Race Line	Timer → 0:24    Quiz → Question	Motivational msg + tips   Looking Score on After Quiz

## Crazy 8's of Janice Maxene Salipande



### Post-it Voting

After generating a wide range of ideas through the Crazy 8's exercise, the team moved on to the Post-it Voting activity to identify the most promising concepts. Using the same shared Canva whiteboard, each idea from the Crazy 8's sketches was placed into a virtual "idea gallery" where all members could review them. Every participant was given a set number of virtual sticky-dot votes to allocate to the ideas they felt were most impactful, feasible, and aligned with the goals of the project. The highest-voted ideas were then marked for further development in the next design stages.



## Desirable, Viable, Feasible

Following the Post-it Voting activity, the team evaluated the top-voted ideas against three core innovation criteria: Desirability, Viability, and Feasibility.

- **Desirable** refers to how well the idea meets the needs, wants, and preferences of the target users.
- **Viable** examines whether the idea can be sustained over time, considering factors like resources, costs, and long-term benefits.
- **Feasible** looks at the practicality of implementing the idea given the current technical skills, available tools, and timeline of the project.

This activity helped the team filter out ideas that were exciting but unrealistic and prioritize those that were both innovative and achievable within the project constraints. The resulting selection would form the foundation for the prototype development phase.

Idea	Desirable	Viable	Feasible
Add 3 other game modes (Memory card game, puzzle, 1v1 battle)	4 YES	4 YES	3 YES; 1 NO
Add player stats	4 YES	4 YES	4 YES
AI feature for quiz question generation	4 YES	4 YES	4 YES

Add analytics for admin panel	4 YES	4 YES	4 YES
Separate Science and Math categories	4 YES	4 YES	4 YES
Game Level Map	4 YES	4 YES	2 YES; 2 NO
Explain the correct answers after every question	4 YES	4 YES	4 YES
Difficulty levels for each game mode	4 YES	4 YES	4 YES
Select categories/themes for Puzzle Game	4 YES	4 YES	4 YES
Create Battle Room for 1v1 Battle Mode (Game code matchmaking)	4 YES	2 YES; 2 NO	2 YES; 2 NO
Offline-first app	4 YES	2 YES; 2 NO	2 YES; 2 NO
Allows data syncing when internet is available	4 YES	2 YES; 2 NO	2 YES; 2 NO

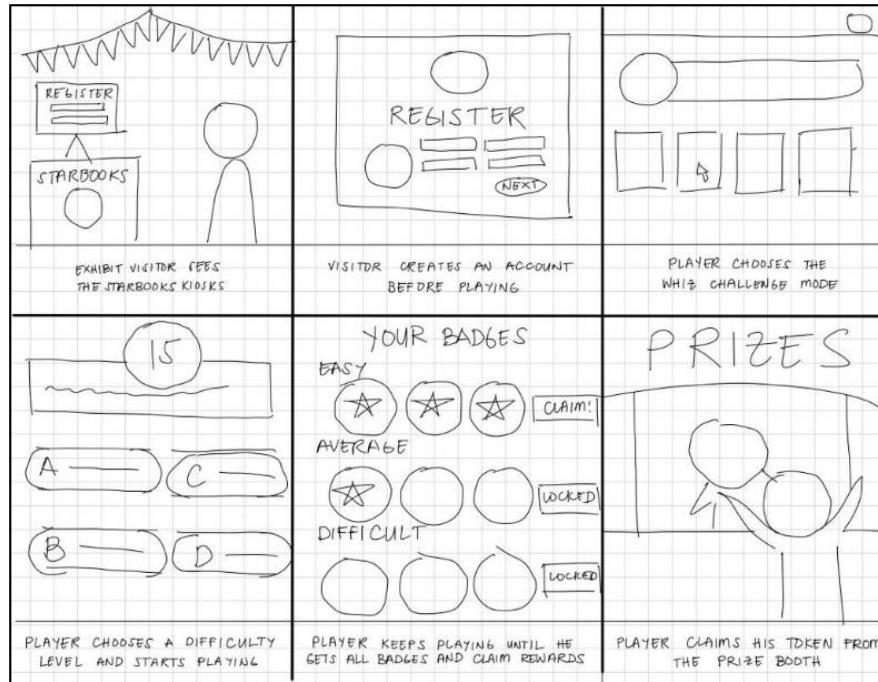
### 3.4 Stage 4: Prototype

After finalizing the most promising ideas from the ideation stage, the team moved forward to the prototyping phase. This stage focused on transforming selected concepts into tangible, testable representations of the STARBOOKS Whiz Challenge app. Using the prioritized features from the DVF analysis, the team began creating wireframes and mockups to visualize the app's interface, flow, and functionality.

#### Storyboarding

For the storyboarding phase, the team mapped out the user journey of the STARBOOKS Whiz Challenge from start to finish. Instead of jumping straight into detailed UI design, we focused on telling the story of how a typical user would interact with the application in real-life scenarios.

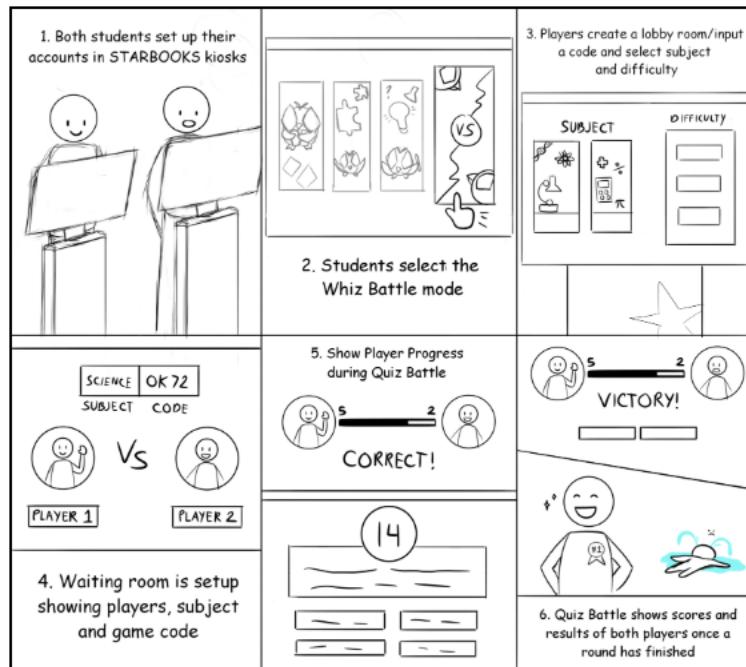
**Storyboard 1: Exhibit visitor playing the Whiz Challenge mode**  
**(made by Janice Maxene Salipande)**



**Storyboard 2: Exhibit visitor playing the Memory Match and Jigsaw Puzzle modes**  
**(made by Arcielle Marie Gercan)**



### Storyboard 3: High school students playing against each other in the 1v1 Battle mode (made by Shandrea Lois Quianzon)



### Storyboard 4: Librarian adding questions and viewing analytics in the admin panel (made by Kelly Dumbrique)



## Wireframing

With the storyboard complete, the team moved on to wireframing to start translating the user journey into a tangible app layout. Using Figma, the team created low-fidelity wireframes that mapped out screens such as the login/register pages, homepage, game mode interfaces, player stats interface, and admin dashboard. These wireframes helped clarify how users would move through the app and ensured the team was aligned on functionality before diving into detailed design.



## Mockup Design

After finalizing wireframes, the team progressed to creating high-fidelity mockups that brought the STARBOOKS Whiz Challenge interface to life with colors, typography, icons, and visual branding. The mockups were designed to be visually appealing while maintaining usability, ensuring clear readability for students and administrators alike. Through several iterations in Figma, the team refined UI elements, making sure the overall experience was engaging and easy to navigate. These mockups serve as a detailed guide for development and help stakeholders visualize the final product before coding begins.

Player Register Page > Terms and Conditions

Player Register Page > Account...

Player Register Page > Account...

Player Log in Page

Homepage

Whiz Challenge > Select Category

Whiz Challenge > Science > Easy

Whiz Challenge > Science > Animals

Whiz Challenge > Select Category

Whiz Challenge > Science > Easy > Question 3

Whiz Challenge > Science > Easy > Quiz Result > Congratulations

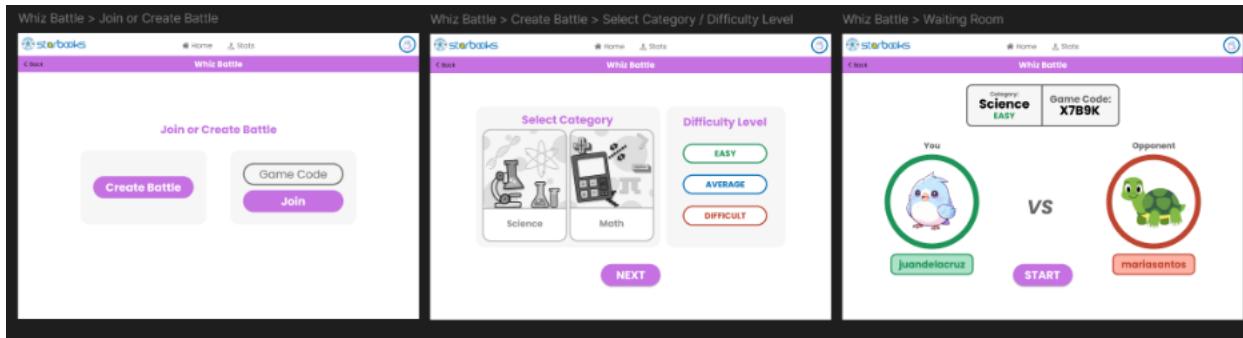
Whiz Memory Match > Difficulty Level

Whiz Memory Match > Difficulty Level > Easy

Whiz Puzzle > Difficulty Level / Select Category

Whiz Puzzle > Easy > Animals > Start

Whiz Puzzle > Easy > Animals > Finish



Battle Type	Opponent	Category	Difficulty	Total Time	Result
Whiz Battle	marioentes	Science	Easy	09/20/2020 10:45	W
Whiz Battle	marioentes	Math	Easy	09/20/2020 10:45	W
Whiz Battle	josephdelacruz	Science	Average	04/26/2020 10:27	W

Category	Name	Address	Action	Image
Student	ronald	HCK Metro Manila Caloocan City		
Student	carla	HCK Metro Manila Makati City		
Student	valerie	HCK Metro Manila Taguig City		
Employee	robert	HCK Metro Manila Paranaque City		
Student	anneke	HCK Metro Manila Pasay City		
Student	use	HCK Metro Manila Pasay City		
Student	hannah	HCK Metro Manila Las Pinas City		
Student	erick	HCK Metro Manila San Juan City		
Employee	mario	HCK Metro Manila Quezon City		

### 3.5 Stage 5: Test

The final stage of the design sprint focused on validating the STARBOOKS Whiz Challenge prototype with real users to gather feedback before moving into full development. The goal was to identify any usability issues, unclear features, or improvement opportunities early on.

For this stage, the team held another onsite meeting on May 20, 2025 with the STARBOOKS team to present and review the completed mockup designs. This session served as an internal validation step to ensure that the planned features, interface layouts, and overall user experience aligned with the STARBOOKS project's objectives. Since the focus was on securing approval and feedback from the project stakeholders, the mockups were shown only to the STARBOOKS team and not yet to the target end-users (players and admins). Their insights and suggestions were documented, with the aim of making refinements ahead of the June 9, 2025 final defense, ensuring that the team would be fully prepared for the presentation.



### Feedback

During the meeting, the STARBOOKS team provided positive feedback as well as constructive suggestions for improvement. They appreciated the color scheme, assets, and visuals, noting that the design truly looked like a game intended for children. They also liked the game level map concept, which they felt would make progression more engaging for players.

For **enhancements**, they suggested the following:

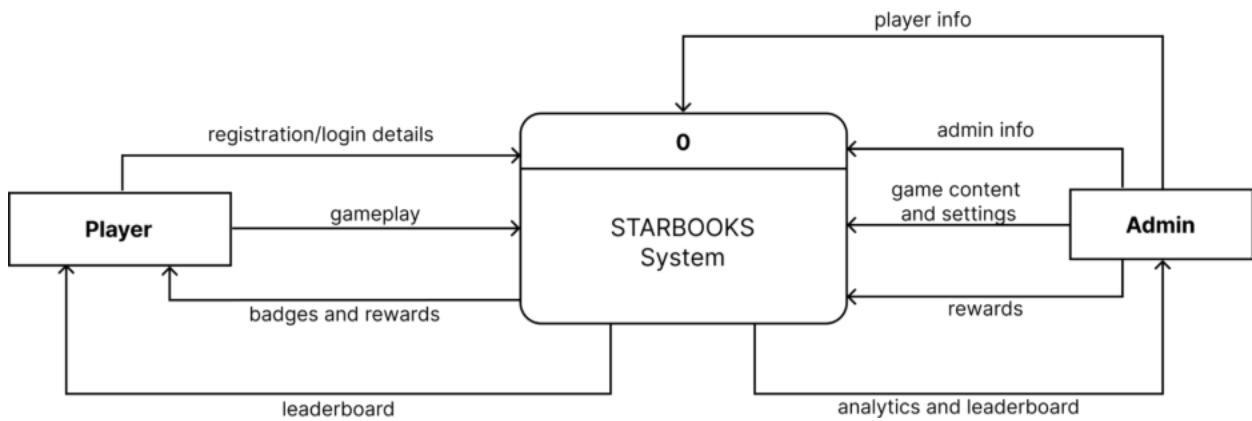
- **Add more cards for the Memory Match Game Mode** in each level to increase challenge.
- Include **categories for the Puzzle Game Mode** so players can choose themes.

- Incorporate a **racing-style visual while players compete in the 1v1 Battle Mode** to make the gameplay more exciting.

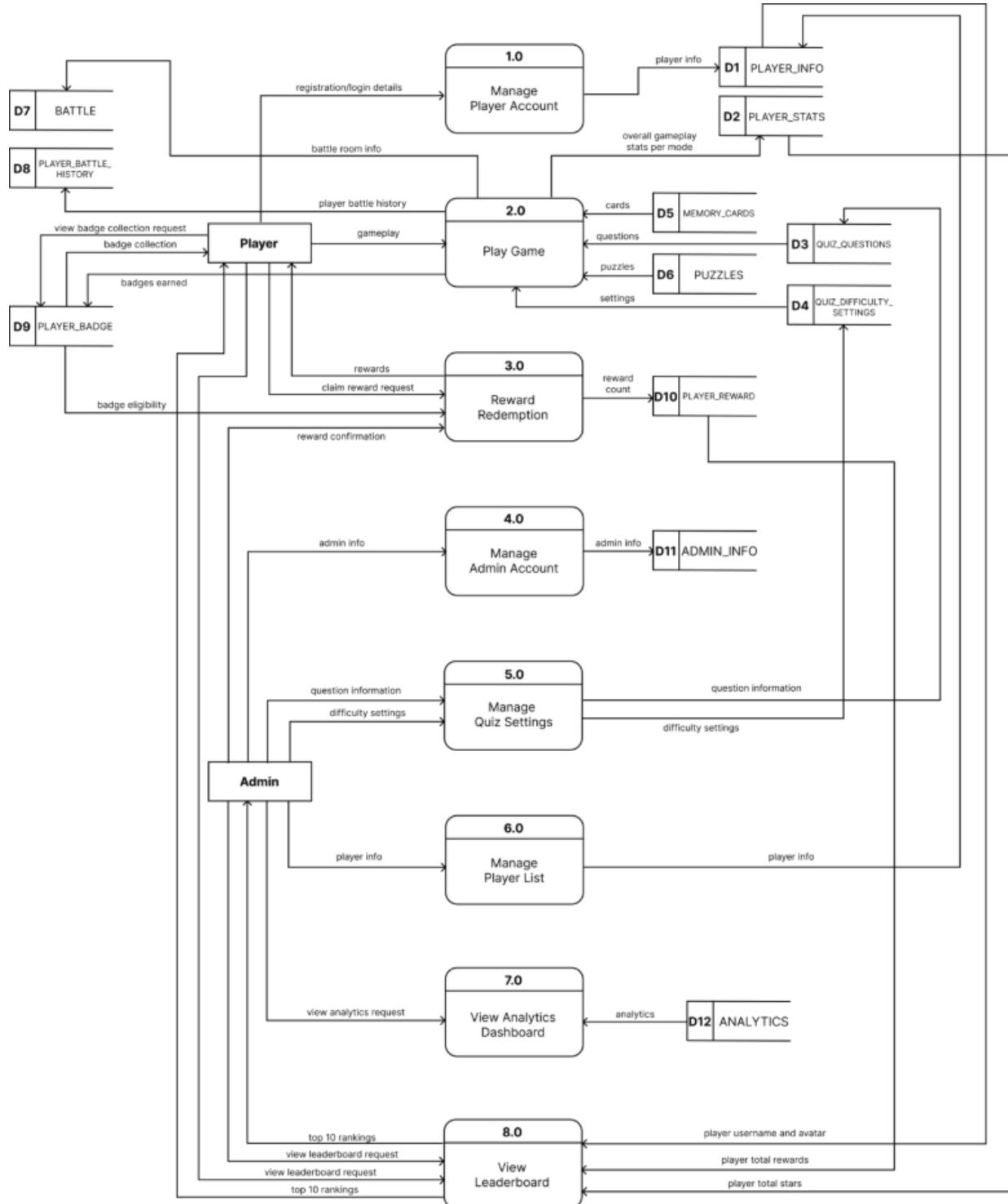
Additionally, Mr. Basaya provided the team with a folder containing all official assets, visuals, icons, and sound effects to be used in the final design. The team also conducted a screen recording of the current STARBOOKS app to study its existing layout and features, helping guide improvements and add more details to the mockup designs.

