Use Case Diagram and Fully Dressed Use Cases 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:

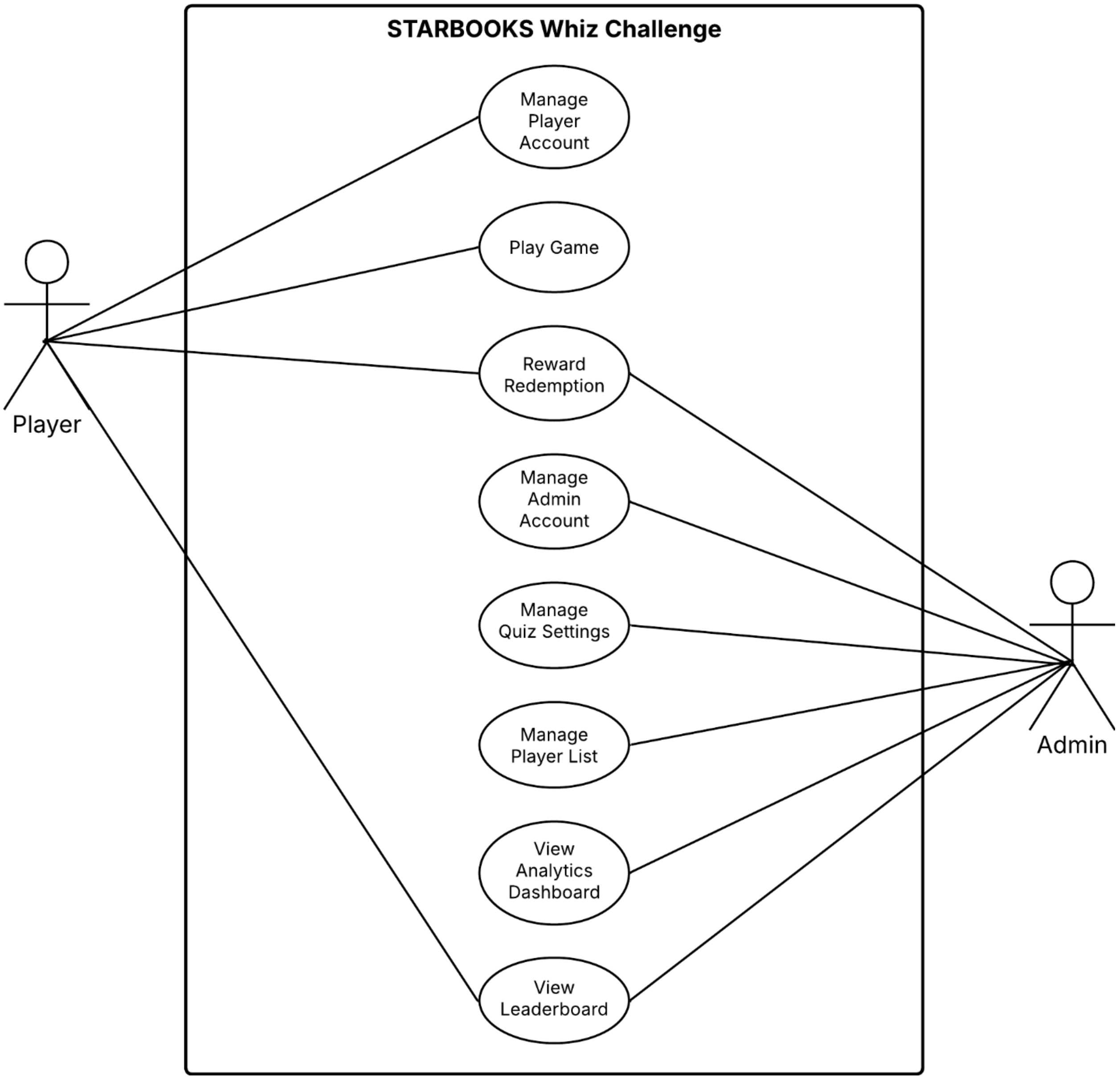
Dumbrique, Kelly (Team Leader)
Gercan, Arcielle Marie (Member)
Quianzon, Shandrae Lois (Member)
Salipande, Janice Maxene (Member)
G06 NEXUS – SF 231

