

Conor Nash

K00278909

Technological University of the Shannon

FYP

Use case documentation

TABLE OF CONTENTS

Use Case No.1a, 1b	4
Use Case No.2.....	6
Use Case No.3.....	7
Use Case No.4.....	8
Use Case No.5.....	10
Use Case No.6.....	11
Use Case No.7	12
Use Case No.8.....	13
Use Case No.9.....	15
Use Case No.9.....	17
Use Case No.10	18
Use Case No.11	19
Use Case No.12	20

USE CASE DIAGRAM

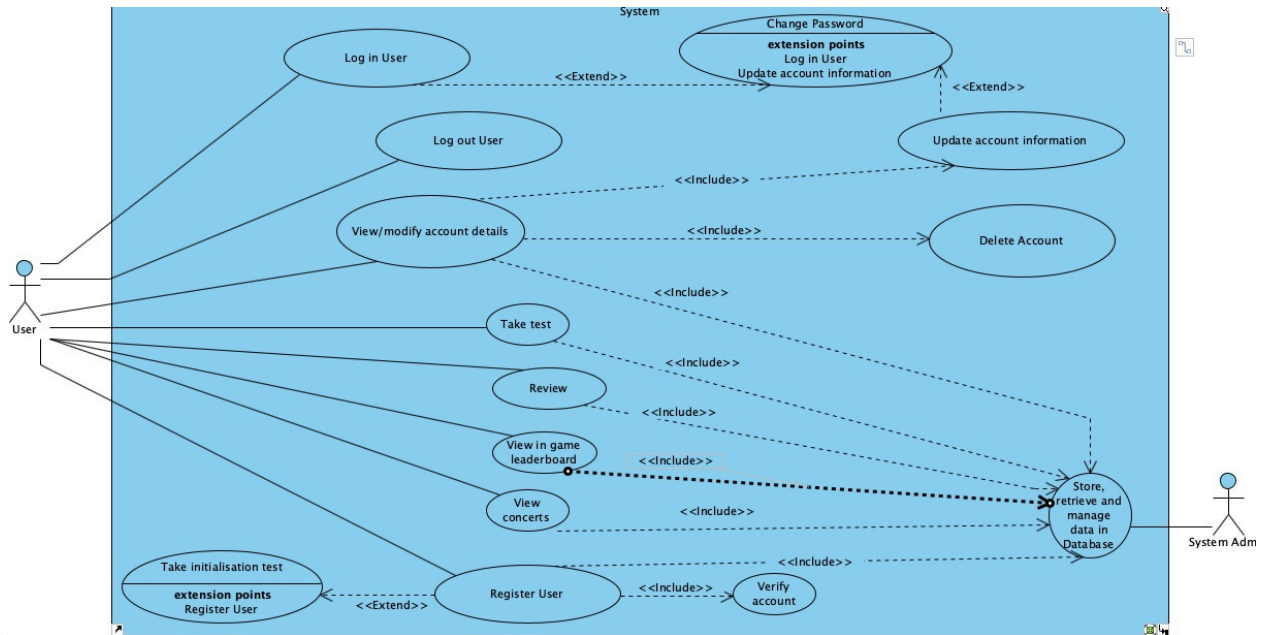


Figure 1 - Use case diagram

USE CASE DOCUMENTATION

USE CASE NO.1A, 1B	Participating Actor(s): Website User	Sheet 1 of 12
<p>Use Case Name:</p> <p>Register User</p> <p>Take initialisation test</p>		
<p>Description:</p> <p>User registers to system to enable login.</p>		
<p>Entry Condition(s):</p> <p>User has accessed the home page on their device</p>		
<p>Flow of Events:</p> <ol style="list-style-type: none"> (1) User selects 'Start learning music today!' (register) option. (2) System displays the question: "Do you have experience reading music?" and the answer options "None at all, I would like to start from scratch", "Yes! I would like to take an initialisation test" (3) User submits response. <ul style="list-style-type: none"> If (user selects "Yes") { System will display an initialisation test with ten questions. Based on the user's responses - how many correct and incorrect answers are submitted - the system will dynamically display questions of varying difficulties. E.g., if the user gets 2 correct questions in a row, a set of more difficult questions will be displayed and vice versa for easier questions. The system will then use that to estimate the user's ability, and, subsequently, restrict more difficult categories on main page, and make easier categories available to test. } (4) System asks user for their first name. (5) User enters name. (6) System asks user for their email address. (7) User submits email (8) System displays a "create password" screen. (9) User enters & confirms new password. (10) System sends a link to the users' email address, requesting to validate the account (see verify account use case) 		