## IV. DATAFLOW DIAGRAMS

### 4.1 Level 0

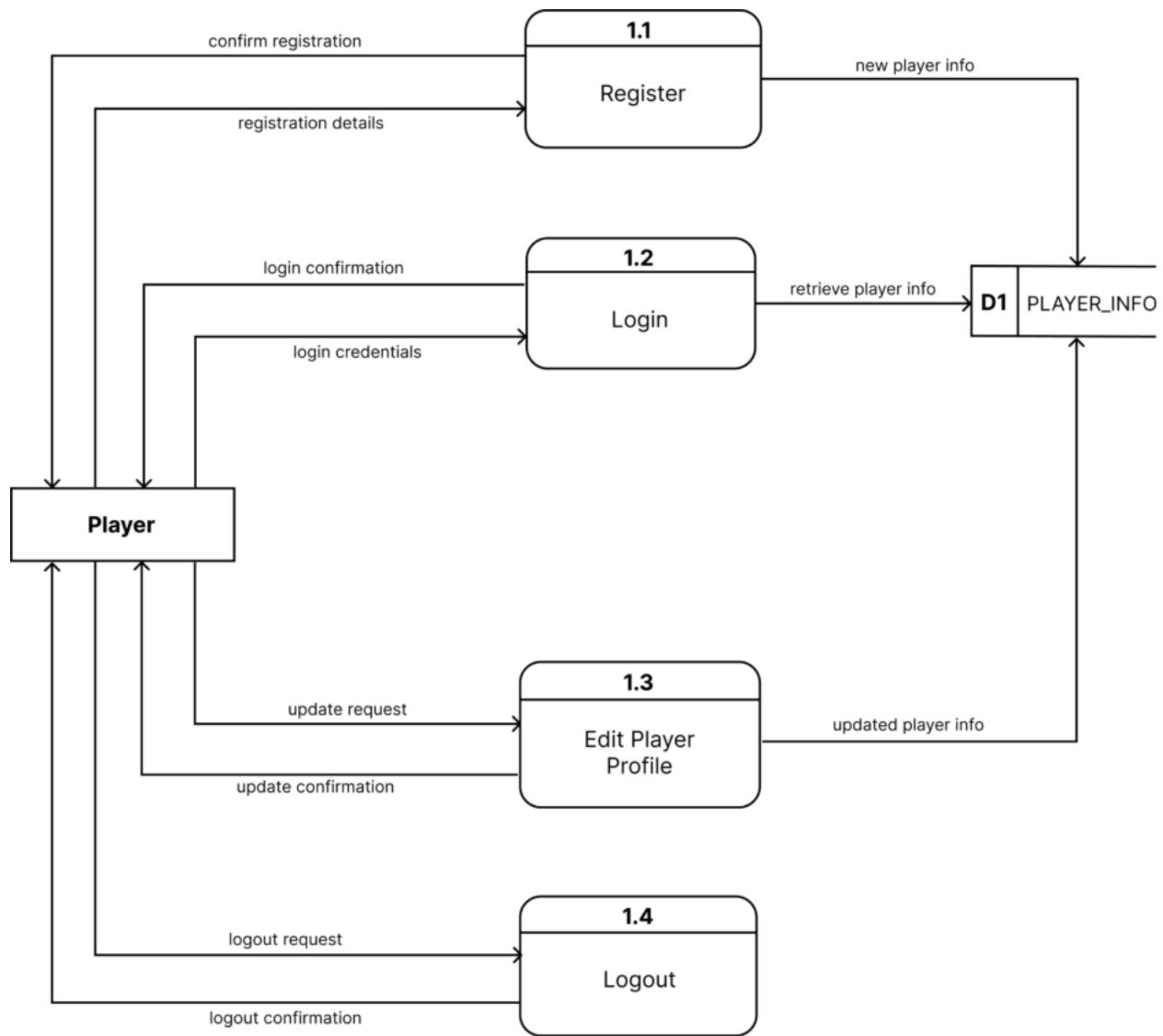


## 4.2 Level 1

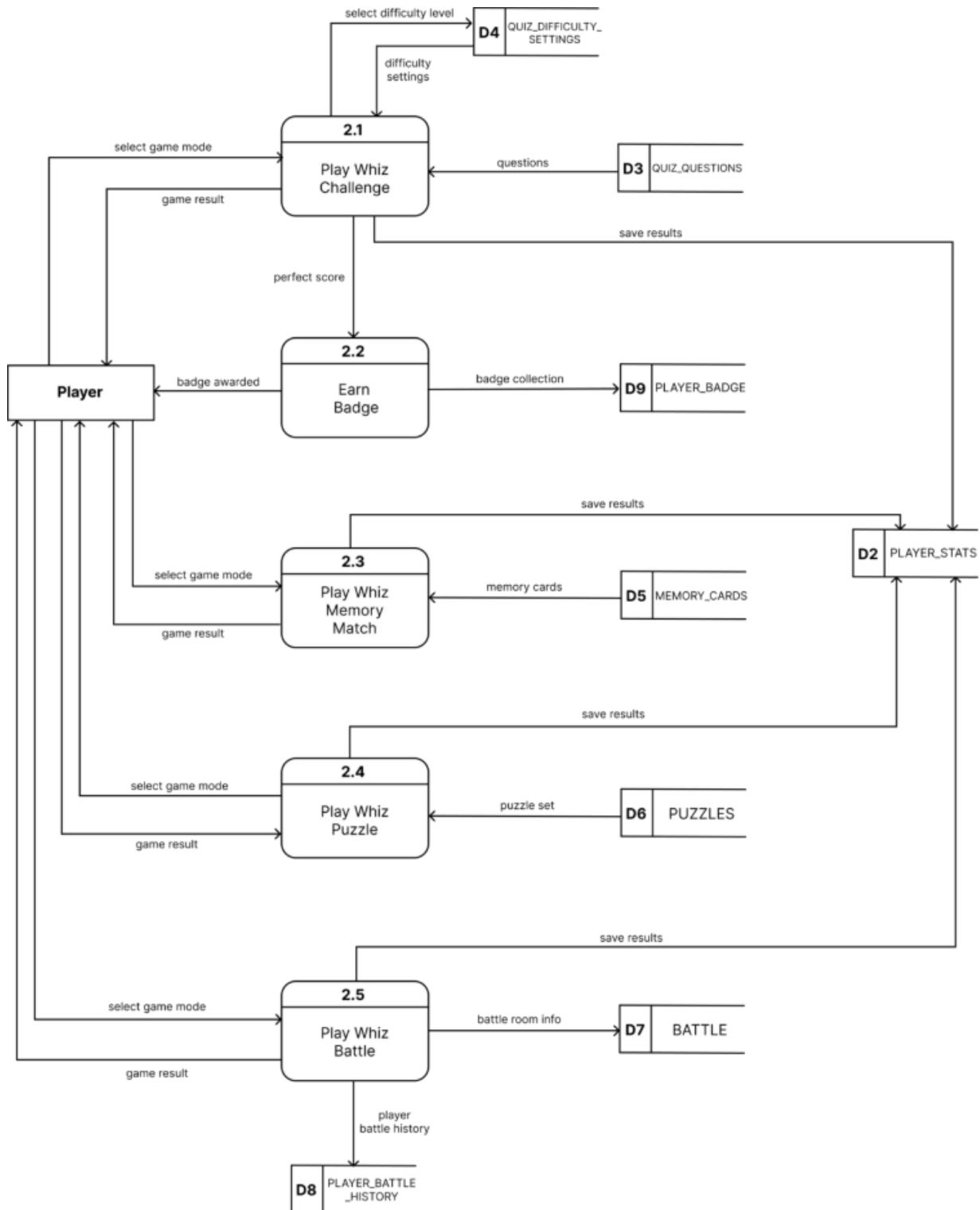


## 4.3 Level 2

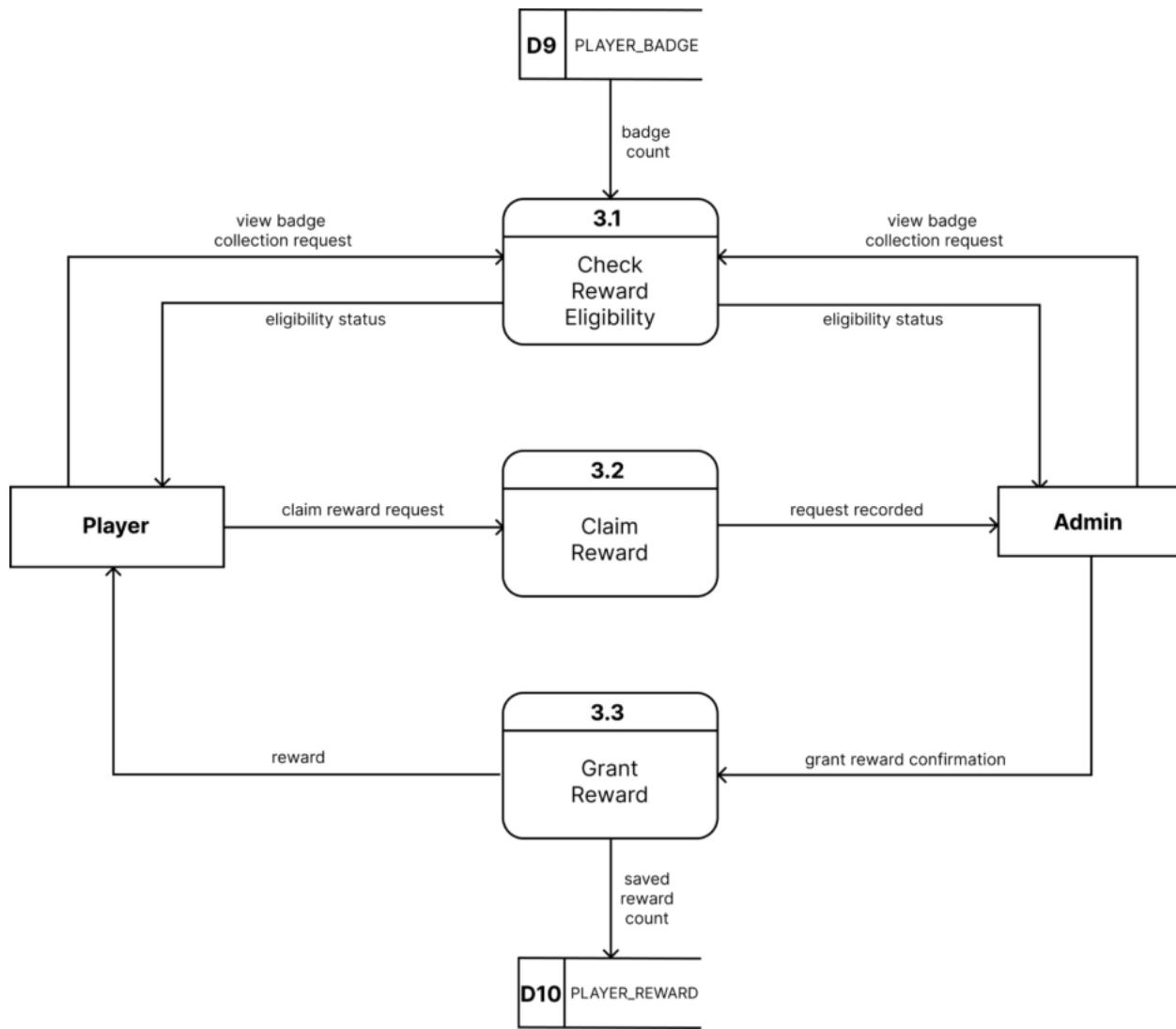
### 1.0 Manage Player Account



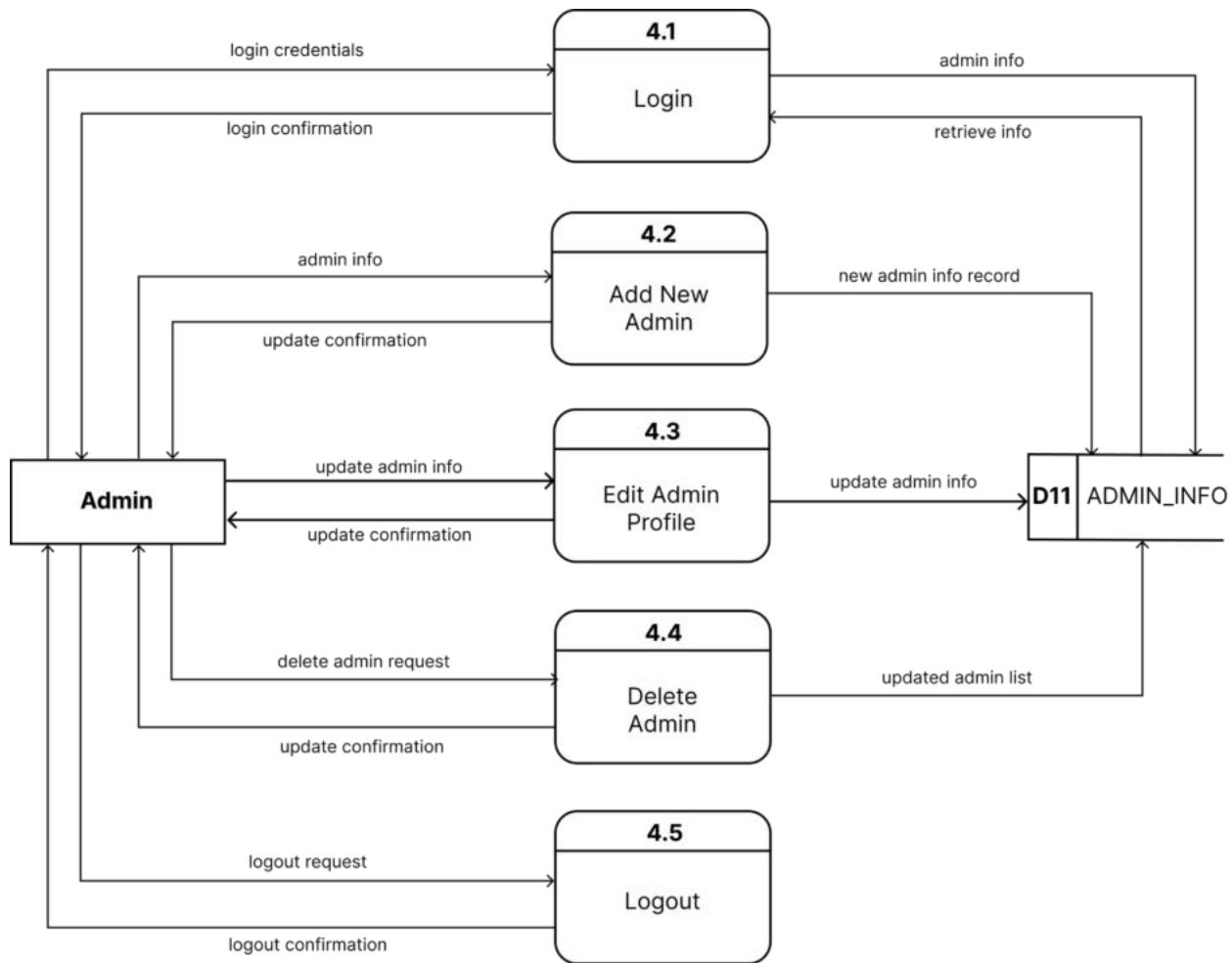
## 2.0 Play Game



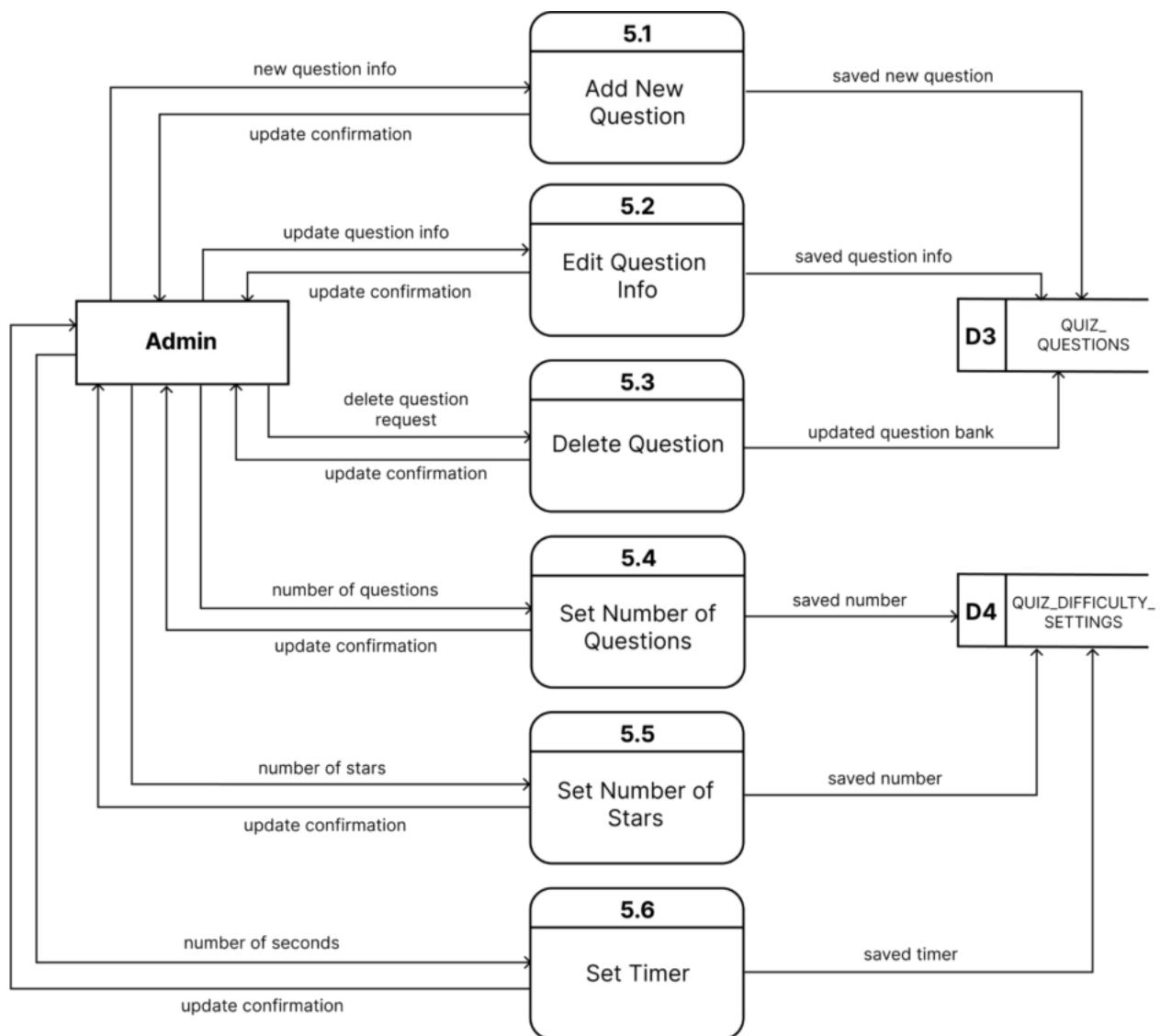
### 3.0 Reward Redemption



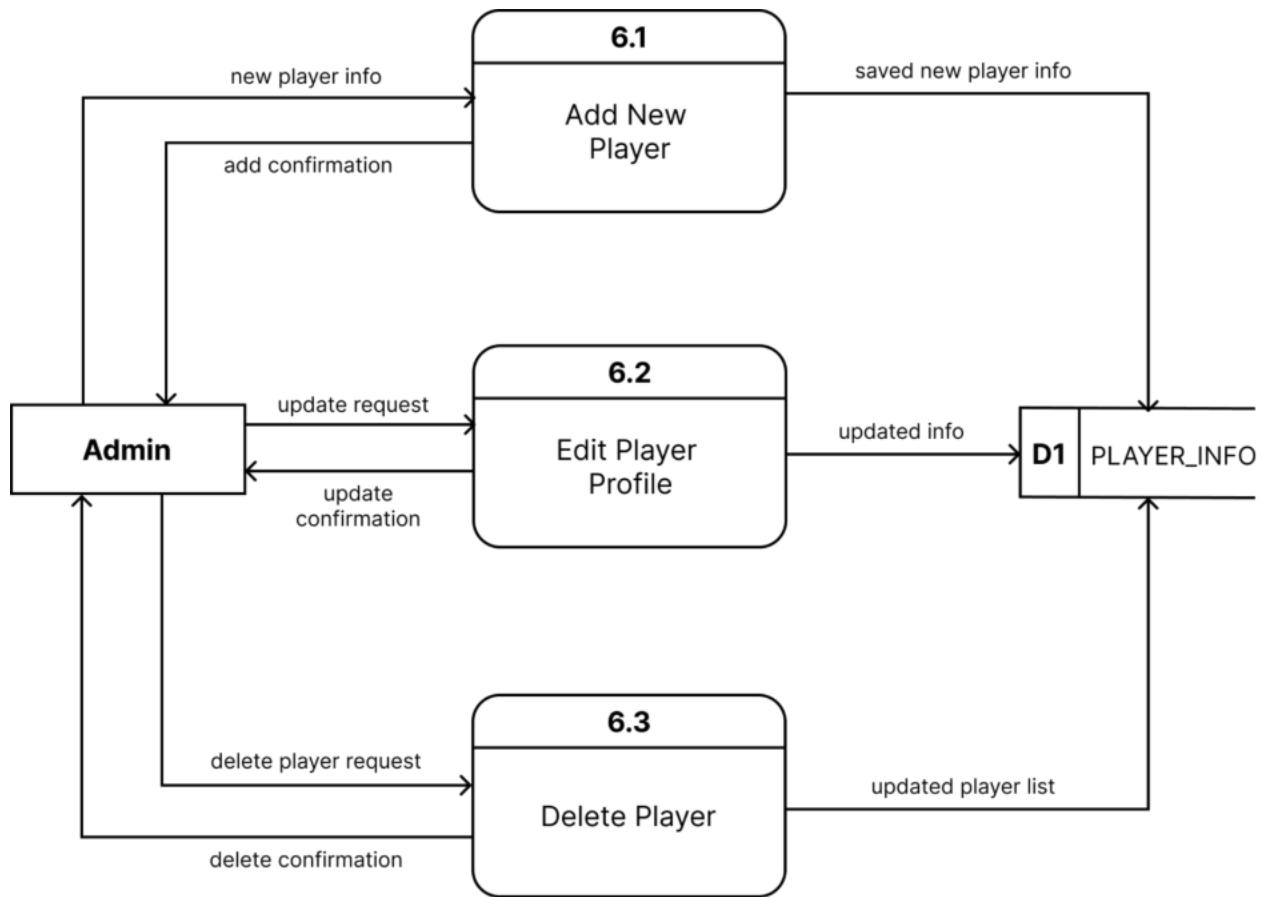
## 4.0 Manage Admin Account



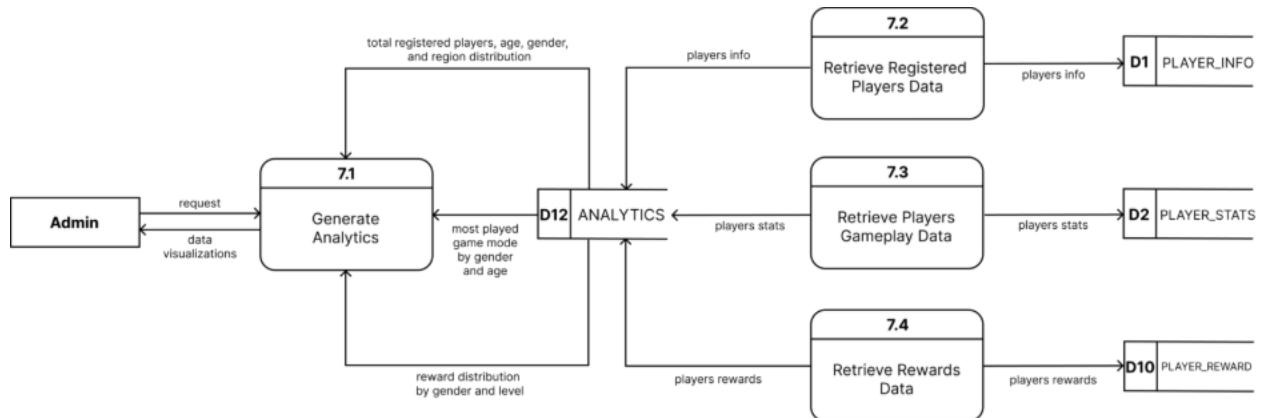
## 5.0 Manage Quiz Settings



## 6.0 Manage Player List



## 7.0 View Analytics Dashboard

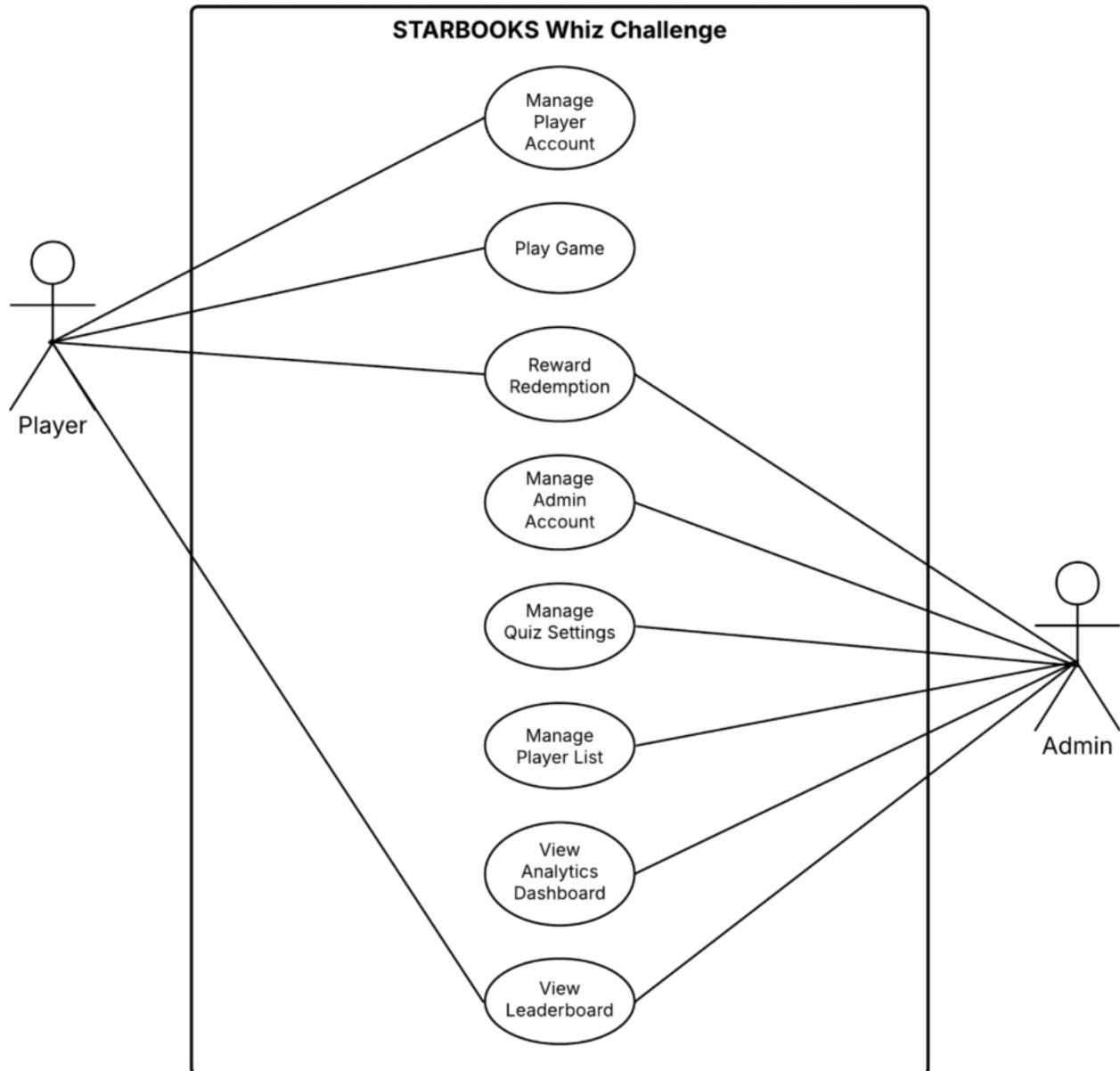


## 8.0 View Leaderboard



## V. USE CASE DOCUMENTATION

### 5.1 Use Case Diagram



## 5.2 Fully Dressed Use Cases

### Manage Player Account

<b>Use Case ID</b>	UC-MPA0
<b>Use Case Name</b>	Manage Player Account
<b>Created By</b>	Shandrae Lois Quianzon
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how a player manages their account in the system, including registering, logging in, updating details, changing password, tracking stats, and logging out.
<b>Priority</b>	High
<b>Primary Actor</b>	Player
<b>Secondary Actor</b>	System
<b>Preconditions</b>	1. Player must have access to the system.
<b>Postconditions</b>	1. Player account is created, updated, or accessed depending on the sub-function executed.
<b>Includes</b>	1. UC-MPA1 Player Register 2. UC-MPA2 Player Login 3. UC-MPA5 Player Logout
<b>Extends</b>	1. UC-MPA3 Edit Player Profile

<b>Use Case ID</b>	UC-MPA1
<b>Use Case Name</b>	Player Register
<b>Created By</b>	Shandrae Lois Quianzon
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how a player registers a new account in the system.
<b>Priority</b>	High
<b>Primary Actor</b>	Player
<b>Secondary Actor</b>	System
<b>Preconditions</b>	1. Player must have access to the system. 2. Player is not yet registered.
<b>Postconditions</b>	1. A new player account is created. 2. Player is redirected to the login page after successful registration.
<b>Main Flow</b>	1. Player selects “Register here” from the login page. 2. System displays registration form. 3. Player enters required details (username, password, school name, age, category, sex, region, province, city, avatar). 4. System validates the details.