Submitted to:

Mr. Jose Eugenio L. Quesada
Course Instructor

October 2025
1st Term
A.Y. 2025 – 2026

USE CASE DIAGRAM



FULLY DRESSED USE CASES

Manage Player Account

Use Case ID	UC-MPA0
Use Case Name	Manage Player Account
Created By	Shandrae Lois Quianzon
Date Created	September 27, 2025
Description	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.
Priority	High
Primary Actor	Player
Secondary Actor	System
Preconditions	1. Player must have access to the system.
Postconditions	1. Player account is created, updated, or accessed depending on the sub-function executed.
Includes	1. UC-MPA1 Player Register 2. UC-MPA2 Player Login 3. UC-MPA5 Player Logout
Extends	1. UC-MPA3 Edit Player Profile

Use Case ID	UC-MPA1
Use Case Name	Player Register
Created By	Shandrae Lois Quianzon
Date Created	September 27, 2025
Description	This use case describes how a player registers a new account in the system.
Priority	High
Primary Actor	Player
Secondary Actor	System
Preconditions	1. Player must have access to the system. 2. Player is not yet registered.
Postconditions	1. A new player account is created. 2. Player is redirected to the login page after successful registration.
Main Flow	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.

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

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

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

Main Flow	1. Player selects “Edit Profile.” 2. System displays editable fields. 3. Player updates information and submits. 4. System validates input and updates the profile. 5. System confirms successful update.
Alternative Flows	**3a. Invalid Profile Update** 1. Player enters invalid or empty fields. 2. System displays validation error and requests correction.

Use Case ID	UC-MPA4
Use Case Name	Change Player Password
Created By	Shandrae Lois Quianzon
Date Created	September 27, 2025
Description	This use case describes how a player changes their account password.
Priority	Medium
Primary Actor	Player
Secondary Actor	System
Preconditions	1. Player must be logged in.
Postconditions	1. Player’s password is updated successfully.
Main Flow	1. Player selects “Change Password” from the Edit Profile page. 2. System prompts for current password and new password. 3. Player enters the required details. 4. System verifies current password and validates new password format. 5. System updates the password. 6. System confirms successful password change.
Alternative Flows	**3a. Wrong Current Password** 1. Player enters incorrect current password. 2. System rejects request and asks to retry. **3b. Weak New Password** 1. Player enters a password that does not meet security requirements. 2. System prompts the player to enter a stronger password.

Use Case ID	UC-MPA5
Use Case Name	Player Logout
Created By	Shandrae Lois Quianzon

Date Created	September 27, 2025
Description	This use case describes how a player logs out of the system.
Priority	Low
Primary Actor	Player
Secondary Actor	System
Preconditions	1. Player must be logged in.
Postconditions	1. Player is logged out of the system. 2. System redirects the player to the login page.
Main Flow	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.
Alternative Flows	**2a. Cancel Logout** 1. Player cancels at the confirmation prompt. 2. System keeps the player logged in and returns them to the previous page.

Play Game

Use Case ID	UC-PG0
Use Case Name	Play Game
Created By	Arcielle Marie Gercan
Date Created	September 27, 2025
Description	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.
Priority	High
Primary Actor	Player
Secondary Actor	System
Preconditions	1. Player must be logged in. 2. Player must have a valid account.
Postconditions	1. Player stats are updated based on gameplay. 2. Badges may be awarded depending on performance and game rules.
Includes	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

Use Case ID	UC-PG1
Use Case Name	Play Whiz Challenge
Created By	Arcielle Marie Gercan
Date Created	September 27, 2025
Description	This use case describes how a player selects and plays the Whiz Challenge game mode, answering questions under a timer and potentially earning badges.
Priority	High
Primary Actor	Player
Secondary Actor	System
Preconditions	1. Player must be logged in. 2. Player must have a valid account.
Postconditions	1. Player stats are updated. 2. A badge may be awarded if conditions are met.
Main Flow	1. Player selects “Whiz Challenge” on the homepage. 2. System displays categories (Science or Math). 3. Player chooses a category. 4. System displays difficulty levels (Easy, Average, Difficult).