(11) System displays main menu screen, with all available test categories (e.g. 'Basics 1') and a horizontal nav bar menu displaying the various site pages: Profile, Leader board, Concerts, Log out.
Exit Condition(s): User has now created an account and is logged in
<p>Alternate Flows / Exceptions Handling:</p> <p>User has not supplied a valid email address.</p> <p>User has not supplied a valid password.</p>
<p>Technology / Data Variations:</p> <p>Technology: CSS, HTML, JS, Java, PHP</p> <p>String: password, email, displayName</p> <p>Int: correctQuestions, incorrectQuestions, userScore, userAbilityLevel</p> <p>Boolean: takeInitialisationTest, loggedIn</p>

USE CASE NO.2	Participating Actor(s): Website User	Sheet 2 of 12
Use Case Name: Log in User		
Description: User logs in to system		
Entry Condition(s): User has accessed the home page on their device and has an account.		
Flow of Events: <ol style="list-style-type: none"> (1) User selects 'Already have an account?' (log in) option. (2) System displays the log in screen, prompting user for username and password. (3) System also displays 'Forgotten your password?' (change password) option (see Change Password use case). (4) User enters their username and password. (5) System displays main menu screen, with all available test categories (e.g. 'Basics 1') and a horizontal nav bar menu displaying the various site pages: Profile, Leader board, Concerts, Log out. 		
Exit Condition(s): User is logged in		
Alternate Flows / Exceptions Handling: User has incorrect email details. User has incorrect password details.		
Technology / Data Variations: Technology: CSS, HTML, JS, Java, PHP String: password, email Boolean:, loggedIn.		

USE CASE NO.3	Participating Actor(s): Website User	Sheet 3 of 12
Use Case Name: Log Out User		
Description: User logs out of system		
Entry Condition(s): User has an account and is logged in.		
Flow of Events: (1) User selects the 'Log Out' option at the top of the horizontal navigation bar of the main pages of the website (home, concerts, leader board, personal profile) (2) System displays the log in screen, prompting user for username and password.		
Exit Condition(s): User is logged out		
Alternate Flows / Exceptions Handling:		
Technology / Data Variations: Technology: CSS, HTML, JS, Java, PHP String: userEmail Boolean: loggedIn.		

USE CASE NO.4	Participating Actor(s): Website User	Sheet 4 of 12
Use Case Name: Change Password		
Description: User changes password for their account		
Entry Condition(s): User is registered with system but does not have correct password details User is on their personal details profile and would like to change password		
Flow of Events: (1) System displays the log in screen, prompting user for username and password OR User selects change password option on the personal details page. (2) If trying to log in{ The system requests the users email address. User enters email address. System sends email, requesting validation. User clicks on link in email. } else { System requests current password The user enters password. } (3) System requests new password and password to be retyped. (4) User enters Information. (5) System stores new password information in database.		

Exit Condition(s): User has successfully changed password

Alternate Flows / Exceptions Handling:

User cannot remember current password

Technology / Data Variations:

Technology: CSS, HTML, JS, Java, PHP

String: password, email

Boolean: loggedIn.

USE CASE NO.5	Participating Actor(s): Website User	Sheet 5 of 12
Use Case Name: View/Modify account details		
Description: User views and modifies personal profile details		
Entry Condition(s): User has logged into system and is on the main pages of the website		
Flow of Events: <ol style="list-style-type: none"> (1) User selects the 'View Profile option at the top of the horizontal navigation bar of the main pages of the website (2) The system retrieves the user data from the database (3) System displays user's personal profile, with all of the user's achievements (current streak, longest streak, member since, milestones/awards, total experience points). System also displays the options to delete account and update account information (see Update account information and Delete account use cases). 		
Exit Condition(s): User has successfully viewed their profile		
Alternate Flows / Exceptions Handling:		
Technology / Data Variations: Technology: CSS, HTML, JS, Java, PHP String: userFirstName, userAchievements Int: correctStreak, longestStreak, memberSince, userExperiencePoints Boolean: isLoggedIn		

USE CASE NO.6	Participating Actor(s): Website User	Sheet 6 of 12
Use Case Names: Update account information		
Description: User modifies personal profile details		
Entry Condition(s): User has logged into system, has accessed their personal profile, and wishes to update their details		
Flow of Events: (1) User selects 'Edit my account' (2) The system retrieves the user account details from database and displays them on screen (3) The user selects the details they wish to modify: Change password (see use case), change Display name, change email address, change avatar (4) The system prompts the user to enter their new details (5) The user enters details (6) The user's details are updated within the database.		
Exit Condition(s): User has successfully modified their personal details		
Alternate Flows / Exceptions Handling:		
Technology / Data Variations: Technology: CSS, HTML, JS, Java, PHP String: userDisplayName, userPassword, userEmail Img: userAvatar Boolean: isLoggedIn		