	<p>5. System creates a new player account.</p> <p>6. System confirms successful registration.</p>
<b>Alternative Flows</b>	<p><b>3a. Invalid Registration Input</b></p> <p>1. Player enters incomplete or invalid details.</p> <p>2. System displays error message and prompts correction.</p>

<b>Use Case ID</b>	UC-MPA2
<b>Use Case Name</b>	Player Login
<b>Created By</b>	Shandrae Lois Quianzon
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how a player logs into the system using valid credentials.
<b>Priority</b>	High
<b>Primary Actor</b>	Player
<b>Secondary Actor</b>	System
<b>Preconditions</b>	1. Player must already be registered.
<b>Postconditions</b>	1. Player is logged into the system. 2. Player is redirected to the homepage.
<b>Main Flow</b>	1. System displays login form. 2. Player enters username and password. 3. System verifies credentials. 4. If valid, system logs in the player and displays the homepage.
<b>Alternative Flows</b>	<p><b>2a. Invalid Login</b></p> <p>1. Player enters incorrect username or password. 2. System rejects login attempt and displays error message.</p>

<b>Use Case ID</b>	UC-MPA3
<b>Use Case Name</b>	Edit Player Profile
<b>Created By</b>	Shandrae Lois Quianzon
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how a player updates their account information.
<b>Priority</b>	Medium
<b>Primary Actor</b>	Player
<b>Secondary Actor</b>	System
<b>Preconditions</b>	1. Player must be logged in.
<b>Postconditions</b>	1. Player profile information is updated successfully.
<b>Extends</b>	1. UC-MPA4 Change Player Password
<b>Main Flow</b>	1. Player selects "Edit Profile."

	<ol style="list-style-type: none"> <li>2. System displays editable fields.</li> <li>3. Player updates information and submits.</li> <li>4. System validates input and updates the profile.</li> <li>5. System confirms successful update.</li> </ol>
<b>Alternative Flows</b>	<p><b>3a. Invalid Profile Update</b></p> <ol style="list-style-type: none"> <li>1. Player enters invalid or empty fields.</li> <li>2. System displays validation error and requests correction.</li> </ol>

<b>Use Case ID</b>	UC-MPA4
<b>Use Case Name</b>	Change Player Password
<b>Created By</b>	Shandrae Lois Quianzon
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how a player changes their account password.
<b>Priority</b>	Medium
<b>Primary Actor</b>	Player
<b>Secondary Actor</b>	System
<b>Preconditions</b>	1. Player must be logged in.
<b>Postconditions</b>	1. Player's password is updated successfully.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Player selects "Change Password" from the Edit Profile page.</li> <li>2. System prompts for current password and new password.</li> <li>3. Player enters the required details.</li> <li>4. System verifies current password and validates new password format.</li> <li>5. System updates the password.</li> <li>6. System confirms successful password change.</li> </ol>
<b>Alternative Flows</b>	<p><b>3a. Wrong Current Password</b></p> <ol style="list-style-type: none"> <li>1. Player enters incorrect current password.</li> <li>2. System rejects request and asks to retry.</li> </ol> <p><b>3b. Weak New Password</b></p> <ol style="list-style-type: none"> <li>1. Player enters a password that does not meet security requirements.</li> <li>2. System prompts the player to enter a stronger password.</li> </ol>

<b>Use Case ID</b>	UC-MPA5
<b>Use Case Name</b>	Player Logout
<b>Created By</b>	Shandrae Lois Quianzon
<b>Date Created</b>	September 27, 2025

<b>Description</b>	This use case describes how a player logs out of the system.
<b>Priority</b>	Low
<b>Primary Actor</b>	Player
<b>Secondary Actor</b>	System
<b>Preconditions</b>	1. Player must be logged in.
<b>Postconditions</b>	1. Player is logged out of the system. 2. System redirects the player to the login page.
<b>Main Flow</b>	1. Player clicks avatar and selects "Logout." 2. System prompts the player with a confirmation dialog ("Are you sure you want to logout?"). 3. Player confirms logout. 4. System ends the session. 5. System redirects player to the login page.
<b>Alternative Flows</b>	<b>2a. Cancel Logout</b> 1. Player cancels at the confirmation prompt. 2. System keeps the player logged in and returns them to the previous page.

### Play Game

<b>Use Case ID</b>	UC-PG0
<b>Use Case Name</b>	Play Game
<b>Created By</b>	Arcielle Marie Gercan
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case represents the general action of playing a game in the STARBOOKS Whiz Challenge system. It is specialized into four different game modes: Whiz Challenge, Whiz Memory Match, Whiz Puzzle, and Whiz Battle.
<b>Priority</b>	High
<b>Primary Actor</b>	Player
<b>Secondary Actor</b>	System
<b>Preconditions</b>	1. Player must be logged in. 2. Player must have a valid account.
<b>Postconditions</b>	1. Player stats are updated based on gameplay. 2. Badges may be awarded depending on performance and game rules.
<b>Includes</b>	1. UC-PG1 Play Whiz Challenge 2. UC-PG2 Play Whiz Memory Match 3. UC-PG3 Play Whiz Puzzle 4. UC-PG4 Play Whiz Battle

<b>Use Case ID</b>	UC-PG1
<b>Use Case Name</b>	Play Whiz Challenge
<b>Created By</b>	Arcielle Marie Gercan
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how a player selects and plays the Whiz Challenge game mode, answering questions under a timer and potentially earning badges.
<b>Priority</b>	High
<b>Primary Actor</b>	Player
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Player must be logged in.</li> <li>2. Player must have a valid account.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Player stats are updated.</li> <li>2. A badge may be awarded if conditions are met.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Player selects "Whiz Challenge" on the homepage.</li> <li>2. System displays categories (Science or Math).</li> <li>3. Player chooses a category.</li> <li>4. System displays difficulty levels (Easy, Average, Difficult).</li> <li>5. Player chooses a difficulty level.</li> <li>6. System loads a set of questions based on the chosen category and difficulty.</li> <li>7. Player answers questions within the given timer.</li> </ol>
<b>Alternative Flows</b>	<p><b>7a. Time Expired</b></p> <ol style="list-style-type: none"> <li>1. Timer runs out before player answers all questions.</li> <li>2. System automatically submits current answers and calculates score.</li> </ol> <p><b>7b. Earn Badge</b></p> <ol style="list-style-type: none"> <li>1. Player answers all questions correctly.</li> <li>2. System awards a badge for that difficulty level.</li> <li>3. System updates badge collection in player profile.</li> </ol> <p><b>7c. No Badge</b></p> <ol style="list-style-type: none"> <li>1. Player fails to answer all questions correctly.</li> <li>2. System does not award a badge.</li> <li>3. System displays results and records stats.</li> </ol> <p><b>7d. Player Exits Mid-Game</b></p> <ol style="list-style-type: none"> <li>1. Player exits before completing all questions.</li> <li>2. System auto-submits current answers.</li> </ol>

	3. System calculates score based on answered questions and updates stats.
--	---

<b>Use Case ID</b>	UC-PG2
<b>Use Case Name</b>	Play Whiz Memory Match
<b>Created By</b>	Arcielle Marie Gercan
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how a player selects and plays the Whiz Memory Match game mode, matching pairs of cards under a timer.
<b>Priority</b>	High
<b>Primary Actor</b>	Player
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>1. Player must be logged in.</li> <li>2. Player must have a valid account.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>1. Stats are updated based on performance.</li> </ul>
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>1. Player selects “Whiz Memory Match” on the homepage.</li> <li>2. System displays difficulty levels (Easy, Average, Difficult).</li> <li>3. Player chooses a difficulty level.</li> <li>4. System loads the game board with cards based on difficulty.</li> <li>5. Player flips cards to find matches.</li> <li>6. System checks matches, updates score and timer.</li> <li>7. When game ends, system displays results and updates stats.</li> </ul>
<b>Alternative Flows</b>	<p><b>5a. Player Exits Mid-Game</b></p> <ul style="list-style-type: none"> <li>1. Player exits before timer ends.</li> <li>2. System ends the game and displays results. System updates stats.</li> </ul>

<b>Use Case ID</b>	UC-PG3
<b>Use Case Name</b>	Play Whiz Puzzle
<b>Created By</b>	Arcielle Marie Gercan
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how a player selects and plays the Whiz Puzzle game mode by arranging scrambled pieces to form a complete image.
<b>Priority</b>	High
<b>Primary Actor</b>	Player
<b>Secondary Actor</b>	System

<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Player must be logged in.</li> <li>2. Player must have a valid account.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Stats are updated based on performance.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Player selects “Whiz Puzzle” on the homepage.</li> <li>2. System displays available difficulty levels (Easy, Average, Difficult) and categories (Solar System, Scientists, The Human Body, Animals, Geometry, Random).</li> <li>3. System displays scrambled puzzle pieces based on chosen category and difficulty.</li> <li>4. Player arranges pieces to form the correct image.</li> <li>5. System validates arrangement and updates score.</li> <li>6. System displays results and updates stats.</li> </ol>
<b>Alternative Flows</b>	<p><b>4a. Player Exits Mid-Game</b></p> <ol style="list-style-type: none"> <li>1. Player exits before puzzle is completed.</li> <li>2. System ends the game and displays results.</li> <li>3. System updates stats.</li> </ol>

<b>Use Case ID</b>	UC-PG4
<b>Use Case Name</b>	Play Whiz Battle
<b>Created By</b>	Arcielle Marie Gercan
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how a player participates in Whiz Battle, either by creating or joining a battle, and competing against an opponent in real-time.
<b>Priority</b>	High
<b>Primary Actor</b>	Player
<b>Secondary Actors</b>	System, Opponent Player
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Player must be logged in.</li> <li>2. Player must have a valid account.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Stats are updated for both players.</li> <li>2. Winner is declared based on performance.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Player selects “Whiz Battle” on the homepage.</li> <li>2. System displays options (Create Battle or Join Battle).</li> </ol> <p><b>2.1. Create Battle</b></p> <ol style="list-style-type: none"> <li>2.1.1. Player chooses “Create Battle.”</li> <li>2.1.2. System displays categories (Science or Math) and difficulty levels (Easy, Average, Difficult).</li> <li>2.1.3. Player chooses a category and difficulty.</li> <li>2.1.4. System creates a battle room and displays the game code.</li> </ol>

	<p>2.1.5. Opponent joins the battle room.</p> <p><b>2.2 Join Battle</b></p> <p>2.2.1. Player enters the game code and clicks “Join.”</p> <p>2.2.2. Player starts the battle.</p> <p>2.2.3. System loads a set of questions based on chosen category and difficulty.</p> <p>2.2.4. Players answer questions in real time.</p> <p>2.2.5. System calculates stars and determines the winner.</p> <p>2.2.6. System updates both players’ stats.</p>
<b>Alternative Flows</b>	<p><b>2.1.5a. Opponent Fails to Join</b></p> <ol style="list-style-type: none"> <li>Opponent does not join.</li> <li>System prompts player to wait longer or cancel battle.</li> </ol> <p><b>2.2.1a. Invalid Code</b></p> <ol style="list-style-type: none"> <li>Player enters invalid or expired code.</li> <li>System displays error: “Invalid code. Please try again.”</li> </ol> <p><b>2.2.4a. Opponent Disconnects</b></p> <ol style="list-style-type: none"> <li>Opponent leaves mid-game.</li> <li>System ends match and declares remaining player winner.</li> <li>Stats are updated accordingly.</li> </ol> <p><b>2.2.4b. Player Exits Mid-Game</b></p> <ol style="list-style-type: none"> <li>Player quits during active battle.</li> <li>System ends session and declares opponent winner.</li> <li>Stats are updated for both players.</li> </ol>

### Reward Redemption

<b>Use Case ID</b>	UC-RR0
<b>Use Case Name</b>	Reward Redemption
<b>Created By</b>	Arcielle Marie Gercan
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how rewards are redeemed in the system. It includes both the player’s process of requesting a reward after earning the required badges, and the admin’s process of verifying and granting the reward request.
<b>Priority</b>	High

<b>Primary Actors</b>	Player, Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Player must be logged in.</li> <li>2. Player must have earned all required badges in a difficulty level.</li> <li>3. Admin must be logged in to process requests.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Player's reward claim is recorded and processed.</li> <li>2. Admin approval or rejection updates the player's profile and system logs.</li> </ol>
<b>Includes</b>	<ol style="list-style-type: none"> <li>1. UC-RR1 Player Requests Reward</li> <li>2. UC-RR2 Admin Grants Reward</li> </ol>

<b>Use Case ID</b>	UC-RR1
<b>Use Case Name</b>	Player Requests Reward
<b>Created By</b>	Arcielle Marie Gercan
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how a player requests to redeem a reward after meeting the badge requirements.
<b>Priority</b>	High
<b>Primary Actor</b>	Player
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Player must be logged in.</li> <li>2. Player must have earned all 3 badges in a difficulty level.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. System records the player's reward request for admin processing.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Player selects "Your Badges."</li> <li>2. System displays badge collection and shows eligibility to claim a reward per difficulty level.</li> <li>3. Player selects "Claim" on the chosen difficulty level.</li> <li>4. System validates eligibility (e.g., 3 badges earned in one difficulty).</li> <li>5. System records the request for admin processing.</li> </ol>
<b>Alternative Flows</b>	<p><b>1a. No Badges Yet</b></p> <ol style="list-style-type: none"> <li>1. Player opens "Your Badges" without having played any games.</li> <li>2. System displays empty badge slots.</li> </ol> <p><b>1b. Claim Button Locked</b></p> <ol style="list-style-type: none"> <li>1. Player sees "Locked" buttons instead of "Claim."</li> <li>2. Player cannot request to claim a reward until requirements are met.</li> </ol>

<b>Use Case ID</b>	UC-RR2
<b>Use Case Name</b>	Admin Grants Reward
<b>Created By</b>	Arcielle Marie Gercan
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin verifies a reward claim request and grants or rejects it.
<b>Priority</b>	High
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. Player must have submitted a reward request.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>1. Reward claim is processed.</li> <li>2. Player profile is updated to reflect claimed rewards.</li> <li>3. System logs the reward claim transaction.</li> </ul>
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>1. Admin navigates to “List of Players” in the admin panel.</li> <li>2. Admin selects the player profile and clicks “View” in the badges column.</li> <li>3. System displays the player’s badge collection with the “Reward” button unlocked (if eligible).</li> <li>4. Admin clicks “Reward.”</li> <li>5. System updates the badge collection of the chosen difficulty level, resetting it to no badges.</li> <li>6. System logs the reward claim transaction.</li> </ul>
<b>Alternative Flows</b>	<p><b>3a. Reward Button Locked</b></p> <ul style="list-style-type: none"> <li>1. Admin views player’s badge collection but the “Reward” button is locked.</li> <li>2. System indicates the player has not met eligibility.</li> <li>3. Admin cannot proceed until requirements are met.</li> </ul> <p><b>3b. Admin Cancels Reward Action</b></p> <ul style="list-style-type: none"> <li>1. Admin views eligible reward but chooses not to proceed.</li> <li>2. System retains current badge collection without resetting.</li> </ul>

## Manage Admin Account

<b>Use Case ID</b>	UC-MAA0
<b>Use Case Name</b>	Manage Admin Account
<b>Created By</b>	Kelly Dumbrique
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin manages their account in the system, including logging in, adding another account, updating details, changing password, deleting other account, and logging out.
<b>Priority</b>	High
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	1. Admin must have access to the system.
<b>Postconditions</b>	1. Admin account is created, updated, or accessed depending on the sub-function executed.
<b>Includes</b>	1. UC-MAA1 Admin Login 2. UC-MAA6 Admin Logout
<b>Extends</b>	1. UC-MAA2 Add New Admin 2. UC-MAA3 Edit Admin Profile 3. UC-MAA4 Change Admin Password 4. UC-MAA5 Delete Admin

<b>Use Case ID</b>	UC-MAA1
<b>Use Case Name</b>	Admin Login
<b>Created By</b>	Kelly Dumbrique
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin logs in to access the system.
<b>Priority</b>	High
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	1. Admin must already exist in the admin list.
<b>Postconditions</b>	1. Admin is logged in and system displays the admin dashboard.
<b>Main Flow</b>	1. System displays the login form. 2. Admin enters username and password. 3. System verifies credentials. 4. If valid, admin is logged in and system displays the admin dashboard.
<b>Alternative Flows</b>	<b>2a. Invalid Login</b> 1. Admin enters incorrect username or password. 2. System rejects login and displays error: "Invalid credentials. Please try again."

<b>Use Case ID</b>	UC-MAA2
<b>Use Case Name</b>	Add New Admin
<b>Created By</b>	Kelly Dumbrigue
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin adds a new administrator account.
<b>Priority</b>	High
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	1. Admin must be logged in.
<b>Postconditions</b>	1. A new admin account is created and added to the list.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to List of Admins.</li> <li>2. Admin clicks “Add New Admin.”</li> <li>3. System displays form for details (username, password, sex, image).</li> <li>4. Admin fills out details and submits.</li> <li>5. System validates input and creates the new account.</li> <li>6. System confirms successful addition.</li> </ol>
<b>Alternative Flows</b>	<p><b>4a. Invalid Input</b></p> <ol style="list-style-type: none"> <li>1. Admin leaves required fields blank or enters invalid details.</li> <li>2. System displays error and requests correction.</li> </ol> <p><b>4b. Duplicate Username</b></p> <ol style="list-style-type: none"> <li>1. Admin enters a username that already exists. System rejects submission and prompts for a unique username.</li> </ol>

<b>Use Case ID</b>	UC-MAA3
<b>Use Case Name</b>	Edit Admin Profile
<b>Created By</b>	Kelly Dumbrigue
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin edits an existing administrator account.
<b>Priority</b>	High
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	1. Admin must be logged in.
<b>Postconditions</b>	1. The selected admin profile is updated.