	5. Player chooses a difficulty level. 6. System loads a set of questions based on the chosen category and difficulty. 7. Player answers questions within the given timer.
Alternative Flows	**7a. Time Expired** 1. Timer runs out before player answers all questions. 2. System automatically submits current answers and calculates score. **7b. Earn Badge** 1. Player answers all questions correctly. 2. System awards a badge for that difficulty level. 3. System updates badge collection in player profile. **7c. No Badge** 1. Player fails to answer all questions correctly. 2. System does not award a badge. 3. System displays results and records stats. **7d. Player Exits Mid-Game** 1. Player exits before completing all questions. 2. System auto-submits current answers. 3. System calculates score based on answered questions and updates stats.

Use Case ID	UC-PG2
Use Case Name	Play Whiz Memory Match
Created By	Arcielle Marie Gercan
Date Created	September 27, 2025
Description	This use case describes how a player selects and plays the Whiz Memory Match game mode, matching pairs of cards under a timer.
Priority	High
Primary Actor	Player
Secondary Actor	System
Preconditions	1. Player must be logged in. 2. Player must have a valid account.
Postconditions	1. Stats are updated based on performance.

Main Flow	1. Player selects “Whiz Memory Match” on the homepage. 2. System displays difficulty levels (Easy, Average, Difficult). 3. Player chooses a difficulty level. 4. System loads the game board with cards based on difficulty. 5. Player flips cards to find matches. 6. System checks matches, updates score and timer. 7. When game ends, system displays results and updates stats.
Alternative Flows	**5a. Player Exits Mid-Game** 1. Player exits before timer ends. 2. System ends the game and displays results. System updates stats.

Use Case ID	UC-PG3
Use Case Name	Play Whiz Puzzle
Created By	Arcielle Marie Gercan
Date Created	September 27, 2025
Description	This use case describes how a player selects and plays the Whiz Puzzle game mode by arranging scrambled pieces to form a complete image.
Priority	High
Primary Actor	Player
Secondary Actor	System
Preconditions	1. Player must be logged in. 2. Player must have a valid account.
Postconditions	1. Stats are updated based on performance.
Main Flow	1. Player selects “Whiz Puzzle” on the homepage. 2. System displays available difficulty levels (Easy, Average, Difficult) and categories (Solar System, Scientists, The Human Body, Animals, Geometry, Random). 3. System displays scrambled puzzle pieces based on chosen category and difficulty. 4. Player arranges pieces to form the correct image. 5. System validates arrangement and updates score. 6. System displays results and updates stats.
Alternative Flows	**4a. Player Exits Mid-Game** 1. Player exits before puzzle is completed.

	2. System ends the game and displays results. 3. System updates stats.

Use Case ID	UC-PG4
Use Case Name	Play Whiz Battle
Created By	Arcielle Marie Gercan
Date Created	September 27, 2025
Description	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.
Priority	High
Primary Actor	Player
Secondary Actors	System, Opponent Player
Preconditions	1. Player must be logged in. 2. Player must have a valid account.
Postconditions	1. Stats are updated for both players. 2. Winner is declared based on performance.
Main Flow	1. Player selects “Whiz Battle” on the homepage. 2. System displays options (Create Battle or Join Battle). **2.1. Create Battle** 2.1.1. Player chooses “Create Battle.” 2.1.2. System displays categories (Science or Math) and difficulty levels (Easy, Average, Difficult). 2.1.3. Player chooses a category and difficulty. 2.1.4. System creates a battle room and displays the game code. 2.1.5. Opponent joins the battle room. **2.2 Join Battle** 2.2.1. Player enters the game code and clicks “Join.” 2.2.2. Player starts the battle. 2.2.3. System loads a set of questions based on chosen category and difficulty. 2.2.4. Players answer questions in real time. 2.2.5. System calculates stars and determines the winner. 2.2.6. System updates both players’ stats.
Alternative Flows	**2.1.5a. Opponent Fails to Join** 1. Opponent does not join. 2. System prompts player to wait longer or cancel battle.

	2.2.1a. Invalid Code 1. Player enters invalid or expired code. 2. System displays error: “Invalid code. Please try again.” **2.2.4a. Opponent Disconnects** 1. Opponent leaves mid-game. 2. System ends match and declares remaining player winner. 3. Stats are updated accordingly. **2.2.4b. Player Exits Mid-Game** 1. Player quits during active battle. 2. System ends session and declares opponent winner. 3. Stats are updated for both players.

Reward Redemption

Use Case ID	UC-RR0
Use Case Name	Reward Redemption
Created By	Arcielle Marie Gercan
Date Created	September 27, 2025
Description	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.
Priority	High
Primary Actors	Player, Admin
Secondary Actor	System
Preconditions	1. Player must be logged in. 2. Player must have earned all required badges in a difficulty level. 3. Admin must be logged in to process requests.
Postconditions	1. Player's reward claim is recorded and processed. 2. Admin approval or rejection updates the player's profile and system logs.
Includes	1. UC-RR1 Player Requests Reward 2. UC-RR2 Admin Grants Reward

Use Case ID	UC-RR1
Use Case Name	Player Requests Reward
Created By	Arcielle Marie Gercan
Date Created	September 27, 2025
Description	This use case describes how a player requests to redeem a reward after meeting the badge requirements.
Priority	High
Primary Actor	Player
Secondary Actor	System
Preconditions	1. Player must be logged in. 2. Player must have earned all 3 badges in a difficulty level.
Postconditions	1. System records the player's reward request for admin processing.
Main Flow	1. Player selects “Your Badges.” 2. System displays badge collection and shows eligibility to claim a reward per difficulty level. 3. Player selects “Claim” on the chosen difficulty level.

	4. System validates eligibility (e.g., 3 badges earned in one difficulty). 5. System records the request for admin processing.
Alternative Flows	**1a. No Badges Yet** 1. Player opens “Your Badges” without having played any games. 2. System displays empty badge slots. **1b. Claim Button Locked** 1. Player sees “Locked” buttons instead of “Claim.” 2. Player cannot request to claim a reward until requirements are met.

Use Case ID	UC-RR2
Use Case Name	Admin Grants Reward
Created By	Arcielle Marie Gercan
Date Created	September 27, 2025
Description	This use case describes how an admin verifies a reward claim request and grants or rejects it.
Priority	High
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. Player must have submitted a reward request.
Postconditions	1. Reward claim is processed. 2. Player profile is updated to reflect claimed rewards. 3. System logs the reward claim transaction.
Main Flow	1. Admin navigates to “List of Players” in the admin panel. 2. Admin selects the player profile and clicks “View” in the badges column. 3. System displays the player’s badge collection with the “Reward” button unlocked (if eligible). 4. Admin clicks “Reward.” 5. System updates the badge collection of the chosen difficulty level, resetting it to no badges. 6. System logs the reward claim transaction.
Alternative Flows	**3a. Reward Button Locked** 1. Admin views player’s badge collection but the “Reward” button is locked. 2. System indicates the player has not met eligibility.

	3. Admin cannot proceed until requirements are met. **3b. Admin Cancels Reward Action** 1. Admin views eligible reward but chooses not to proceed. 2. System retains current badge collection without resetting.