<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to List of Admins.</li> <li>2. Admin selects an admin and clicks Edit icon.</li> <li>3. System displays editable fields.</li> <li>4. Admin updates information and submits.</li> <li>5. System validates and updates the profile.</li> <li>6. System confirms successful update.</li> </ol>
<b>Alternative Flows</b>	<p><b>4a. Invalid Profile Update</b></p> <ol style="list-style-type: none"> <li>1. Admin enters invalid or empty fields.</li> <li>2. System displays validation error and requests correction.</li> </ol>

<b>Use Case ID</b>	UC-MAA4
<b>Use Case Name</b>	Change Admin Password
<b>Created By</b>	Kelly Dumbrigue
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin changes their account password.
<b>Priority</b>	Medium
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin must be logged in.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Admin's password is updated successfully.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to List of Admins.</li> <li>2. Admin selects an admin and clicks Password icon.</li> <li>3. System prompts for current password and new password.</li> <li>4. Admin enters the required details.</li> <li>5. System verifies current password and validates new password format.</li> <li>6. System updates the password.</li> <li>7. System confirms successful password change.</li> </ol>
<b>Alternative Flows</b>	<p><b>3a. Wrong Current Password</b></p> <ol style="list-style-type: none"> <li>1. Admin enters incorrect current password.</li> <li>2. System rejects request and asks to retry.</li> </ol> <p><b>3b. Weak New Password</b></p> <ol style="list-style-type: none"> <li>1. Admin enters a password that does not meet security requirements.</li> <li>2. System prompts the admin to enter a stronger password.</li> </ol>

<b>Use Case ID</b>	UC-MAA5
<b>Use Case Name</b>	Delete Admin
<b>Created By</b>	Kelly Dumbrique
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin deletes an administrator account.
<b>Priority</b>	High
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	1. Admin must be logged in.
<b>Postconditions</b>	1. The selected admin account is removed from the list.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to List of Admins.</li> <li>2. Admin selects an admin and clicks Delete icon.</li> <li>3. System prompts confirmation.</li> <li>4. Admin confirms deletion.</li> <li>5. System removes account and updates the list.</li> </ol>
<b>Alternative Flows</b>	<p><b>3a. Cancel Delete</b></p> <ol style="list-style-type: none"> <li>1. Admin clicks Delete but cancels at confirmation prompt.</li> <li>2. System keeps the account unchanged.</li> </ol> <p><b>4a. Attempt to Delete Self</b></p> <ol style="list-style-type: none"> <li>1. Admin tries to delete their own account while logged in.</li> <li>2. System rejects action with message: "You cannot delete your own account while logged in."</li> </ol>

<b>Use Case ID</b>	UC-MAA6
<b>Use Case Name</b>	Admin Logout
<b>Created By</b>	Kelly Dumbrique
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin logs out of the system.
<b>Priority</b>	High
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	1. Admin must be logged in.
<b>Postconditions</b>	1. Admin is logged out and redirected to the login page.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin clicks the avatar and selects "Logout."</li> <li>2. System ends the session.</li> <li>3. System redirects admin to login page.</li> </ol>

## Manage Quiz Settings

<b>Use Case ID</b>	UC-MQS0
<b>Use Case Name</b>	Manage Quiz Settings
<b>Created By</b>	Kelly Dumbrique
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin manages the quiz content and difficulty settings in the system. It includes adding, editing, and deleting questions in the question bank, as well as configuring quiz parameters such as the number of questions, star values, and timers per difficulty level.
<b>Priority</b>	Medium
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. Question bank and difficulty levels must exist in the database.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The question bank and quiz settings are updated based on the admin's actions.</li> </ol>
<b>Includes</b>	<ol style="list-style-type: none"> <li>1. UC-MQS1 Add New Question</li> <li>2. UC-MQS2 Edit Question Info</li> <li>3. UC-MQS3 Delete Question</li> <li>4. UC-MQS4 Set Number of Questions</li> <li>5. UC-MQS5 Set Stars</li> <li>6. UC-MQS6 Set Timer</li> </ol>

<b>Use Case ID</b>	UC-MQS1
<b>Use Case Name</b>	Add New Question
<b>Created By</b>	Kelly Dumbrique
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin adds a new quiz question into the question bank.
<b>Priority</b>	Medium
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. Question bank exists in the database.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. A new question is added to the question bank.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to Questions.</li> <li>2. Admin clicks "Add New Question."</li> <li>3. System displays a form for question details (text, options, correct answer, category, year level, difficulty, image).</li> </ol>

	<p>4. Admin fills out the form and submits.</p> <p>5. System validates input.</p> <p>6. System saves the new question.</p> <p>7. System confirms successful addition and updates the list.</p>
<b>Alternative Flows</b>	<p><b>4a. Invalid Input</b></p> <ol style="list-style-type: none"> <li>1. Admin leaves required fields blank or enters invalid data.</li> <li>2. System displays error message and requests correction.</li> </ol>

<b>Use Case ID</b>	UC-MQS2
<b>Use Case Name</b>	Edit Question Info
<b>Created By</b>	Kelly Dumbrigue
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin edits the details of an existing quiz question.
<b>Priority</b>	Medium
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. At least one question exists.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The selected question is updated with new details.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to Questions.</li> <li>2. Admin selects an existing question and clicks Edit.</li> <li>3. System displays editable fields.</li> <li>4. Admin updates information and submits.</li> <li>5. System validates and updates the question.</li> <li>6. System confirms successful update.</li> </ol>
<b>Alternative Flows</b>	<p><b>4a. Invalid Update</b></p> <ol style="list-style-type: none"> <li>1. Admin provides invalid or incomplete information.</li> <li>2. System displays validation error and requests correction.</li> </ol>

<b>Use Case ID</b>	UC-MQS3
<b>Use Case Name</b>	Delete Question
<b>Created By</b>	Kelly Dumbrigue
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin deletes an existing quiz question.
<b>Priority</b>	Medium
<b>Primary Actor</b>	Admin

<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. At least one question exists.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The selected question is removed from the database.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to Questions.</li> <li>2. Admin selects a question and clicks Delete.</li> <li>3. System prompts confirmation.</li> <li>4. Admin confirms deletion.</li> <li>5. System removes the question and updates the list.</li> </ol>
<b>Alternative Flows</b>	<p><b>3a. Cancel Delete</b></p> <ol style="list-style-type: none"> <li>1. Admin cancels at the confirmation prompt.</li> <li>2. System keeps the question unchanged.</li> </ol>

<b>Use Case ID</b>	UC-MQS4
<b>Use Case Name</b>	Set Number of Questions
<b>Created By</b>	Kelly Dumbrique
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin sets the number of questions per difficulty level.
<b>Priority</b>	Medium
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. Difficulty levels exist in the database.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Number of questions per difficulty level is updated.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to Difficulty settings.</li> <li>2. System displays current number of questions per level.</li> <li>3. Admin specifies new number of questions.</li> <li>4. System validates input.</li> <li>5. System updates the setting and confirms the change.</li> </ol>
<b>Alternative Flows</b>	<p><b>2a. Insufficient Questions</b></p> <ol style="list-style-type: none"> <li>1. Admin specifies more than available in the database.</li> <li>2. System displays warning: "Not enough questions available. Please reduce the number."</li> </ol> <p><b>3a. Invalid Number</b></p> <ol style="list-style-type: none"> <li>1. Admin enters non-numeric or negative value.</li> <li>2. System rejects input and prompts for a valid number.</li> </ol>

<b>Use Case ID</b>	UC-MQS5
<b>Use Case Name</b>	Set Stars
<b>Created By</b>	Kelly Dumbrique
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin sets the star value awarded for correct answers per difficulty level.
<b>Priority</b>	Medium
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. Difficulty levels exist in the database.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Star values are updated per difficulty level.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to Difficulty settings.</li> <li>2. System displays current star values.</li> <li>3. Admin specifies new star values.</li> <li>4. System updates and confirms the change.</li> </ol>
<b>Alternative Flows</b>	<p><b>3a. Invalid Star Value</b></p> <ol style="list-style-type: none"> <li>1. Admin enters invalid star values (negative or zero).</li> <li>2. System rejects input and prompts for a valid value.</li> </ol>

<b>Use Case ID</b>	UC-MQS6
<b>Use Case Name</b>	Set Timer
<b>Created By</b>	Kelly Dumbrique
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin sets the time limit per difficulty level.
<b>Priority</b>	Medium
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. Difficulty levels exist in the database.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Timer settings are updated per difficulty level.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to Difficulty settings.</li> <li>2. System displays current timer values.</li> <li>3. Admin specifies a new time limit.</li> <li>4. System updates and confirms the change.</li> </ol>
<b>Alternative Flows</b>	<p><b>3a. Invalid Timer Value</b></p> <ol style="list-style-type: none"> <li>1. Admin enters invalid timer values (negative, zero, or excessively high).</li> <li>2. System rejects input and prompts for a valid time.</li> </ol>

## Manage Player List

<b>Use Case ID</b>	UC-MPL0
<b>Use Case Name</b>	Manage Player List
<b>Created By</b>	Shandrae Lois Quianzon
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin manages the player list in the system. It includes adding new players, editing player profiles, changing player passwords, and deleting player accounts.
<b>Priority</b>	Medium
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. Player list exists in the database.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>1. Player list is updated based on the action performed (add, edit, change password, or delete).</li> </ul>
<b>Includes</b>	<ul style="list-style-type: none"> <li>1. UC-MPL1 Add New Player</li> <li>2. UC-MPL2 Edit Player Profile</li> <li>3. UC-MPL3 Change Player Password</li> <li>4. UC-MPL4 Delete Player</li> </ul>

<b>Use Case ID</b>	UC-MPL1
<b>Use Case Name</b>	Add New Player
<b>Created By</b>	Shandrae Lois Quianzon
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin adds a new player into the system.
<b>Priority</b>	Medium
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. Player list exists in the database.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>1. A new player is added to the player list.</li> </ul>
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>1. Admin navigates to Player List.</li> <li>2. System displays the list of players.</li> <li>3. Admin clicks “Add New Player.”</li> <li>4. System displays a form for player details (username, password, school, age, category, sex, region, province, city, avatar).</li> <li>5. Admin fills out the form and submits.</li> <li>6. System validates input.</li> <li>7. If valid, system saves the new player.</li> </ul>

	<p>8. System confirms successful addition and updates the list.</p>
<b>Alternative Flows</b>	<p><b>5a. Invalid Input</b></p> <ol style="list-style-type: none"> <li>1. Admin leaves required fields blank or enters invalid data.</li> <li>2. System displays error message and requests correction.</li> </ol> <p><b>5b. Duplicate Username</b></p> <ol style="list-style-type: none"> <li>1. Admin enters a username that already exists.</li> <li>2. System rejects submission and prompts for a unique username.</li> </ol>

<b>Use Case ID</b>	UC-MPL2
<b>Use Case Name</b>	Edit Player Profile
<b>Created By</b>	Shandrae Lois Quianzon
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin edits the details of an existing player profile.
<b>Priority</b>	Medium
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. At least one player exists.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Player profile information is updated.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to Player List.</li> <li>2. Admin selects an existing player and clicks Edit.</li> <li>3. System displays editable fields.</li> <li>4. Admin updates information and submits.</li> <li>5. System validates and updates the profile.</li> <li>6. System confirms successful update.</li> </ol>
<b>Alternative Flows</b>	<p><b>4a. Invalid Update</b></p> <ol style="list-style-type: none"> <li>1. Admin enters invalid or incomplete data.</li> <li>2. System displays validation error and requests correction.</li> </ol>

<b>Use Case ID</b>	UC-MPL3
<b>Use Case Name</b>	Change Player Password
<b>Created By</b>	Shandrae Lois Quianzon
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin resets or changes a player's password.
<b>Priority</b>	Medium

<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. At least one player exists.</li> </ol>
<b>Postconditions</b>	1. Player password is updated in the system.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to Player List.</li> <li>2. System prompts for current password and new password.</li> <li>3. Admin enters the required details with the help of the Player.</li> <li>4. System verifies current password and validates new password format.</li> <li>5. System updates the password.</li> <li>6. System confirms successful password change.</li> </ol>
<b>Alternative Flows</b>	<p><b>3a. Wrong Current Password</b></p> <ol style="list-style-type: none"> <li>1. Admin enters incorrect current password.</li> <li>2. System rejects request and asks to retry.</li> </ol> <p><b>3b. Weak Password</b></p> <ol style="list-style-type: none"> <li>1. Admin enters a password that does not meet security requirements.</li> <li>2. System rejects and prompts for a stronger password.</li> </ol>

<b>Use Case ID</b>	UC-MPL4
<b>Use Case Name</b>	Delete Player
<b>Created By</b>	Shandrae Lois Quianzon
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin deletes a player account from the system.
<b>Priority</b>	Medium
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. At least one player exists.</li> </ol>
<b>Postconditions</b>	1. The selected player account is removed from the system.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to Player List.</li> <li>2. Admin selects a player and clicks Delete.</li> <li>3. System prompts confirmation.</li> <li>4. Admin confirms deletion.</li> <li>5. System removes the player and updates the list.</li> </ol>
<b>Alternative Flows</b>	<p><b>3a. Cancel Delete</b></p> <ol style="list-style-type: none"> <li>1. Admin cancels at the confirmation prompt.</li> </ol>

	2. System keeps the player account unchanged.
--	---

### View Analytics Dashboard

<b>Use Case ID</b>	UC-VAD1
<b>Use Case Name</b>	View Analytics Dashboard
<b>Created By</b>	Janice Maxene Salipande
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how the admin views and analyzes player and game-related data through the Analytics Dashboard. The dashboard provides insights on trends, player demographics, game performance, and reward claims.
<b>Priority</b>	High
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. Analytics data must be available in the database.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Admin views and interprets analytics reports.</li> <li>2. Data insights are displayed but no changes are made to the database.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to the Analytics Dashboard.</li> <li>2. System displays dashboard sections, which may include: <ul style="list-style-type: none"> <li>a. Total Registered Players</li> <li>b. Most Played Game Mode</li> <li>c. Average Session Duration</li> <li>d. Male vs. Female Registered Players (pie chart)</li> <li>e. Age Distribution of Players (bar chart)</li> <li>f. Registered Players by Region (bar chart)</li> <li>g. Male vs. Female Players Per Game Mode (bar chart)</li> <li>h. Reward Distribution by Gender and Level (bar chart)</li> <li>i. Most Played Game Mode by Age (bar chart)</li> </ul> </li> <li>3. Admin interacts with sort/filter options.</li> <li>4. System updates and redisplays graphs based on selected filters.</li> <li>5. Admin views insights and trends.</li> <li>6. Admin clicks Export icon.</li> <li>7. System generates an Excel file and prompts the admin to download it.</li> </ol>
<b>Alternative Flows</b>	<b>2a. No Data Available</b>

	<p>1. System detects no available data (e.g., new system with no players yet).</p> <p>2. System displays an empty state message such as “No analytics data available yet.”</p> <p><b>4a. Invalid Filter Selection</b></p> <ol style="list-style-type: none"> <li>1. Admin applies a filter combination that returns no results (e.g., selecting a date range before launch).</li> <li>2. System displays “No results found for the selected criteria.”</li> </ol> <p><b>4b. System Error During Data Load</b></p> <ol style="list-style-type: none"> <li>1. If the dashboard fails to retrieve analytics data due to a system error.</li> <li>2. System displays an error message and prompts the admin to retry.</li> </ol> <p><b>6a. Export Fails</b></p> <ol style="list-style-type: none"> <li>1. If the system encounters an error during Excel export (e.g., file generation issue).</li> <li>2. System displays “Export failed. Please try again.”</li> </ol>
--	--

### View Leaderboard

<b>Use Case ID</b>	UC-VLB0
<b>Use Case Name</b>	View Leaderboard
<b>Created By</b>	Janice Maxene Salipande
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how the leaderboard is viewed within the system. It includes both the player's ability to view the Top 10 leaderboard in the game interface and the admin's ability to view the same leaderboard data in the admin panel. The leaderboard displays the top-performing players based on total stars or rewards earned.
<b>Priority</b>	High
<b>Primary Actors</b>	Player, Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Player must be logged in.</li> <li>2. Admin must be logged in to access the admin panel.</li> <li>3. Leaderboard data must be available in the system database.</li> </ol>

<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Leaderboard data is successfully displayed to the actor.</li> <li>2. Admin may export the leaderboard data to Excel.</li> </ol>
<b>Includes</b>	<ol style="list-style-type: none"> <li>1. UC-VLB1 Player View Leaderboard</li> <li>2. UC-VLB2 Admin View Leaderboard</li> </ol>

<b>Use Case ID</b>	UC-VLB1
<b>Use Case Name</b>	Player View Leaderboard
<b>Created By</b>	Janice Maxene Salipande
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how a player views the leaderboard for Whiz Challenge and Whiz Battle game modes.
<b>Priority</b>	High
<b>Primary Actor</b>	Player
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Player must be logged in.</li> <li>2. At least one player must have played a game to generate leaderboard data.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Leaderboard data is displayed.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Player navigates to the Leaderboard page.</li> <li>2. System displays two leaderboard sections:             <ol style="list-style-type: none"> <li>a. Whiz Challenge Leaderboard (Top 10)                     <ol style="list-style-type: none"> <li>i. Username</li> <li>ii. Total Rewards for All Levels</li> <li>iii. Rewards for Easy, Average, Difficult</li> <li>iv. Last Reward Claim date/time</li> </ol> </li> <li>b. Whiz Battle Leaderboard (Top 10)                     <ol style="list-style-type: none"> <li>i. Username</li> <li>ii. Total Stars</li> <li>iii. Last Battle date/time</li> </ol> </li> </ol> </li> <li>3. Player reviews leaderboard standings.</li> </ol>
<b>Alternative Flows</b>	<p><b>2a. No Leaderboard Data</b></p> <ol style="list-style-type: none"> <li>1. System detects that no players have participated yet in either game mode.</li> <li>2. System displays “No leaderboard data available yet.”</li> </ol> <p><b>2b. System Error While Loading Leaderboard</b></p> <ol style="list-style-type: none"> <li>1. If the system fails to fetch leaderboard data due to a technical issue.</li> <li>2. System displays an error message and prompts the admin to retry.</li> </ol>

<b>Use Case ID</b>	UC-VLB2
<b>Use Case Name</b>	Admin View Leaderboard
<b>Created By</b>	Janice Maxene Salipande
<b>Date Created</b>	September 27, 2025
<b>Description</b>	This use case describes how an admin views the leaderboard for Whiz Challenge and Whiz Battle game modes.
<b>Priority</b>	High
<b>Primary Actor</b>	Admin
<b>Secondary Actor</b>	System
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin must be logged in.</li> <li>2. At least one player must have played a game to generate leaderboard data.</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Leaderboard data is displayed based on selected filters.</li> <li>2. Admin may export the leaderboard data to Excel.</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to the Leaderboard page.</li> <li>2. System displays two leaderboard sections: <ul style="list-style-type: none"> <li>a. Whiz Challenge Leaderboard (Top 10) <ul style="list-style-type: none"> <li>i. Username</li> <li>ii. Total Rewards for All Levels</li> <li>iii. Rewards for Easy, Average, Difficult</li> <li>iv. Last Reward Claim date/time</li> </ul> </li> <li>b. Whiz Battle Leaderboard (Top 10) <ul style="list-style-type: none"> <li>i. Username</li> <li>ii. Total Stars</li> <li>iii. Last Battle date/time</li> </ul> </li> </ul> </li> <li>3. Admin reviews leaderboard standings.</li> <li>2. Admin clicks Export option.</li> <li>3. System generates an Excel file containing leaderboard data and prompts the admin to download it.</li> </ol>
<b>Alternative Flows</b>	<p><b>2a. No Leaderboard Data</b></p> <ol style="list-style-type: none"> <li>1. System detects that no players have participated yet in either game mode.</li> <li>2. System displays “No leaderboard data available yet.”</li> </ol> <p><b>2b. System Error While Loading Leaderboard</b></p> <ol style="list-style-type: none"> <li>3. If the system fails to fetch leaderboard data due to a technical issue.</li> <li>4. System displays an error message and prompts the admin to retry.</li> </ol>

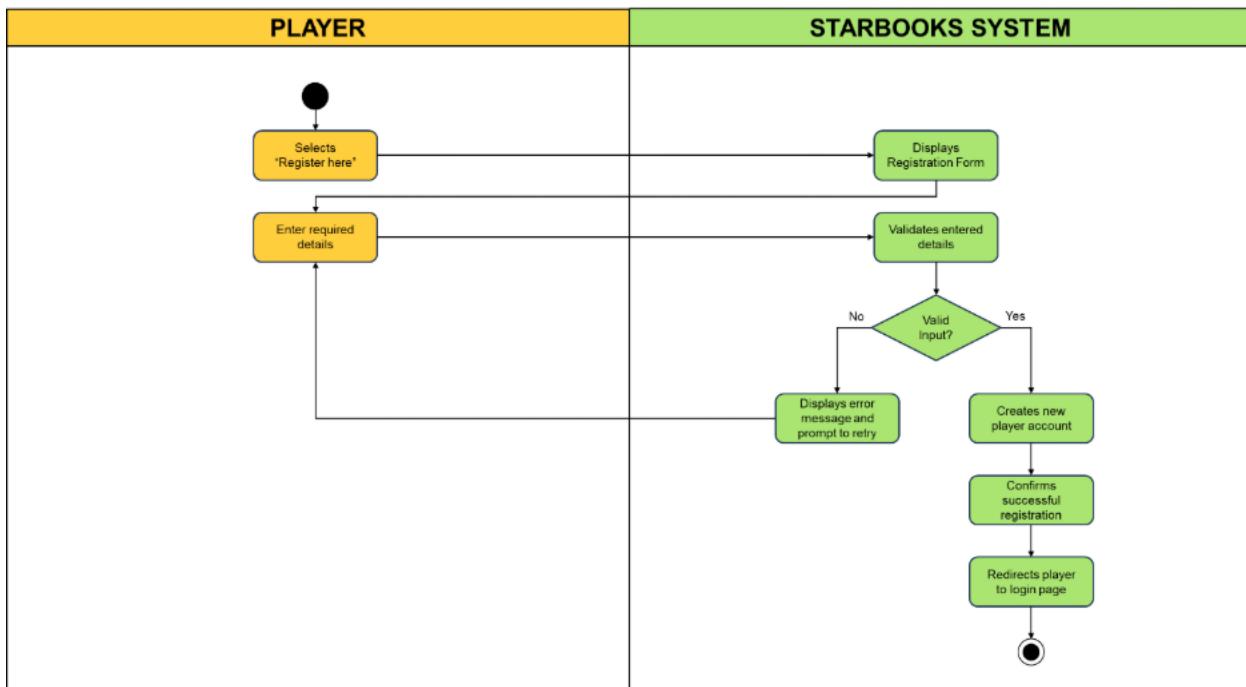
#### 4a. Export Fails

1. If the system encounters an error during Excel export.
2. System displays "Export failed. Please try again."

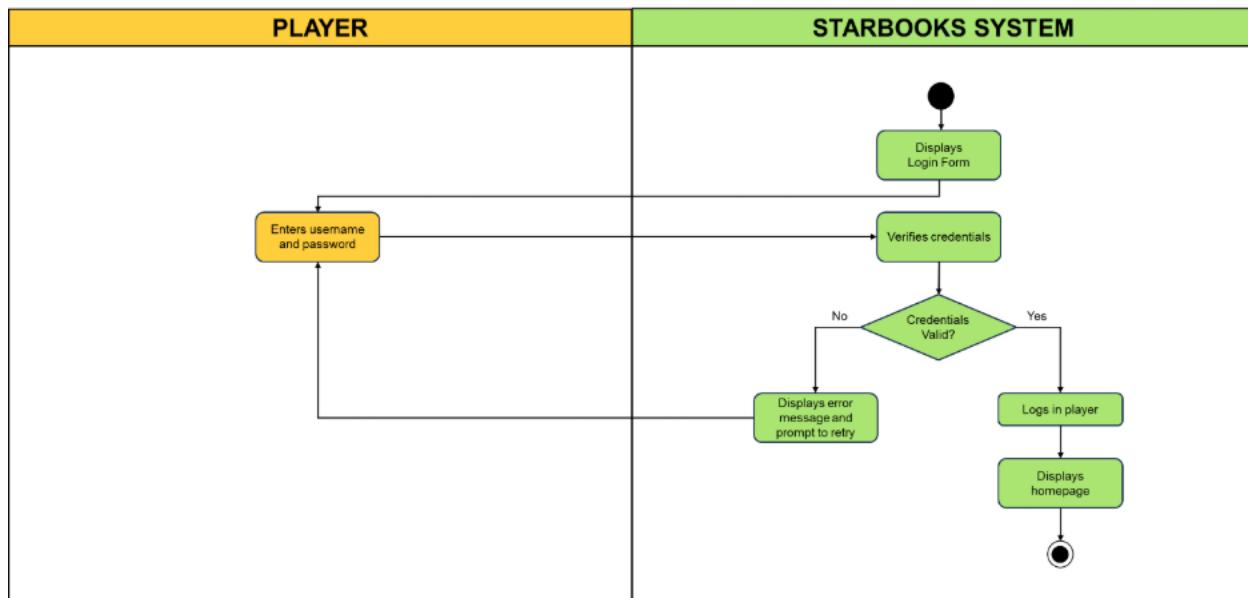
### 5.3 Activity Diagrams with Swim-lanes