Manage Admin Account

Use Case ID	UC-MAA0
Use Case Name	Manage Admin Account
Created By	Kelly Dumbrique
Date Created	September 27, 2025
Description	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.
Priority	High
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must have access to the system.
Postconditions	1. Admin account is created, updated, or accessed depending on the sub-function executed.
Includes	1. UC-MAA1 Admin Login 2. UC-MAA6 Admin Logout
Extends	1. UC-MAA2 Add New Admin 2. UC-MAA3 Edit Admin Profile 3. UC-MAA4 Change Admin Password 4. UC-MAA5 Delete Admin

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

Use Case ID	UC-MAA2
Use Case Name	Add New Admin
Created By	Kelly Dumbrique
Date Created	September 27, 2025
Description	This use case describes how an admin adds a new administrator account.
Priority	High
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in.
Postconditions	1. A new admin account is created and added to the list.
Main Flow	1. Admin navigates to List of Admins. 2. Admin clicks “Add New Admin.” 3. System displays form for details (username, password, sex, image). 4. Admin fills out details and submits. 5. System validates input and creates the new account. 6. System confirms successful addition.
Alternative Flows	**4a. Invalid Input** 1. Admin leaves required fields blank or enters invalid details. 2. System displays error and requests correction. **4b. Duplicate Username** 1. Admin enters a username that already exists. System rejects submission and prompts for a unique username.

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

Main Flow	1. Admin navigates to List of Admins. 2. Admin selects an admin and clicks Edit icon. 3. System displays editable fields. 4. Admin updates information and submits. 5. System validates and updates the profile. 6. System confirms successful update.
Alternative Flows	**4a. Invalid Profile Update** 1. Admin enters invalid or empty fields. 2. System displays validation error and requests correction.

Use Case ID	UC-MAA4
Use Case Name	Change Admin Password
Created By	Kelly Dumbrique
Date Created	September 27, 2025
Description	This use case describes how an admin changes their account password.
Priority	Medium
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in.
Postconditions	1. Admin’s password is updated successfully.
Main Flow	1. Admin navigates to List of Admins. 2. Admin selects an admin and clicks Password icon. 3. System prompts for current password and new password. 4. Admin enters the required details. 5. System verifies current password and validates new password format. 6. System updates the password. 7. System confirms successful password change.
Alternative Flows	**3a. Wrong Current Password** 1. Admin enters incorrect current password. 2. System rejects request and asks to retry. **3b. Weak New Password** 1. Admin enters a password that does not meet security requirements. 2. System prompts the admin to enter a stronger password.

Use Case ID	UC-MAA5
Use Case Name	Delete Admin
Created By	Kelly Dumbrique
Date Created	September 27, 2025
Description	This use case describes how an admin deletes an administrator account.
Priority	High
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in.
Postconditions	1. The selected admin account is removed from the list.
Main Flow	1. Admin navigates to List of Admins. 2. Admin selects an admin and clicks Delete icon. 3. System prompts confirmation. 4. Admin confirms deletion. 5. System removes account and updates the list.
Alternative Flows	**3a. Cancel Delete** 1. Admin clicks Delete but cancels at confirmation prompt. 2. System keeps the account unchanged. **4a. Attempt to Delete Self** 1. Admin tries to delete their own account while logged in. 2. System rejects action with message: “You cannot delete your own account while logged in.”

Use Case ID	UC-MAA6
Use Case Name	Admin Logout
Created By	Kelly Dumbrique
Date Created	September 27, 2025
Description	This use case describes how an admin logs out of the system.
Priority	High
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in.
Postconditions	1. Admin is logged out and redirected to the login page.
Main Flow	1. Admin clicks the avatar and selects “Logout.” 2. System ends the session. 3. System redirects admin to login page.

Manage Quiz Settings

Use Case ID	UC-MQS0
Use Case Name	Manage Quiz Settings
Created By	Kelly Dumbrique
Date Created	September 27, 2025
Description	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.
Priority	Medium
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. Question bank and difficulty levels must exist in the database.
Postconditions	1. The question bank and quiz settings are updated based on the admin's actions.
Includes	1. UC-MQS1 Add New Question 2. UC-MQS2 Edit Question Info 3. UC-MQS3 Delete Question 4. UC-MQS4 Set Number of Questions 5. UC-MQS5 Set Stars 6. UC-MQS6 Set Timer

Use Case ID	UC-MQS1
Use Case Name	Add New Question
Created By	Kelly Dumbrique
Date Created	September 27, 2025
Description	This use case describes how an admin adds a new quiz question into the question bank.
Priority	Medium
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. Question bank exists in the database.
Postconditions	1. A new question is added to the question bank.
Main Flow	1. Admin navigates to Questions. 2. Admin clicks "Add New Question." 3. System displays a form for question details (text, options, correct answer, category, year level, difficulty, image).

	4. Admin fills out the form and submits. 5. System validates input. 6. System saves the new question. 7. System confirms successful addition and updates the list.
Alternative Flows	**4a. Invalid Input** 1. Admin leaves required fields blank or enters invalid data. 2. System displays error message and requests correction.

Use Case ID	UC-MQS2
Use Case Name	Edit Question Info
Created By	Kelly Dumbrique
Date Created	September 27, 2025
Description	This use case describes how an admin edits the details of an existing quiz question.
Priority	Medium
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. At least one question exists.
Postconditions	1. The selected question is updated with new details.
Main Flow	1. Admin navigates to Questions. 2. Admin selects an existing question and clicks Edit. 3. System displays editable fields. 4. Admin updates information and submits. 5. System validates and updates the question. 6. System confirms successful update.
Alternative Flows	**4a. Invalid Update** 1. Admin provides invalid or incomplete information. 2. System displays validation error and requests correction.

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

Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. At least one question exists.
Postconditions	1. The selected question is removed from the database.
Main Flow	1. Admin navigates to Questions. 2. Admin selects a question and clicks Delete. 3. System prompts confirmation. 4. Admin confirms deletion. 5. System removes the question and updates the list.
Alternative Flows	**3a. Cancel Delete** 1. Admin cancels at the confirmation prompt. 2. System keeps the question unchanged.

Use Case ID	UC-MQS4
Use Case Name	Set Number of Questions
Created By	Kelly Dumbrique
Date Created	September 27, 2025
Description	This use case describes how an admin sets the number of questions per difficulty level.
Priority	Medium
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. Difficulty levels exist in the database.
Postconditions	1. Number of questions per difficulty level is updated.
Main Flow	1. Admin navigates to Difficulty settings. 2. System displays current number of questions per level. 3. Admin specifies new number of questions. 4. System validates input. 5. System updates the setting and confirms the change.
Alternative Flows	**2a. Insufficient Questions** 1. Admin specifies more than available in the database. 2. System displays warning: “Not enough questions available. Please reduce the number.” **3a. Invalid Number** 1. Admin enters non-numeric or negative value.

	2. System rejects input and prompts for a valid number.

Use Case ID	UC-MQS5
Use Case Name	Set Stars
Created By	Kelly Dumbrique
Date Created	September 27, 2025
Description	This use case describes how an admin sets the star value awarded for correct answers per difficulty level.
Priority	Medium
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. Difficulty levels exist in the database.
Postconditions	1. Star values are updated per difficulty level.
Main Flow	1. Admin navigates to Difficulty settings. 2. System displays current star values. 3. Admin specifies new star values. 4. System updates and confirms the change.
Alternative Flows	**3a. Invalid Star Value** 1. Admin enters invalid star values (negative or zero). 2. System rejects input and prompts for a valid value.

Use Case ID	UC-MQS6
Use Case Name	Set Timer
Created By	Kelly Dumbrique
Date Created	September 27, 2025
Description	This use case describes how an admin sets the time limit per difficulty level.
Priority	Medium
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. Difficulty levels exist in the database.
Postconditions	1. Timer settings are updated per difficulty level.
Main Flow	1. Admin navigates to Difficulty settings. 2. System displays current timer values. 3. Admin specifies a new time limit. 4. System updates and confirms the change.
Alternative Flows	**3a. Invalid Timer Value**

	1. Admin enters invalid timer values (negative, zero, or excessively high). 2. System rejects input and prompts for a valid time.