#### MANAGE PLAYER ACCOUNT

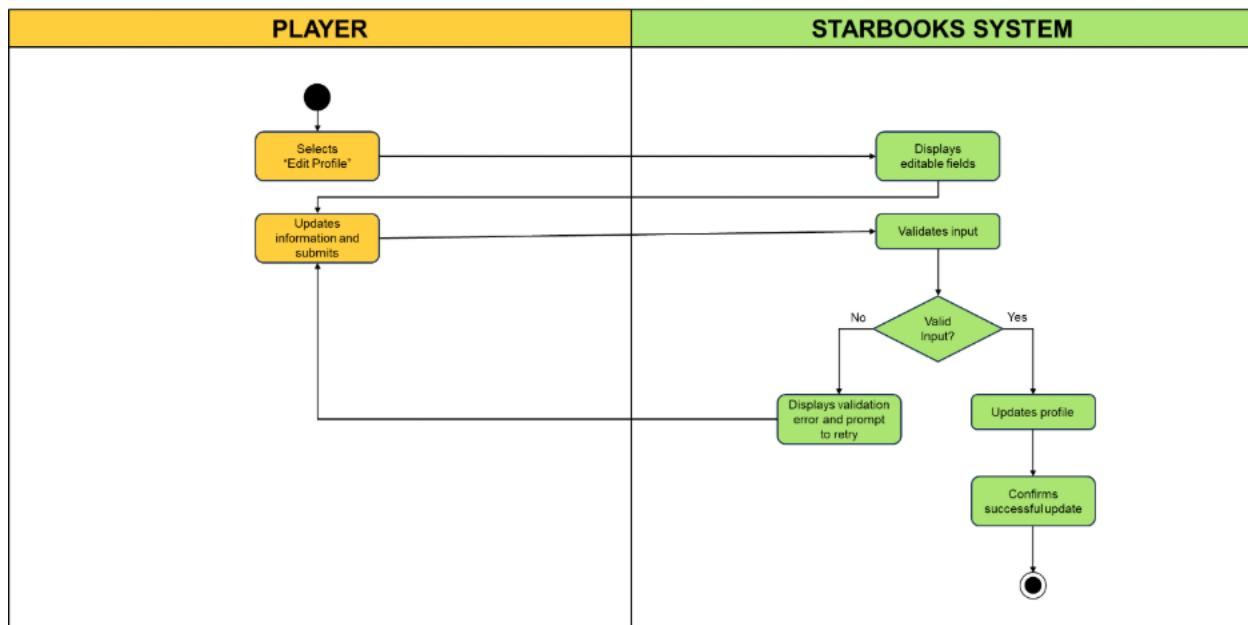
##### UC-MPA1: Player Register



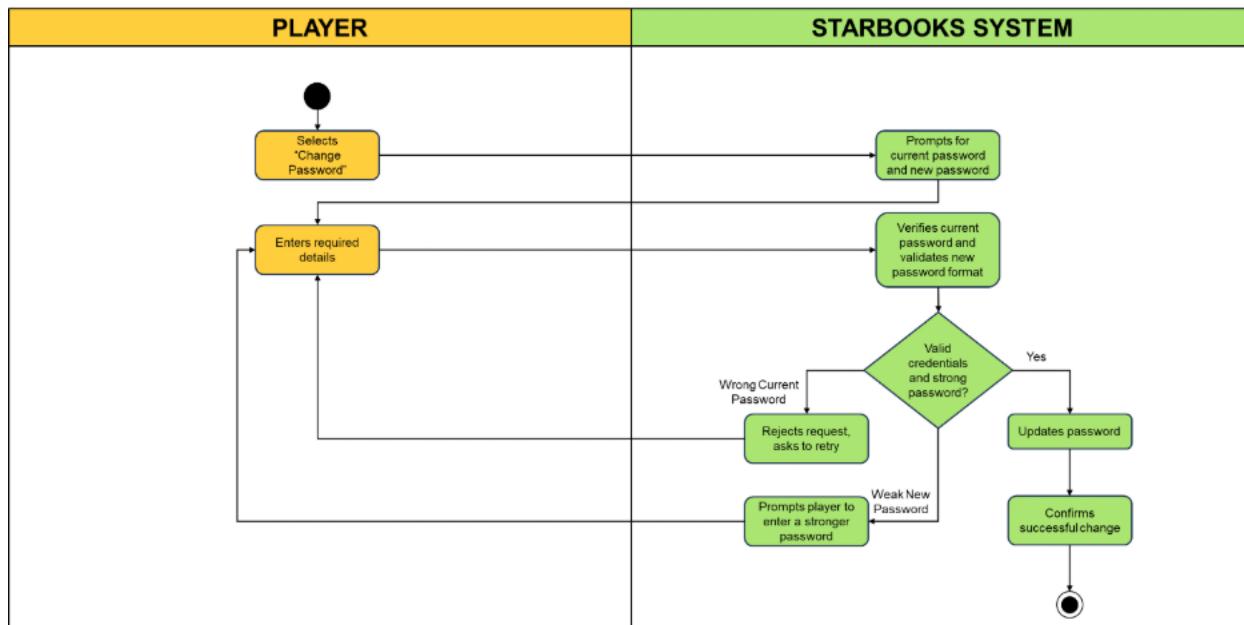
## UC-MP2: Player Login



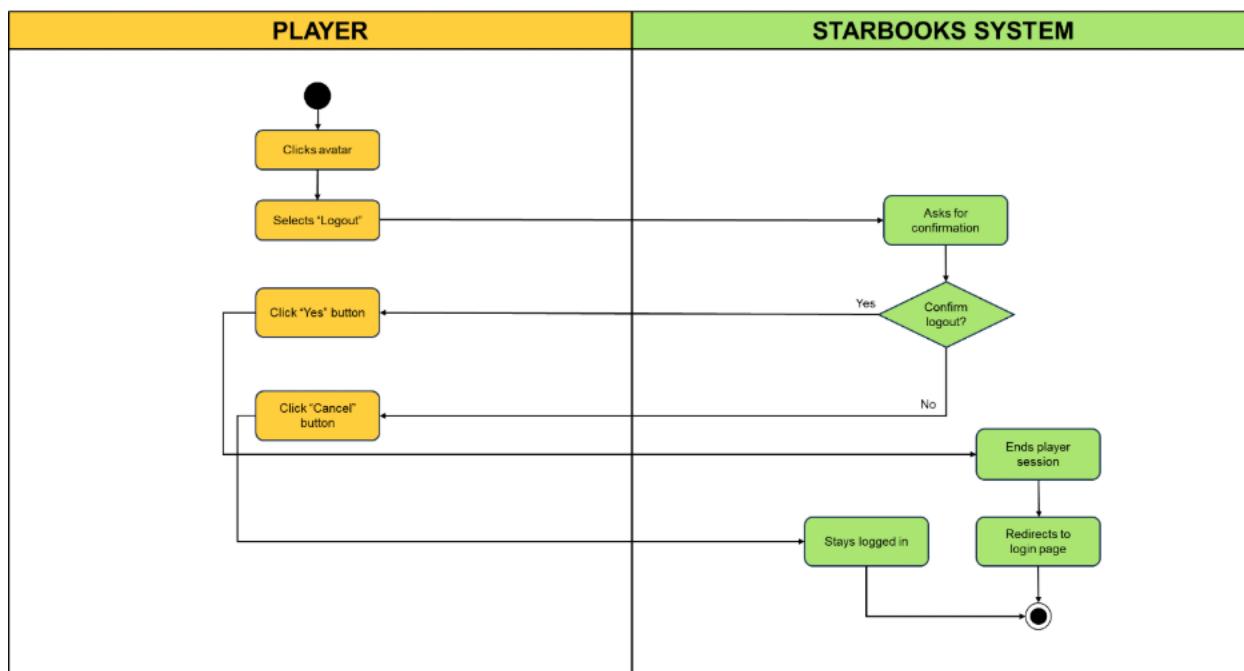
## UC-MP3: Edit Player Profile



### UC-MP4: Change Player Password

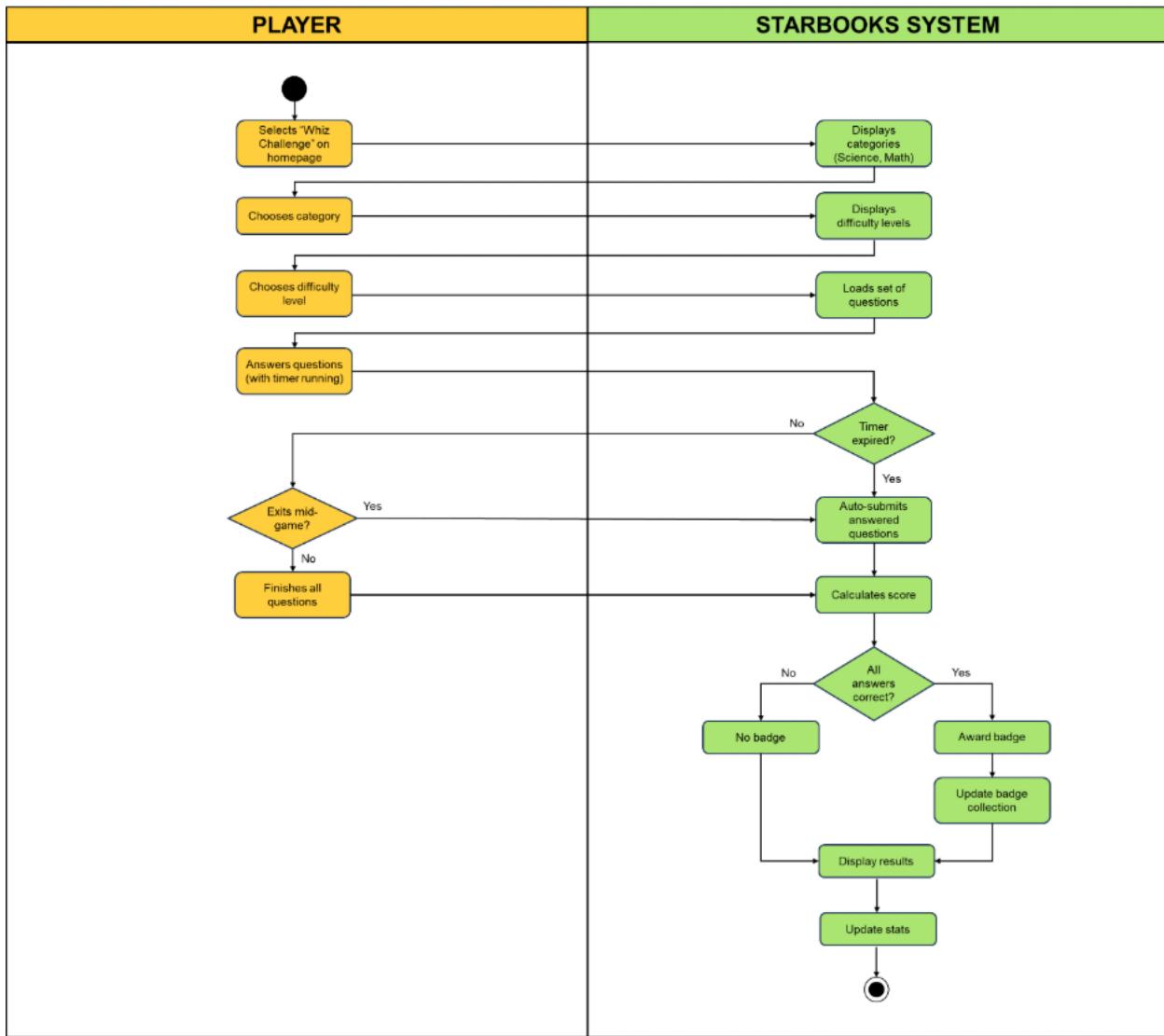


### UC-MP5: Player Logout

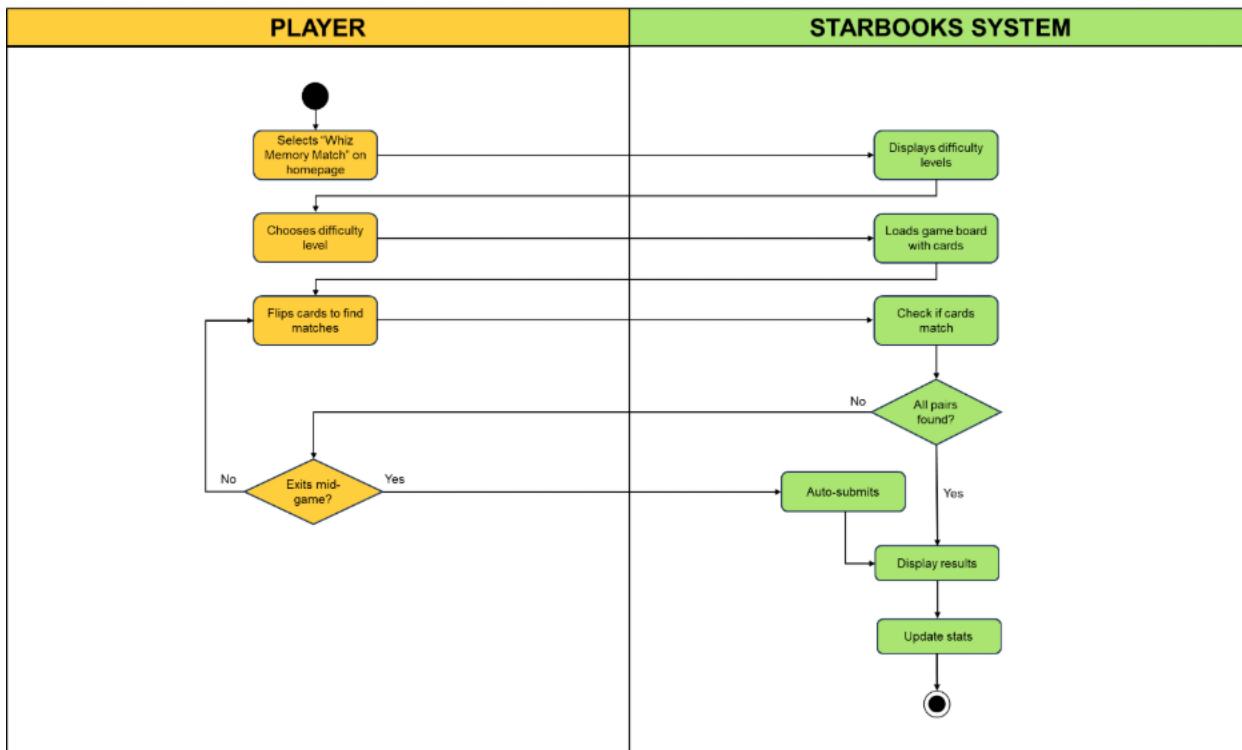


## PLAY GAME

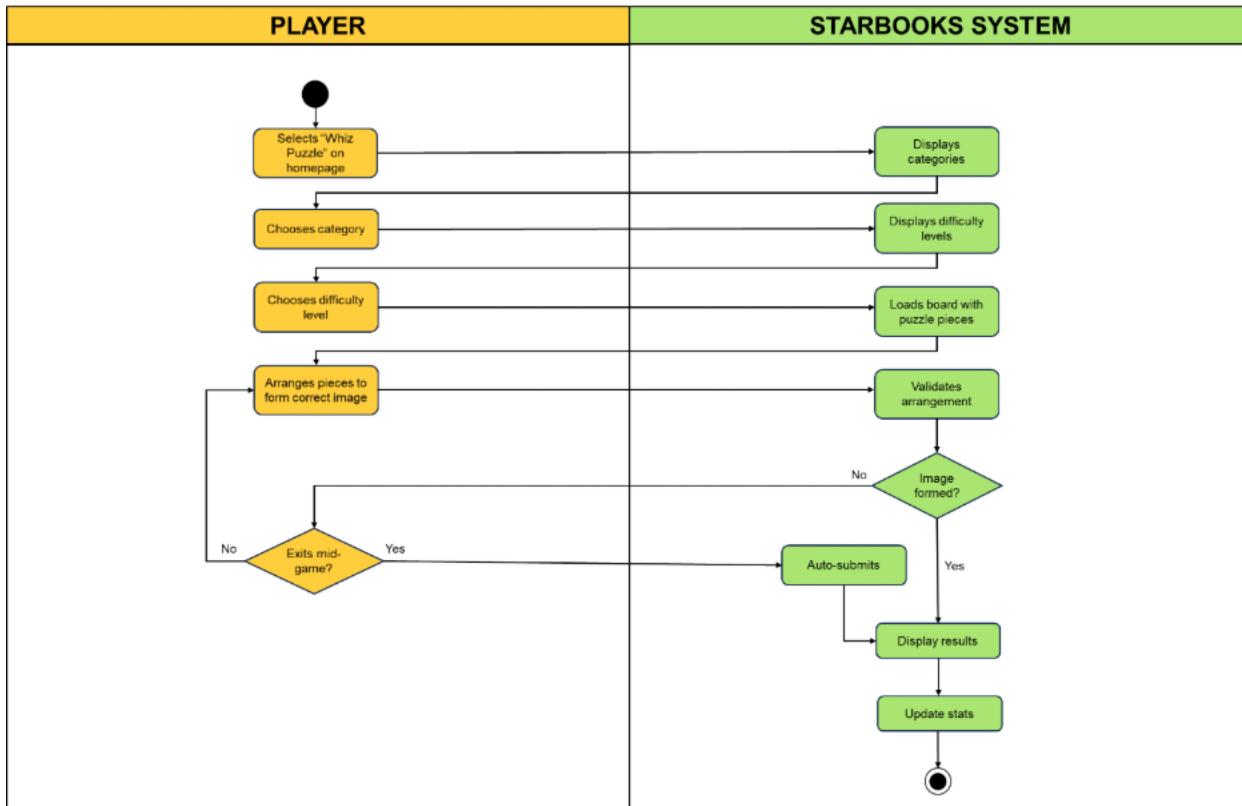
### UC-PG1: Play Whiz Challenge



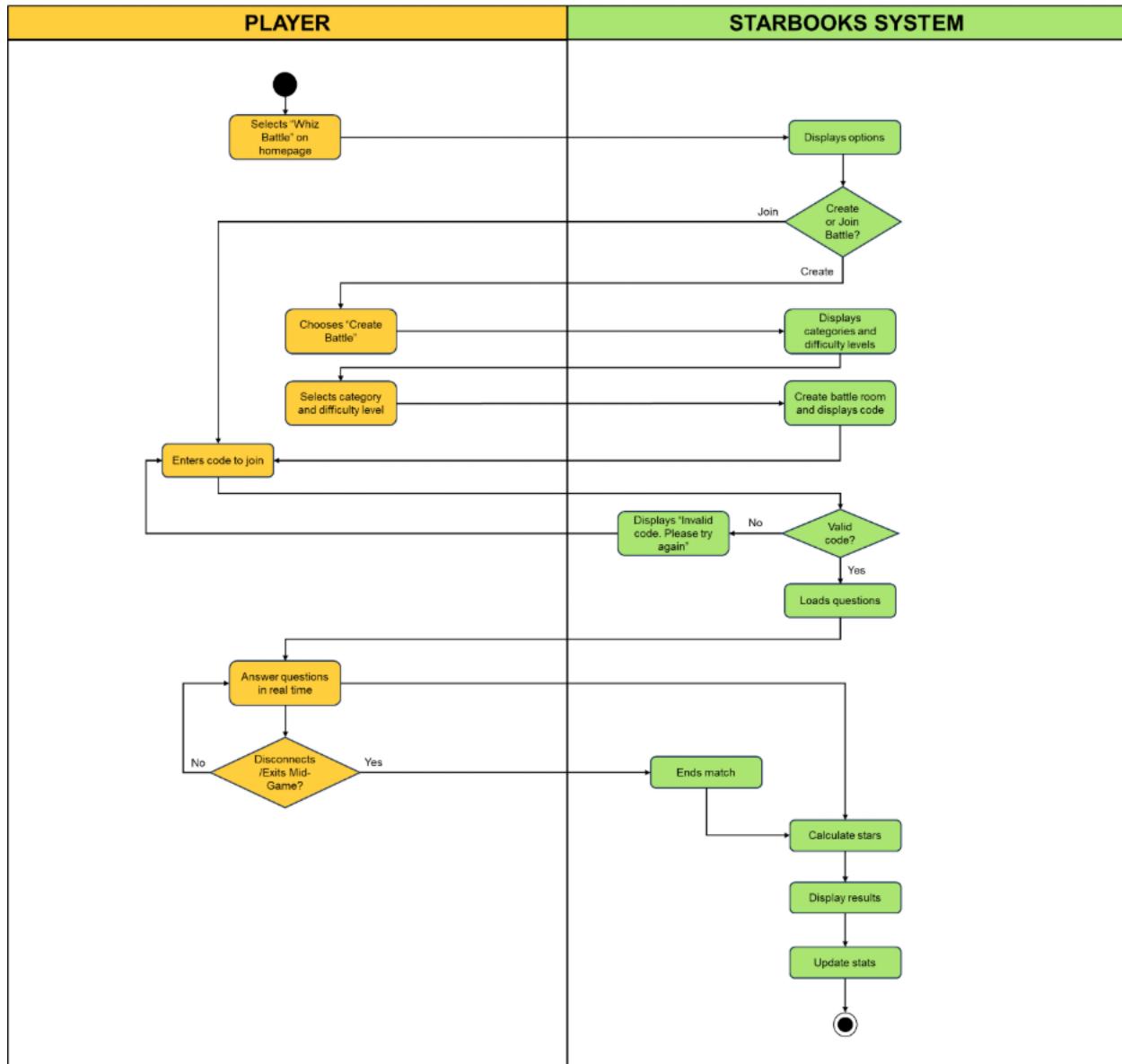
## UC-PG2: Play Whiz Memory Match



## UC-PG3: Play Whiz Puzzle

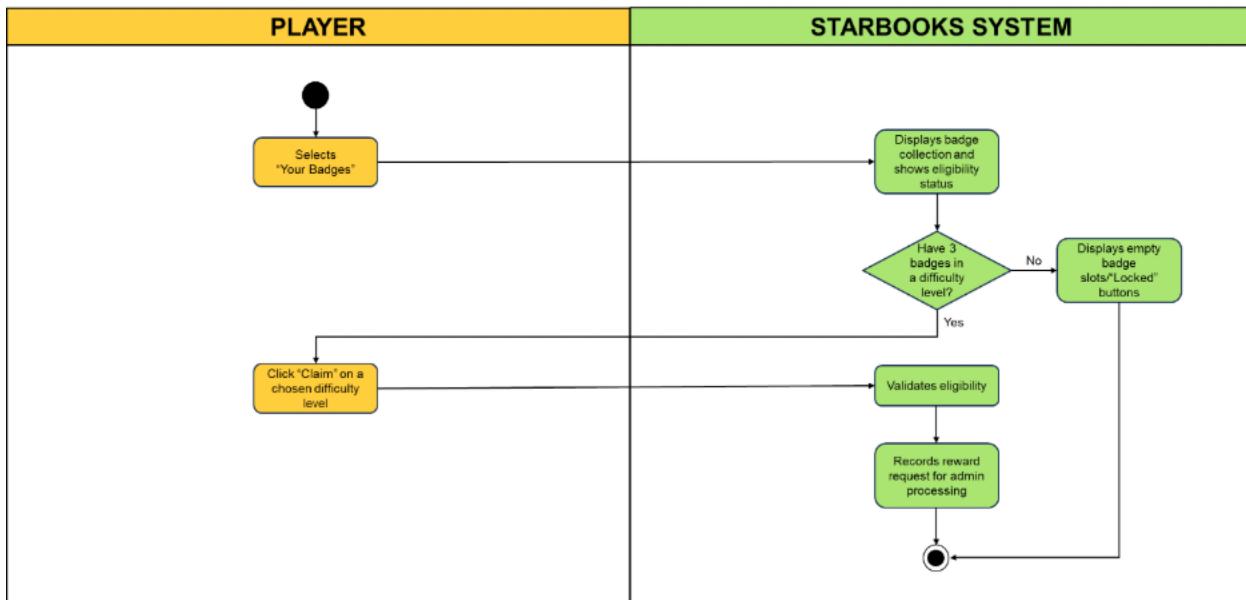


## UC-PG4: Play Whiz Battle

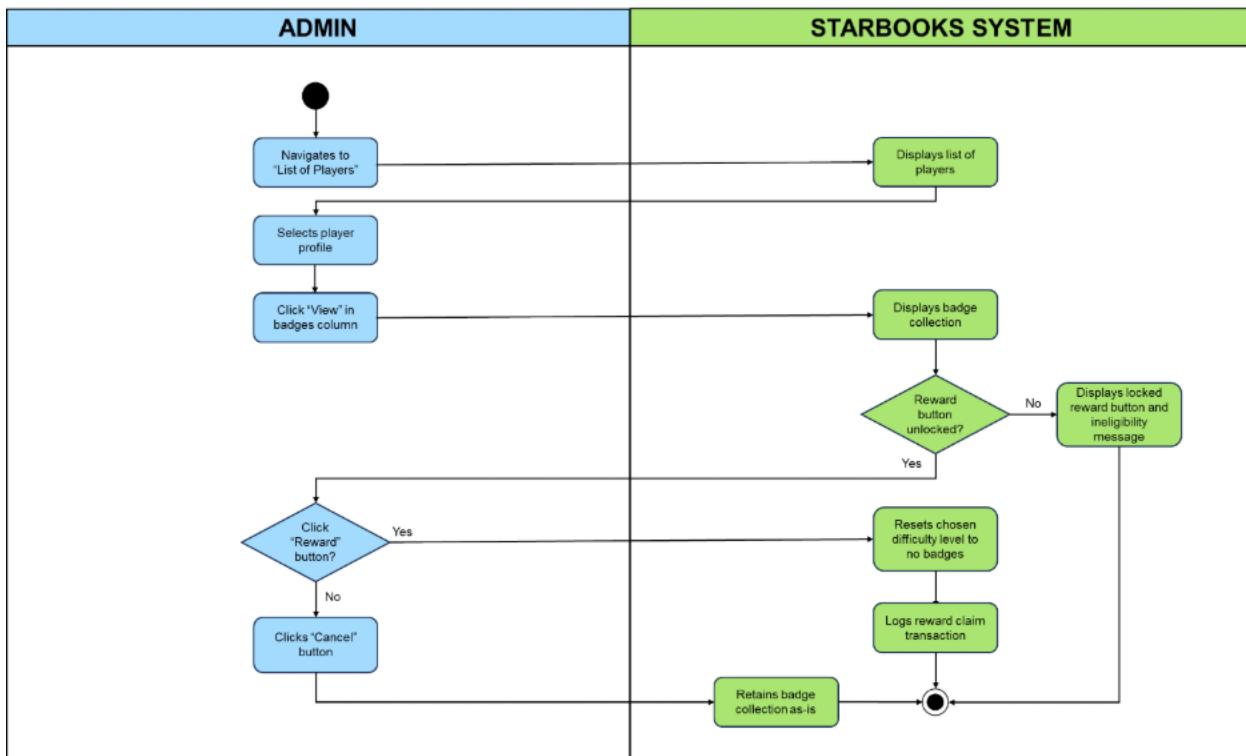


## REWARD REDEMPTION

### UC-RR1: Player Requests Reward

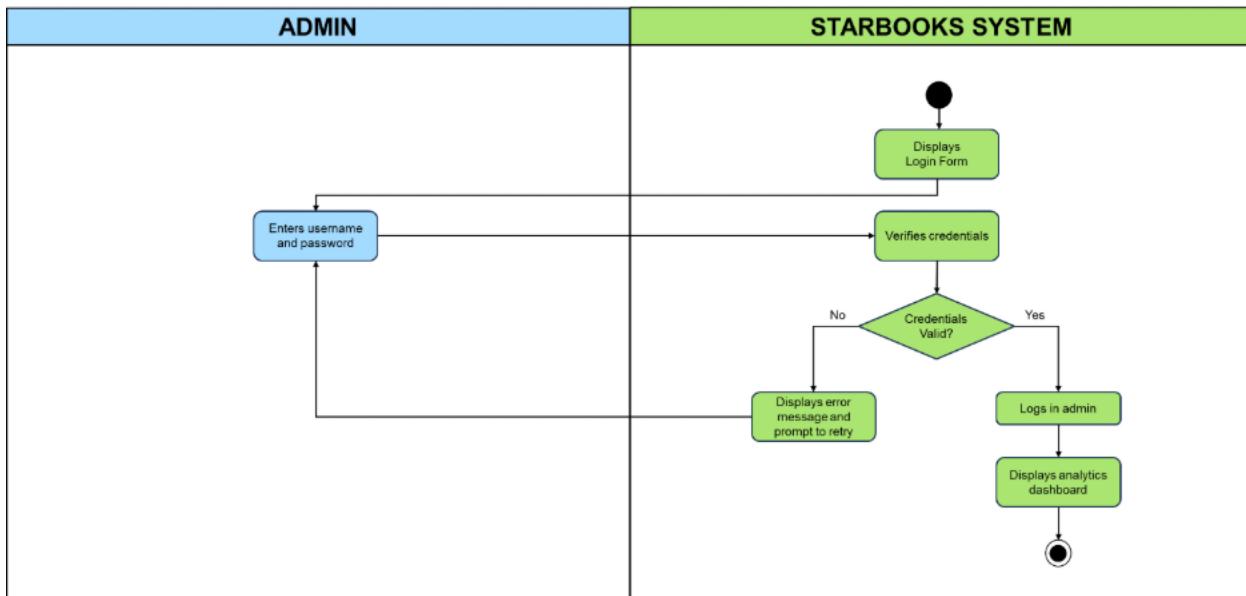


### UC-RR2: Admin Grants Reward

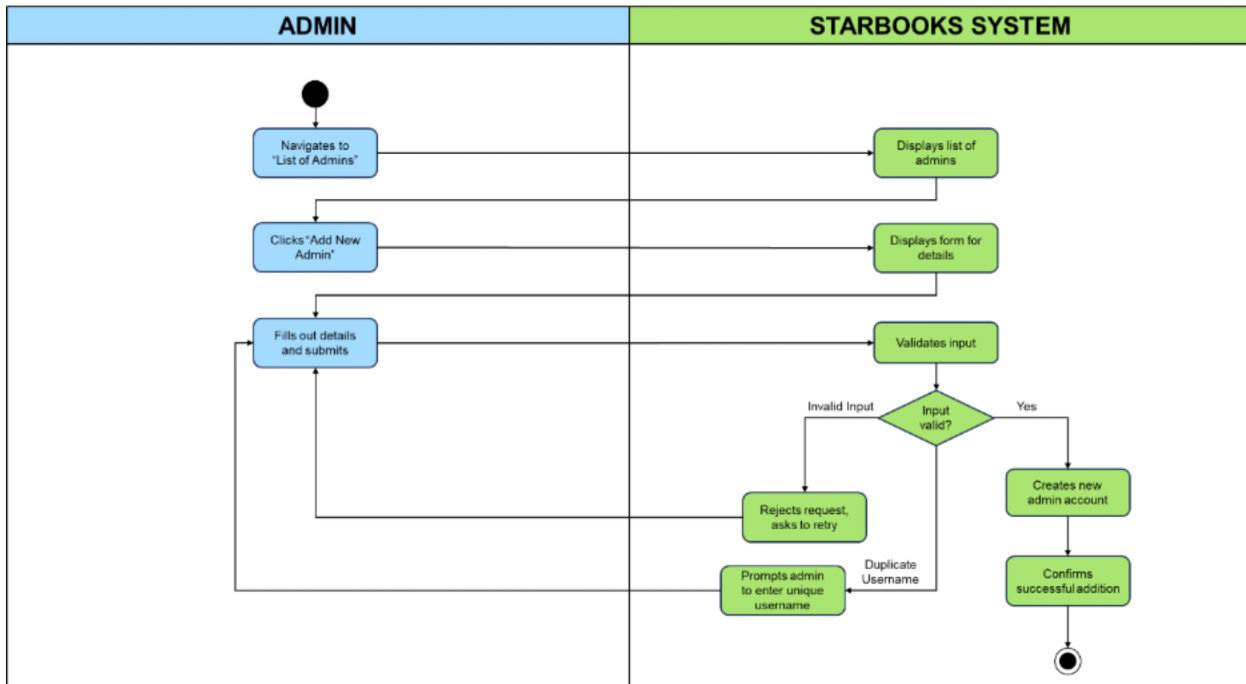


## MANAGE ADMIN ACCOUNT

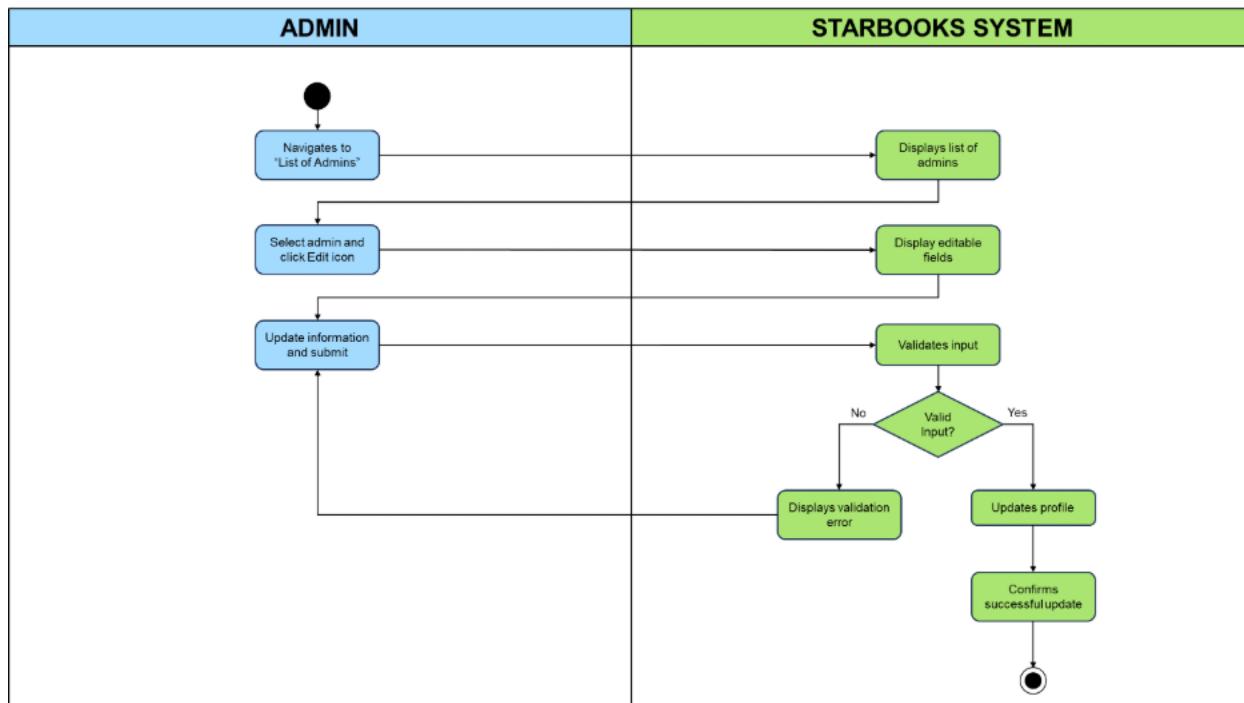