Manage Player List

Use Case ID	UC-MPL0
Use Case Name	Manage Player List
Created By	Shandrae Lois Quianzon
Date Created	September 27, 2025
Description	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.
Priority	Medium
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. Player list exists in the database.
Postconditions	1. Player list is updated based on the action performed (add, edit, change password, or delete).
Includes	1. UC-MPL1 Add New Player 2. UC-MPL2 Edit Player Profile 3. UC-MPL3 Change Player Password 4. UC-MPL4 Delete Player

Use Case ID	UC-MPL1
Use Case Name	Add New Player
Created By	Shandrae Lois Quianzon
Date Created	September 27, 2025
Description	This use case describes how an admin adds a new player into the system.
Priority	Medium
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. Player list exists in the database.
Postconditions	1. A new player is added to the player list.
Main Flow	1. Admin navigates to Player List. 2. System displays the list of players. 3. Admin clicks "Add New Player." 4. System displays a form for player details (username, password, school, age, category, sex, region, province, city, avatar). 5. Admin fills out the form and submits. 6. System validates input. 7. If valid, system saves the new player.

	8. System confirms successful addition and updates the list.
Alternative Flows	**5a. Invalid Input** 1. Admin leaves required fields blank or enters invalid data. 2. System displays error message and requests correction. **5b. Duplicate Username** 1. Admin enters a username that already exists. 2. System rejects submission and prompts for a unique username.

Use Case ID	UC-MPL2
Use Case Name	Edit Player Profile
Created By	Shandrae Lois Quianzon
Date Created	September 27, 2025
Description	This use case describes how an admin edits the details of an existing player profile.
Priority	Medium
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. At least one player exists.
Postconditions	1. Player profile information is updated.
Main Flow	1. Admin navigates to Player List. 2. Admin selects an existing player and clicks Edit. 3. System displays editable fields. 4. Admin updates information and submits. 5. System validates and updates the profile. 6. System confirms successful update.
Alternative Flows	**4a. Invalid Update** 1. Admin enters invalid or incomplete data. 2. System displays validation error and requests correction.

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

Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. At least one player exists.
Postconditions	1. Player password is updated in the system.
Main Flow	1. Admin navigates to Player List. 2. System prompts for current password and new password. 3. Admin enters the required details with the help of the Player. 4. System verifies current password and validates new password format. 5. System updates the password. 6. System confirms successful password change.
Alternative Flows	**3a. Wrong Current Password** 1. Admin enters incorrect current password. 2. System rejects request and asks to retry. **3b. Weak Password** 1. Admin enters a password that does not meet security requirements. 2. System rejects and prompts for a stronger password.

Use Case ID	UC-MPL4
Use Case Name	Delete Player
Created By	Shandrae Lois Quianzon
Date Created	September 27, 2025
Description	This use case describes how an admin deletes a player account from the system.
Priority	Medium
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. At least one player exists.
Postconditions	1. The selected player account is removed from the system.
Main Flow	1. Admin navigates to Player List. 2. Admin selects a player and clicks Delete. 3. System prompts confirmation. 4. Admin confirms deletion. 5. System removes the player and updates the list.
Alternative Flows	**3a. Cancel Delete** 1. Admin cancels at the confirmation prompt.

	2. System keeps the player account unchanged.

View Analytics Dashboard

Use Case ID	UC-VAD1
Use Case Name	View Analytics Dashboard
Created By	Janice Maxene Salipande
Date Created	September 27, 2025
Description	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.
Priority	High
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. Analytics data must be available in the database.
Postconditions	1. Admin views and interprets analytics reports. 2. Data insights are displayed but no changes are made to the database.
Main Flow	1. Admin navigates to the Analytics Dashboard. 2. System displays dashboard sections, which may include: a. Total Registered Players b. Most Played Game Mode c. Average Session Duration d. Male vs. Female Registered Players (pie chart) e. Age Distribution of Players (bar chart) f. Registered Players by Region (bar chart) g. Male vs. Female Players Per Game Mode (bar chart) h. Reward Distribution by Gender and Level (bar chart) i. Most Played Game Mode by Age (bar chart) 3. Admin interacts with sort/filter options. 4. System updates and redisplays graphs based on selected filters. 5. Admin views insights and trends. 6. Admin clicks Export icon. 7. System generates an Excel file and prompts the admin to download it.
Alternative Flows	**2a. No Data Available** 1. System detects no available data (e.g., new system with no players yet).