### UC-MAA1: Admin Login



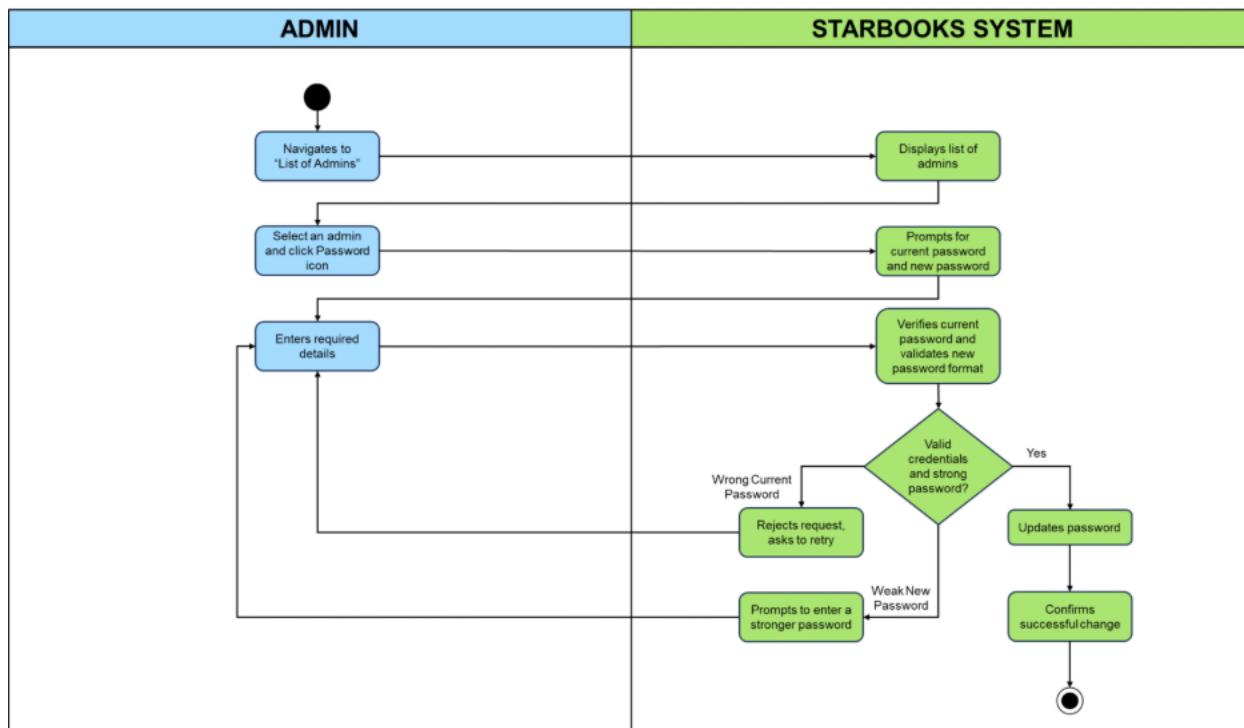
### UC-MAA2: Add New Admin



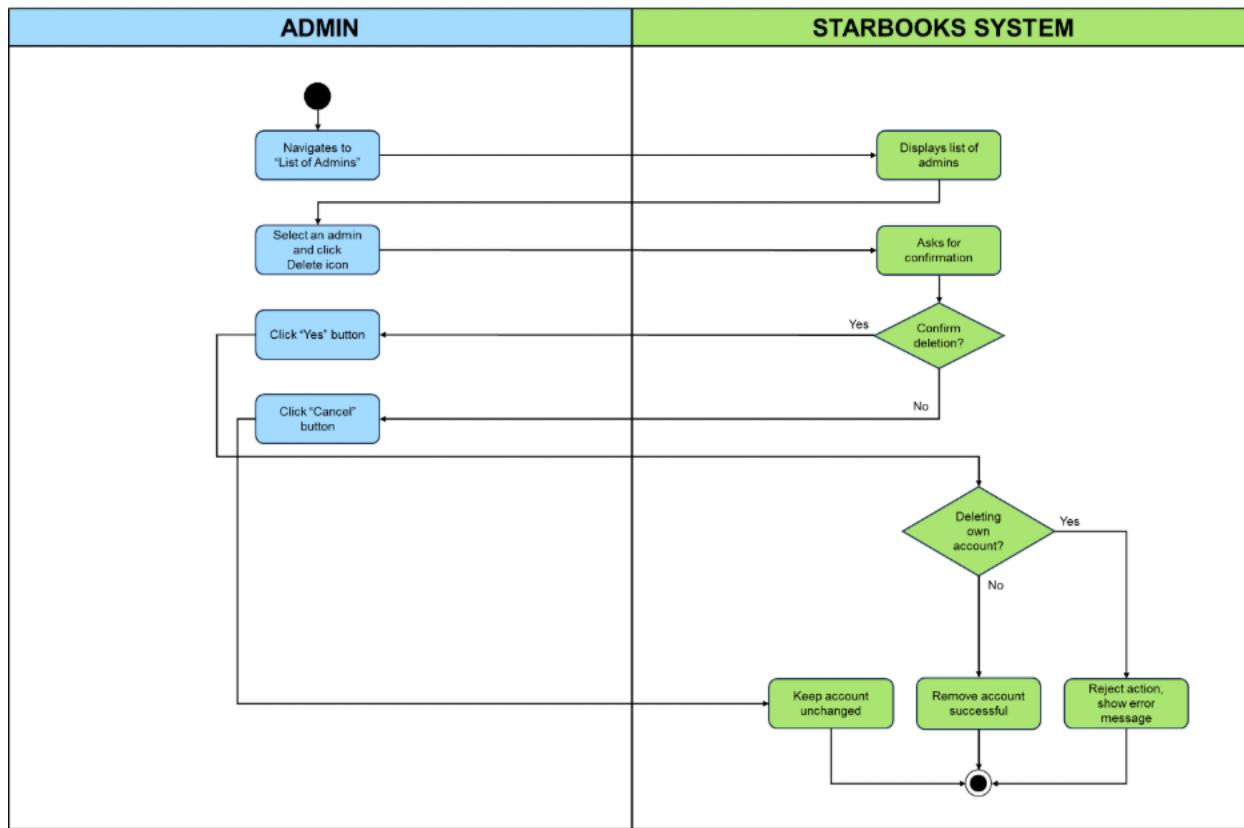
### UC-MAA3: Edit Admin Profile



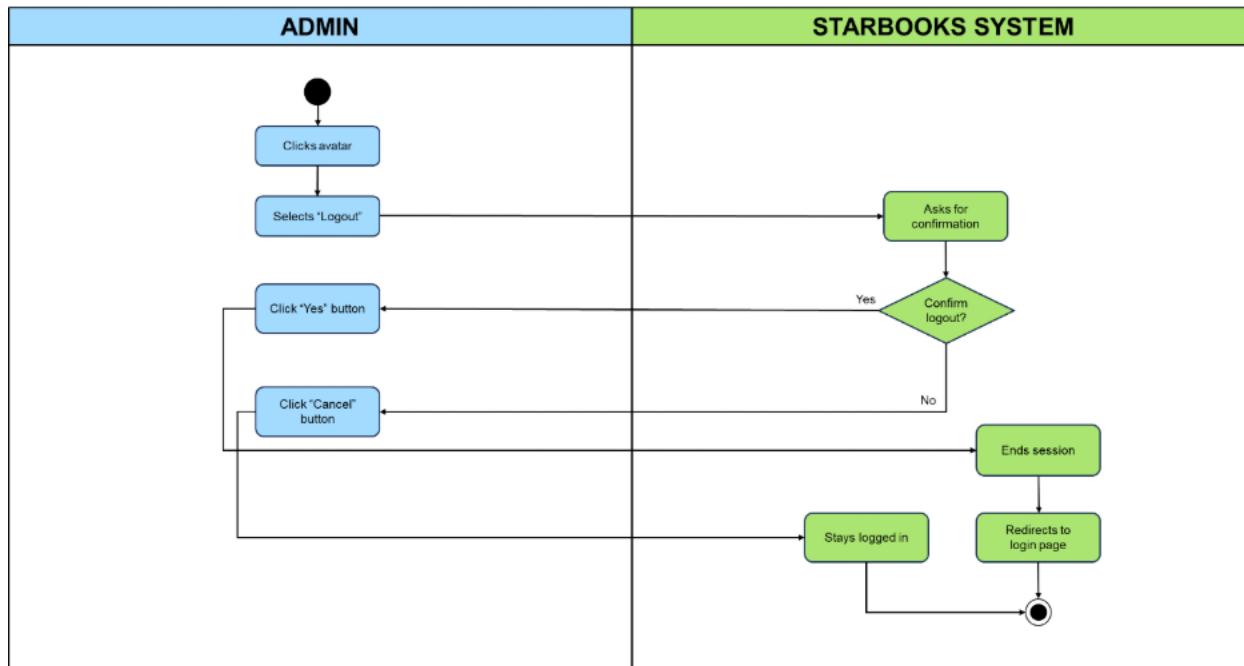
### UC-MAA4: Change Admin Password



## UC-MAA5: Delete Admin

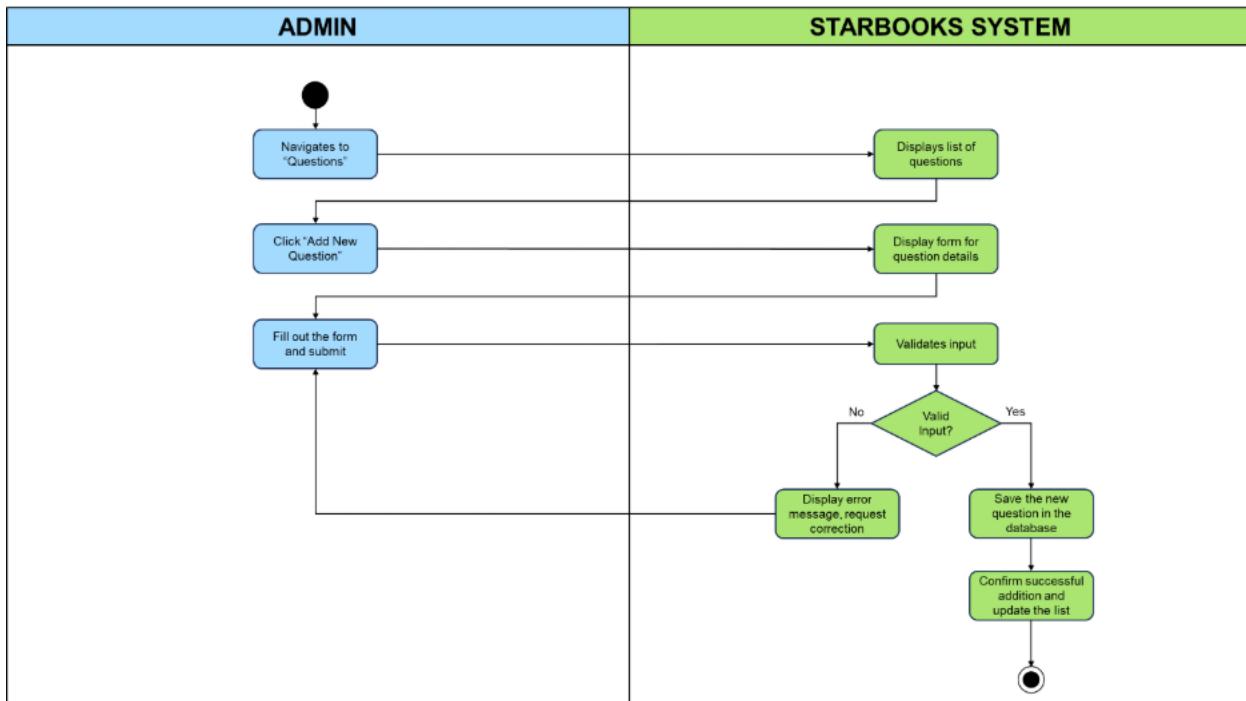


## UC-MAA6: Admin Logout

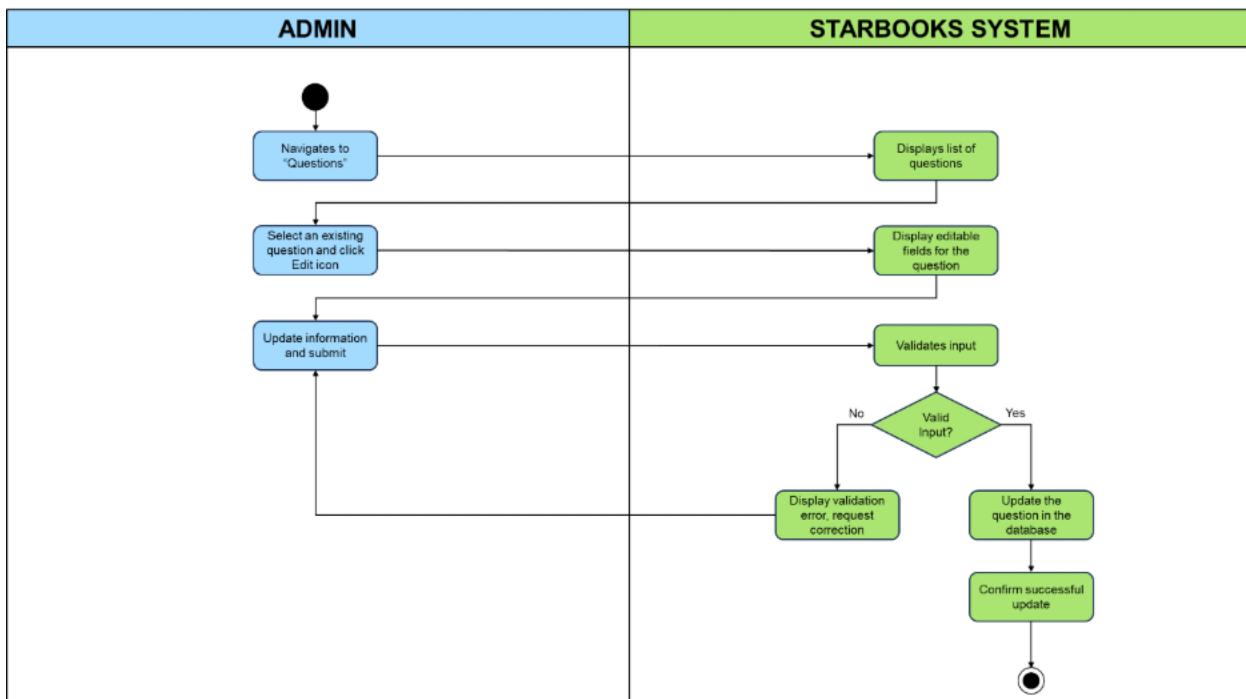


## MANAGE QUIZ SETTINGS

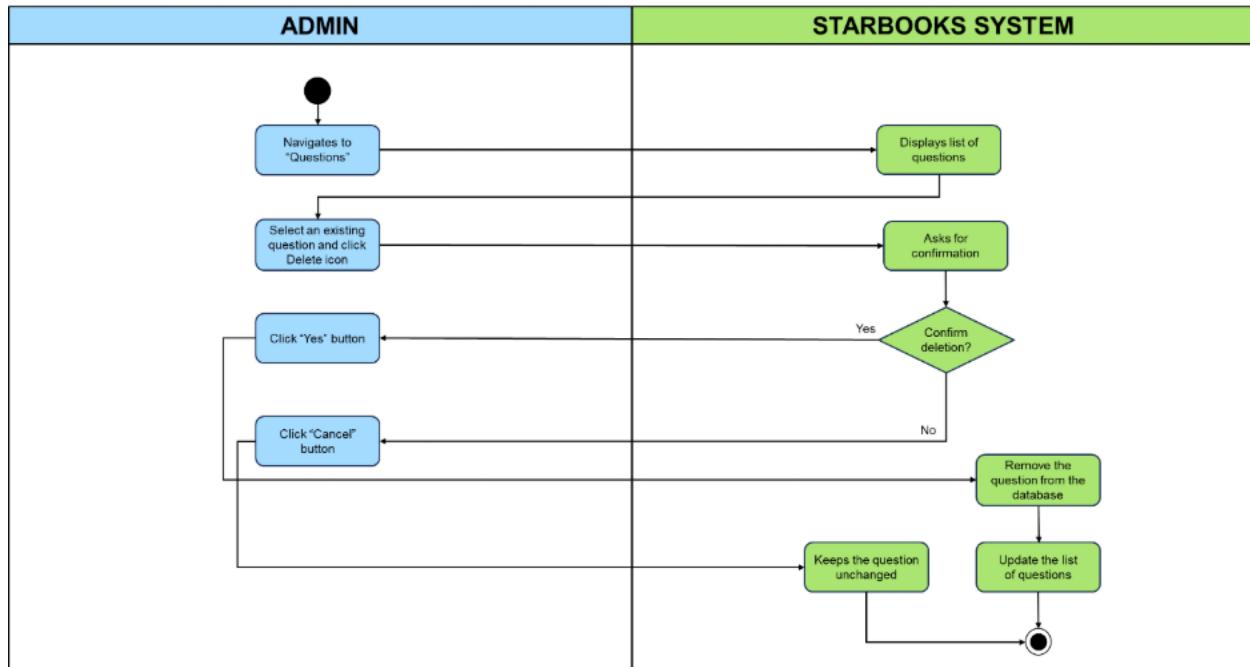
### UC-MQS1: Add New Question



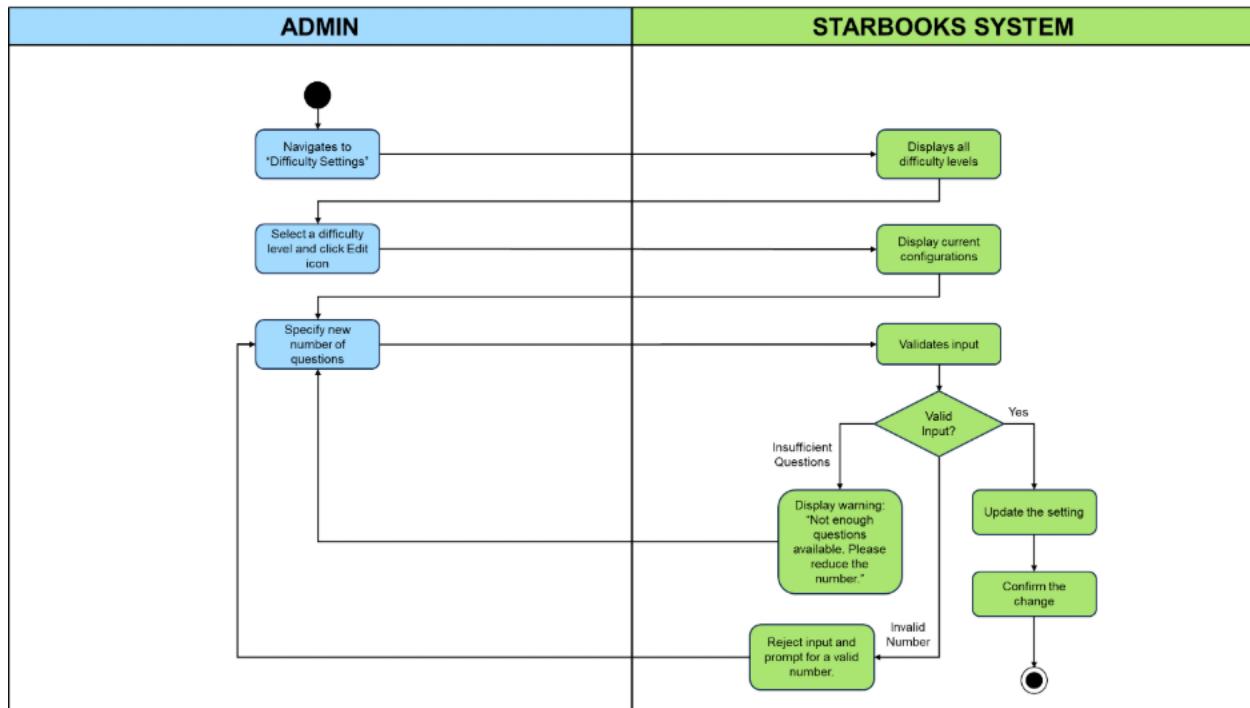
### UC-MQS2: Edit Question Info



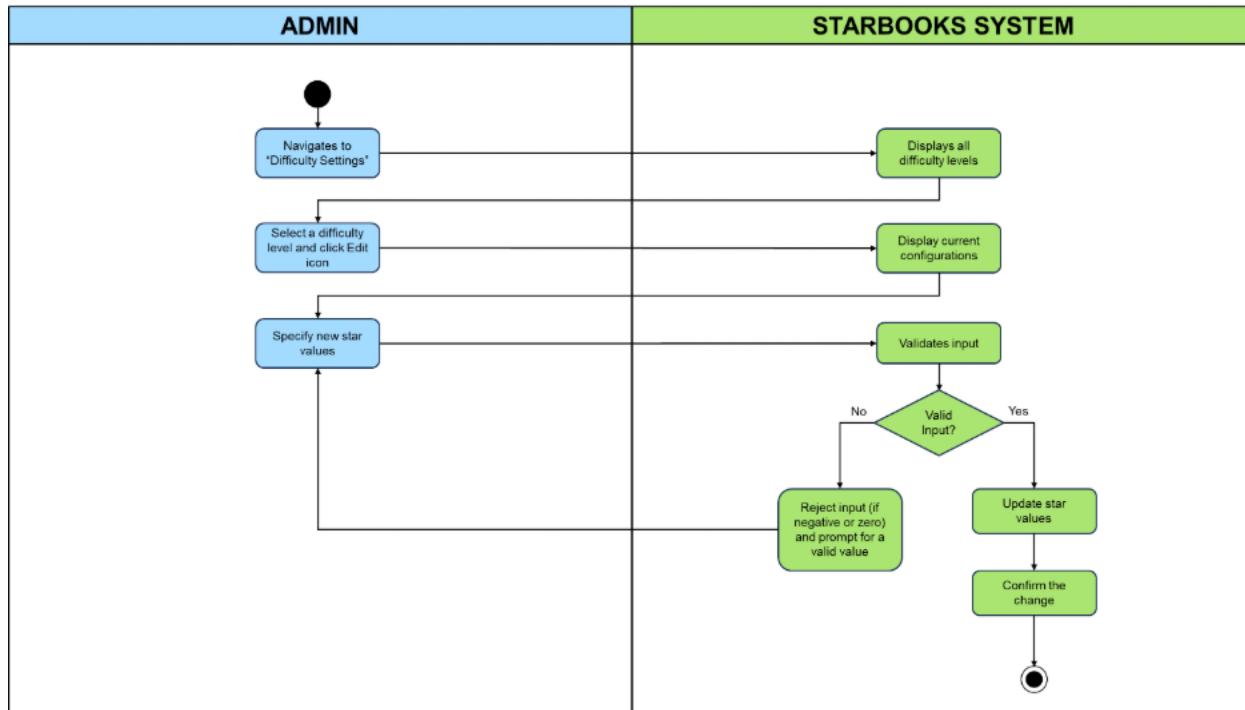
### UC-MQS3: Delete Question



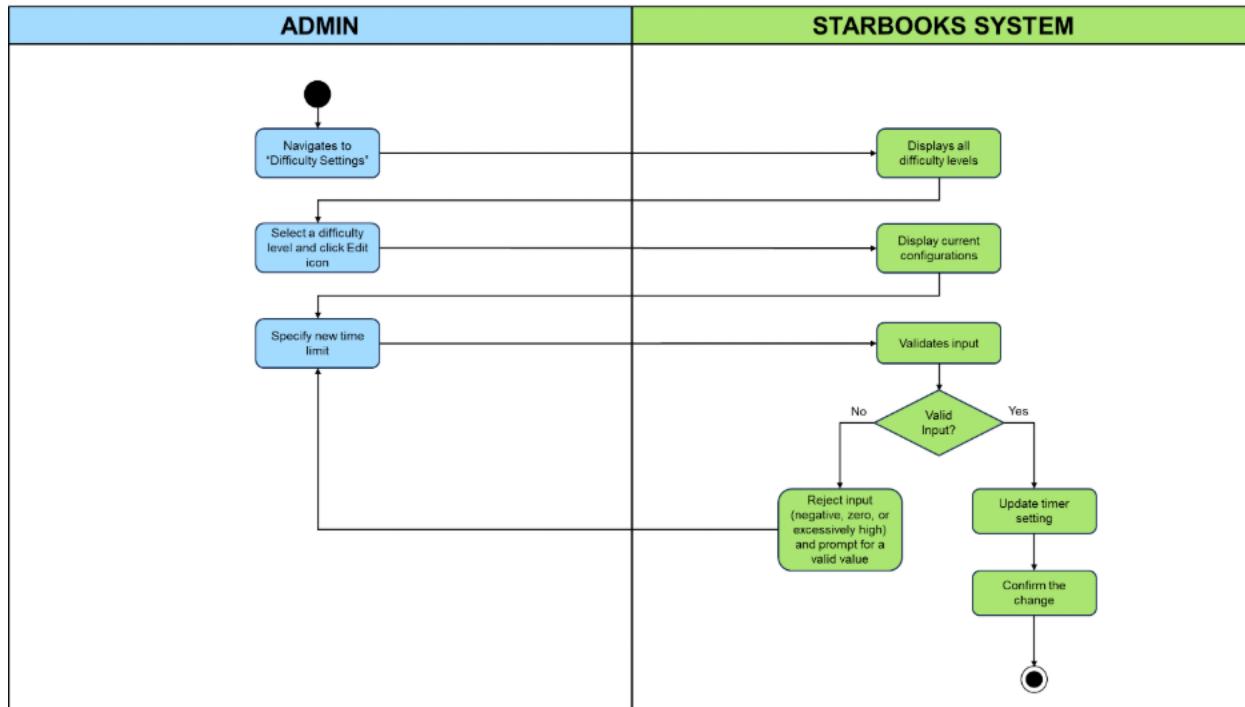
### UC-MQS4: Set Number of Questions



### UC-MQS5: Set Stars

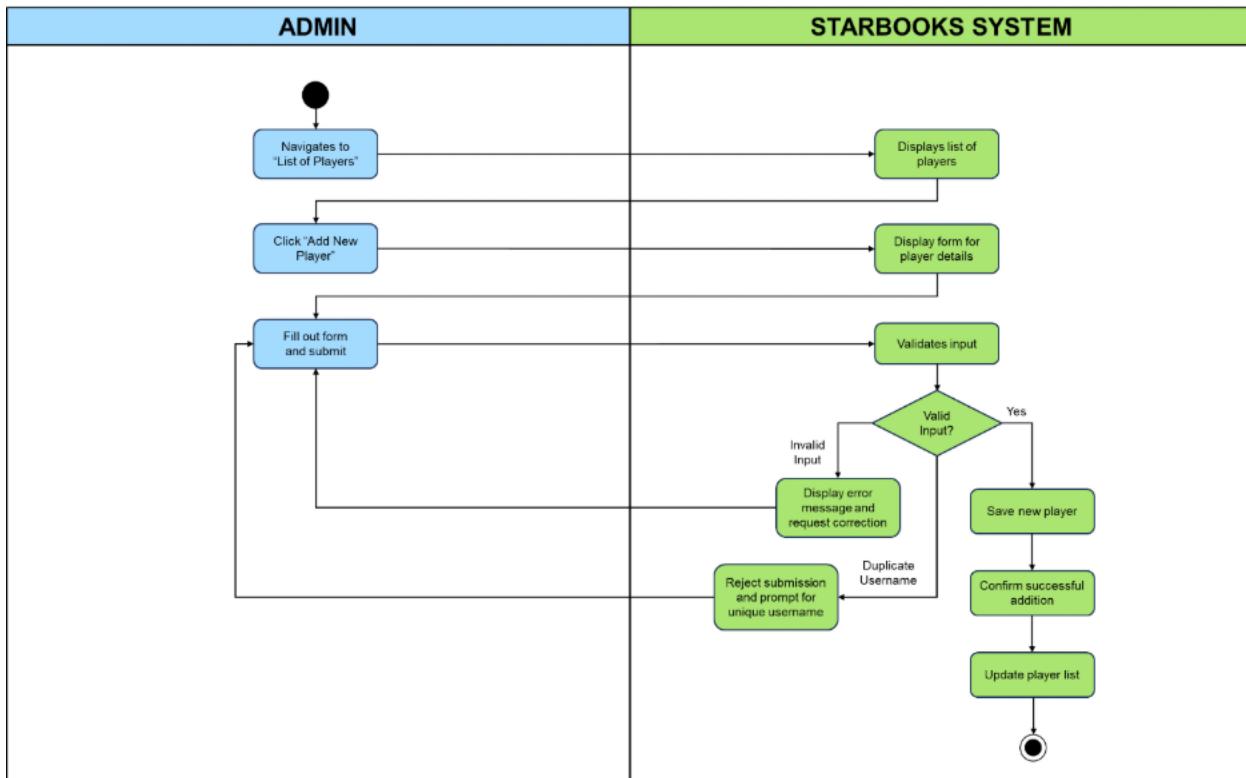


### UC-MQS6: Set Timer

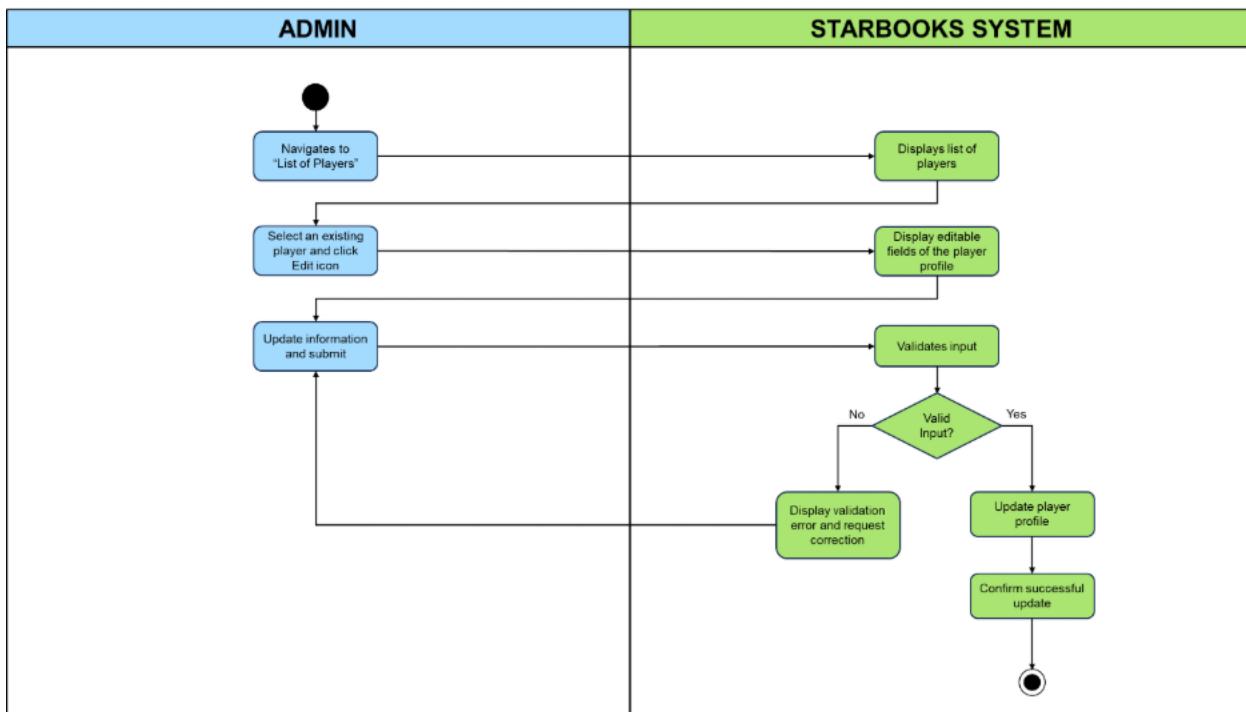


## MANAGE PLAYER LIST

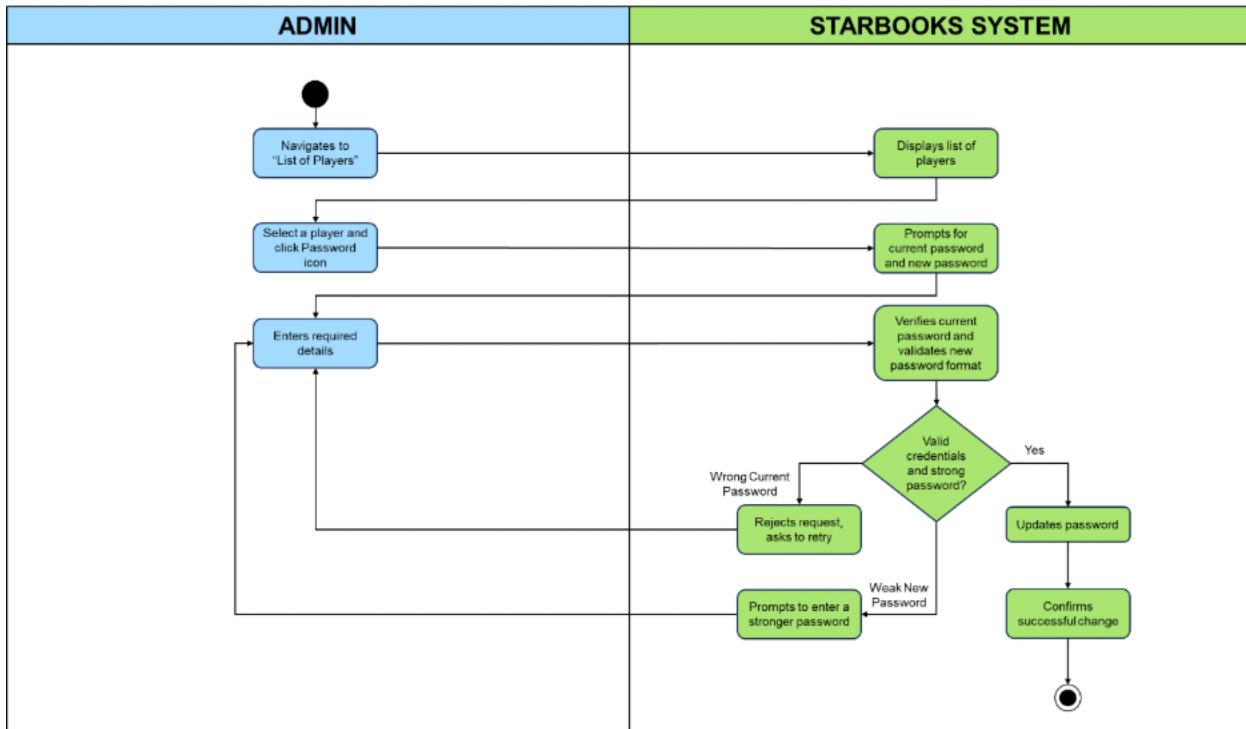
### UC-MPL1: Add New Player



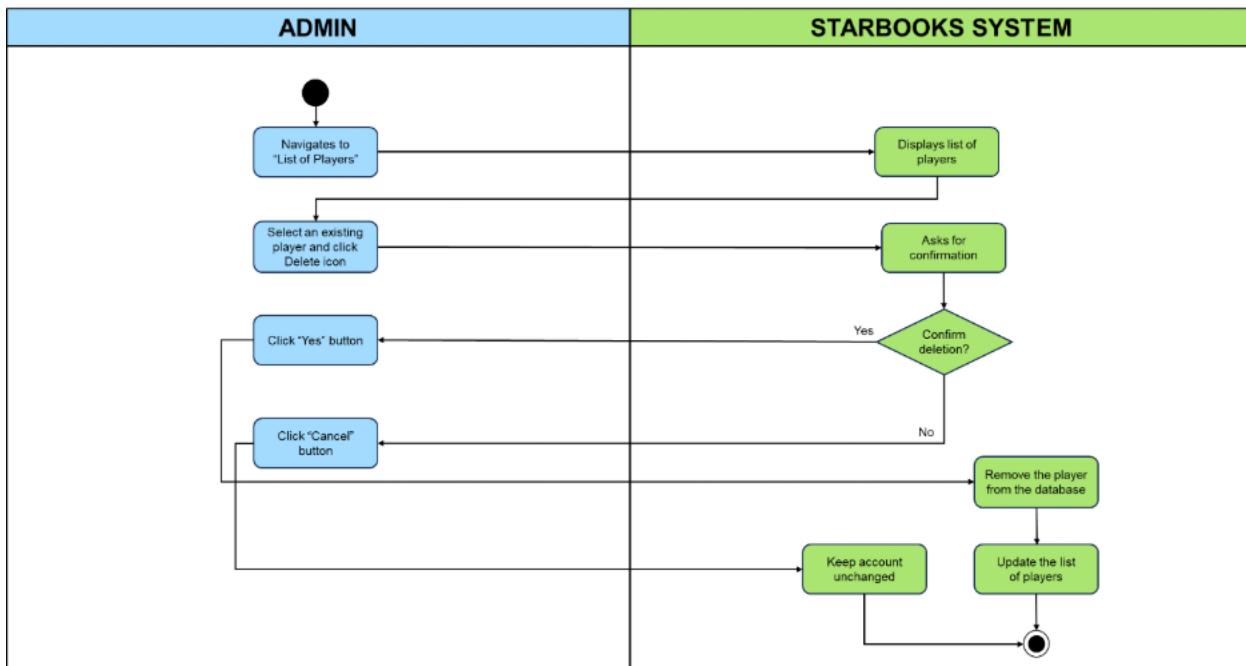
### UC-MPL2: Edit Player Profile



### UC-MPL3: Change Player Password

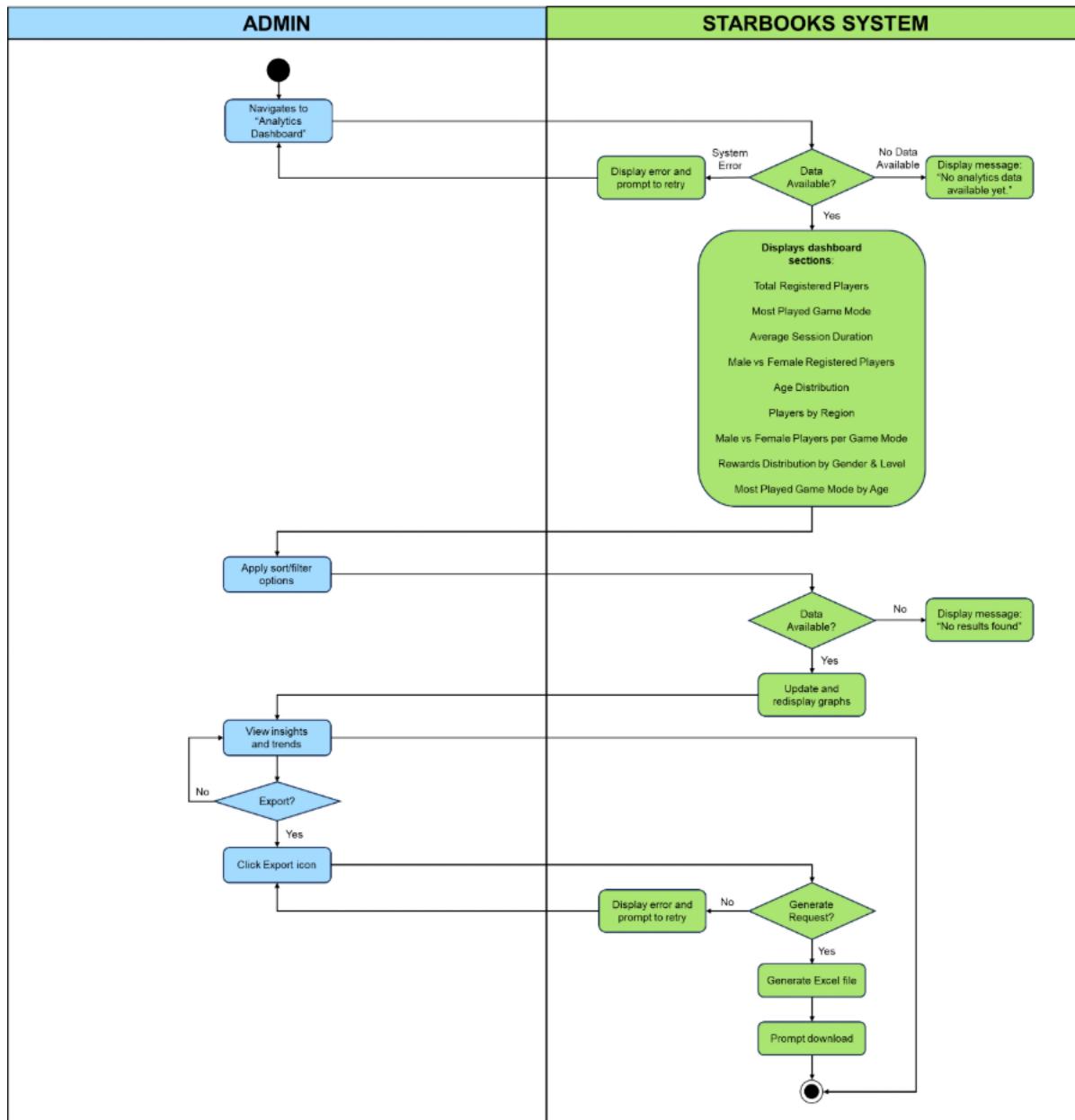


### UC-MPL4: Delete Player



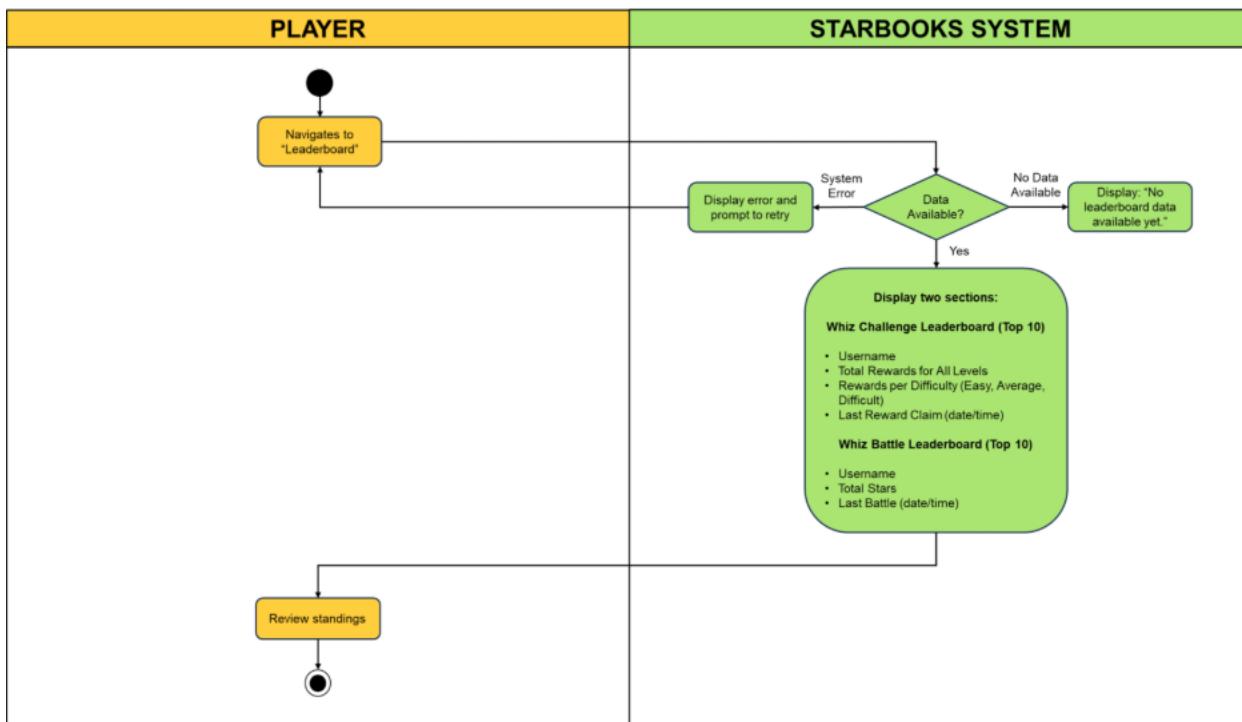
## VIEW ANALYTICS DASHBOARD

### UC-VAD1: View Analytics Dashboard

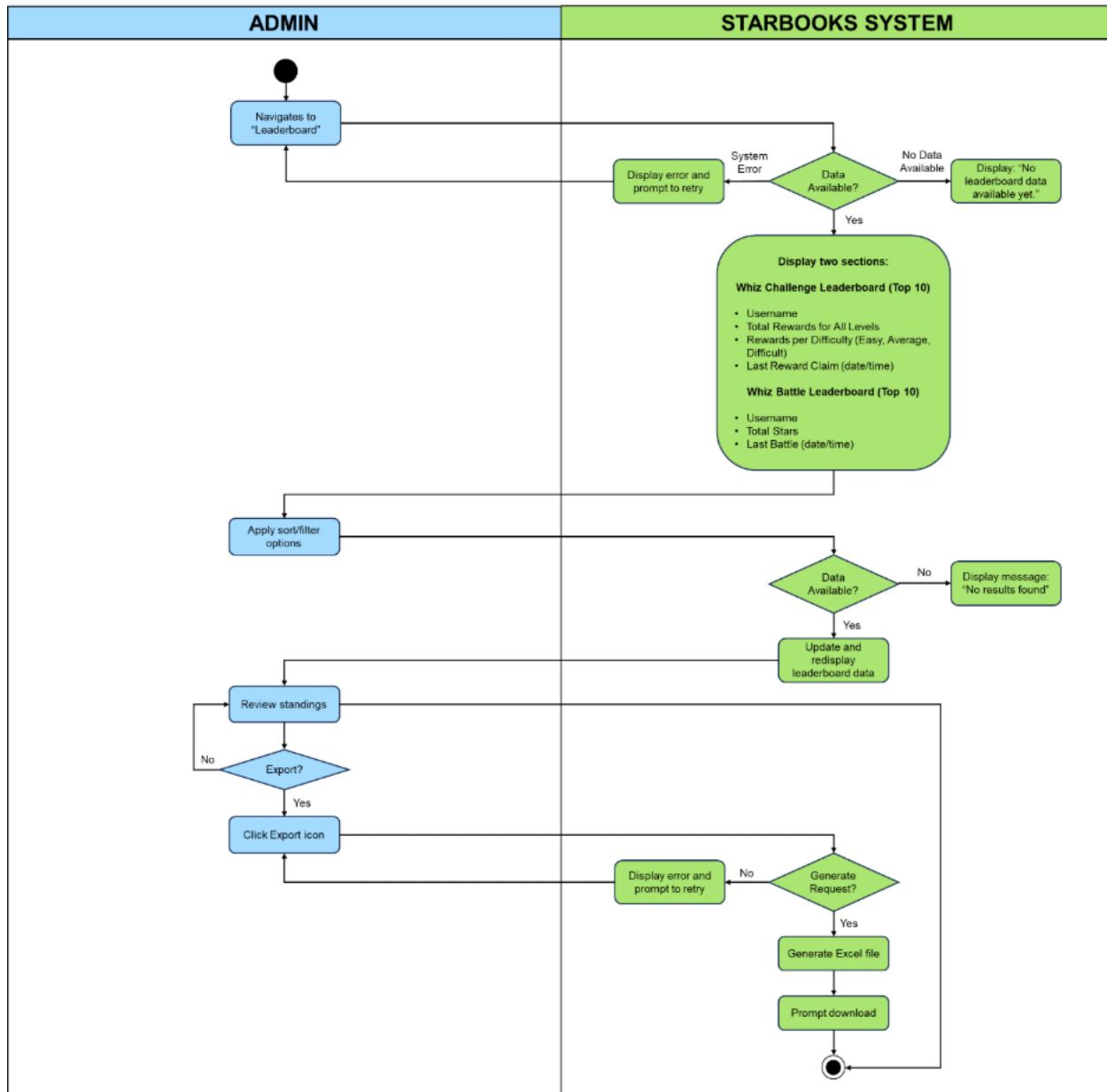


## VIEW LEADERBOARD

### UC-VLB1: Player View Leaderboard



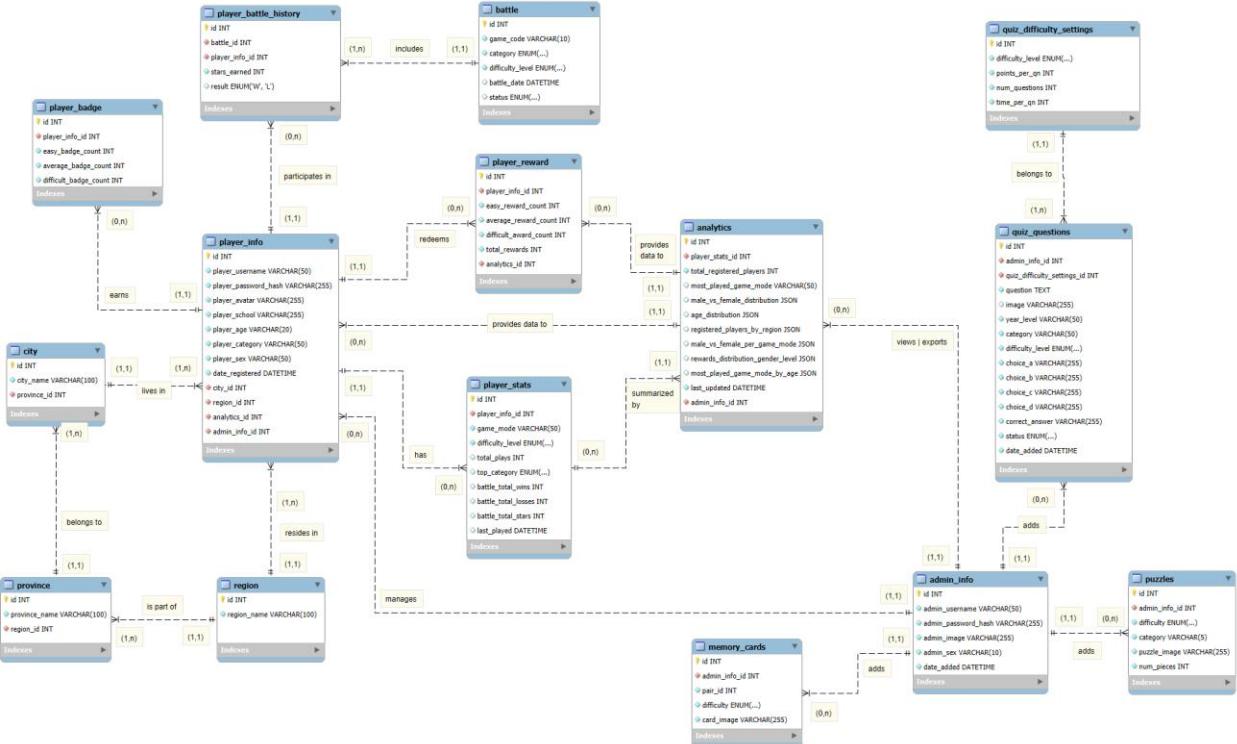
## UC-VLB2: Admin View Leaderboard



### 5.4 Test Cases

Excel File Link: [Test Cases.xlsx](#)

## VI. ENTITY RELATIONSHIP DIAGRAM



## VII. TECHNOLOGY STACK

### 7.1 Frontend Framework

- **Flutter**
  - Serves as the app's frontend with responsive, cross-platform UI. Uses Dart as its main language.
  - Manages user registration, login, and gameplay with dynamic screens and animations for smooth web experiences.

### 7.2 Backend Framework

- **Laravel**
  - Manages data flow between the Flutter app and the MongoDB database.
  - Handles API requests such as fetching region, province, and city data, and securely stores user information.
- **MongoDB (Database)**
  - A NoSQL database used to store user data and gameplay records using JSON structures.
  - Its scalability and schema flexibility are ideal for managing dynamic data like user profiles and game progress across multiple game modes.
- **Laragon (Hosting Service)**
  - Acts as the local development server to host the Laravel backend during development.
  - A lightweight setup that simulates a web server, letting the Flutter frontend connect to Laravel APIs or local network for testing.

### 7.3 Frontend/ Backend Communication

- **Dio**
  - Flutter's HTTP client for connecting to the Laravel backend.
  - Supports error handling, interceptors, and response formatting for smoother, more reliable API communication.

GitHub APC-SoCIT Project Repository URL [Link](#)

OpenProject Team Site URL [Link](#)

Cloud Hosted Site URL [Link](#)