	2. System displays an empty state message such as “No analytics data available yet.” **4a. Invalid Filter Selection** 1. Admin applies a filter combination that returns no results (e.g., selecting a date range before launch). 2. System displays “No results found for the selected criteria.” **4b. System Error During Data Load** 1. If the dashboard fails to retrieve analytics data due to a system error. 2. System displays an error message and prompts the admin to retry. **6a. Export Fails** 1. If the system encounters an error during Excel export (e.g., file generation issue). 2. System displays “Export failed. Please try again.”

View Leaderboard

Use Case ID	UC-VLB0
Use Case Name	View Leaderboard
Created By	Janice Maxene Salipande
Date Created	September 27, 2025
Description	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.
Priority	High
Primary Actors	Player, Admin
Secondary Actor	System
Preconditions	1. Player must be logged in. 2. Admin must be logged in to access the admin panel. 3. Leaderboard data must be available in the system database.
Postconditions	1. Leaderboard data is successfully displayed to the actor. 2. Admin may export the leaderboard data to Excel.
Includes	1. UC-VLB1 Player View Leaderboard 2. UC-VLB2 Admin View Leaderboard

Use Case ID	UC-VLB1
Use Case Name	Player View Leaderboard
Created By	Janice Maxene Salipande
Date Created	September 27, 2025
Description	This use case describes how a player views the leaderboard for Whiz Challenge and Whiz Battle game modes.
Priority	High
Primary Actor	Player
Secondary Actor	System
Preconditions	1. Player must be logged in. 2. At least one player must have played a game to generate leaderboard data.
Postconditions	1. Leaderboard data is displayed.
Main Flow	1. Player navigates to the Leaderboard page. 2. System displays two leaderboard sections: a. Whiz Challenge Leaderboard (Top 10)

	i. Username ii. Total Rewards for All Levels iii. Rewards for Easy, Average, Difficult iv. Last Reward Claim date/time b. Whiz Battle Leaderboard (Top 10) i. Username ii. Total Stars iii. Last Battle date/time 3. Player reviews leaderboard standings.
Alternative Flows	**2a. No Leaderboard Data** 1. System detects that no players have participated yet in either game mode. 2. System displays “No leaderboard data available yet.” **2b. System Error While Loading Leaderboard** 1. If the system fails to fetch leaderboard data due to a technical issue. 2. System displays an error message and prompts the admin to retry.

Use Case ID	UC-VLB2
Use Case Name	Admin View Leaderboard
Created By	Janice Maxene Salipande
Date Created	September 27, 2025
Description	This use case describes how an admin views the leaderboard for Whiz Challenge and Whiz Battle game modes.
Priority	High
Primary Actor	Admin
Secondary Actor	System
Preconditions	1. Admin must be logged in. 2. At least one player must have played a game to generate leaderboard data.
Postconditions	1. Leaderboard data is displayed based on selected filters. 2. Admin may export the leaderboard data to Excel.
Main Flow	1. Admin navigates to the Leaderboard page. 2. System displays two leaderboard sections: a. Whiz Challenge Leaderboard (Top 10) i. Username ii. Total Rewards for All Levels

	iii. Rewards for Easy, Average, Difficult iv. Last Reward Claim date/time b. Whiz Battle Leaderboard (Top 10) i. Username ii. Total Stars iii. Last Battle date/time 3. Admin reviews leaderboard standings. 4. Admin clicks Export option. 5. System generates an Excel file containing leaderboard data and prompts the admin to download it.
Alternative Flows	**2a. No Leaderboard Data** 1. System detects that no players have participated yet in either game mode. 2. System displays “No leaderboard data available yet.” **2b. System Error While Loading Leaderboard** 3. If the system fails to fetch leaderboard data due to a technical issue. 4. System displays an error message and prompts the admin to retry. **4a. Export Fails** 1. If the system encounters an error during Excel export. 2. System displays “Export failed. Please try again.”