USE CASE NO.7	Participating Actor(s): Website User	Sheet 7 of 12
Use Case Names: Delete account		
Description: User deletes their account		
Entry Condition(s): User has logged into system, has accessed their personal profile, and wishes to delete their account		
Flow of Events: <ol style="list-style-type: none"> (1) User selects 'Delete my account' (2) The system displays: 'We're sad to see you go! Can you tell us why?' and radio options "Not really interested anymore", "I found a better app", "I'll come back later, I promise!", "Rather not say". (3) The user submits their response (4) The system retrieves response and stores it within database. (5) The system requests the user to confirm password (6) User enters password (7) System validates that the password is correct. (8) System sends data to 'accounts to be deleted' database. (9) The system displays the home page of the website. 		
Exit Condition(s): User has successfully deleted their account		
Alternate Flows / Exceptions Handling: User enters an incorrect password		
Technology / Data Variations: Technology: CSS, HTML, JS, Java, PHP, MySQL String: userEmail, userPassword, userResponse Boolean: isLoggedIn		

USE CASE NO.8	Participating Actor(s): Website User	Sheet 8 of 12
Use Case Names: Take test		
Description: User takes a test to develop a particular musical skill		
Entry Condition(s): User has logged into system and is on the main page		
Flow of Events: <ol style="list-style-type: none"> (1) User selects the available category they wish to test themselves in. (2) System will display a short test with ten questions. The user will receive three hearts to symbolise how many errors they will be able to make. Moreover, each question, pertaining to the category chosen, will be randomly selected with multiple choice options displayed for each question. If the user answers a question incorrectly, a heart will be deducted and, if all three hearts are used up, the quiz will terminate. Towards the end of the quiz, if the user is still playing, the previously incorrect questions will be asked again. For every heart remaining towards the end of the quiz, the user will receive a portion of experience points (e.g. 1 heart – 15 exp, 2 hearts – 20 exp; 3 hearts – 25 exp). The system will display explanations of each question after they are answered. (3) The user takes the test. (4) If successful, the system will print out the total exp the user has received. (5) The system retrieves exp points and adds it to the users total exp points. (6) The system receives the incorrect questions and adds it to any array, which will be used to formulate the review sections of the website. (see use case) (7) The system updates this data in the database. 		
Exit Condition(s): User has successfully taken the quiz and has received exp points.		
Alternate Flows / Exceptions Handling: User has failed the quiz. A display screen, which prints 'hard luck', will be displayed to the user, and will return them to the main page (or they can click the 'try again' button to retake test)		
Technology / Data Variations: Technology: CSS, HTML, JS, Java, PHP		

String: questions[], answers[], reviewQuestionsForTopic[].

Int: correctAnswers, incorrectAnswers, userExp, testExp, heartCount

Boolean: isLoggedIn, gameOver, hasHearts,

USE CASE NO.9	Participating Actor(s): Website User	Sheet 9 of 12
Use Case Names: Review		
Description: User reviews questions they previously got incorrect		
Entry Condition(s): User has logged into system and is on the main page		
Flow of Events: <ol style="list-style-type: none"> (1) User selects the Review questions option. (2) System will display a short test with previously incorrect options from their tests. The user will receive three hearts to symbolise how many errors they will be able to make. Moreover, each question, pertaining to the category chosen, will be randomly selected with multiple choice options displayed for each question. If the user answers a question incorrectly, a heart will be deducted and, if all three hearts are used up, the quiz will terminate. Towards the end of the quiz, if the user is still playing, the previously incorrect questions will be asked again. For every heart remaining towards the end of the quiz, the user will receive a portion of experience points (e.g. 1 heart – 15 exp, 2 hearts – 20 exp; 3 hearts – 25 exp). The system will display explanations of each question after they are answered. (3) The user takes the test. (4) If successful, the system will print out the total exp the user has received. (5) The system retrieves exp points and adds it to the users total exp points. (6) The system receives the incorrect questions and adds it to any array, which will be used to formulate the review sections of the website (see use case). (7) The system updates this data in the database. 		
Exit Condition(s): User has reviewed old questions and has received exp points.		
Alternate Flows / Exceptions Handling: User has no questions to review. The system will display the message: “Nothing to review. You’re doing great!”		
Technology / Data Variations: Technology: CSS, HTML, JS, Java, PHP		

String: questions[], answers[], reviewQuestionsForTopic[].

Int: correctAnswers, incorrectAnswers, userExp, testExp, heartCount

Boolean: isLoggedIn, gameOver, hasHearts

USE CASE NO.9	Participating Actor(s): Website User	Sheet 9 of 12
Use Case Names: View in game leader board		
Description: User views the leader board to see the users with the highest exp points scores.		
Entry Condition(s): User has logged into system and is on the main page		
Flow of Events: (1) User selects the View leader board option. (2) The system gets up-to-date information from the database regarding the users with the highest exp points (3) The system will display a leader board, showcasing the top 20 users of the website.		
Exit Condition(s): User has viewed the leader board		
Alternate Flows / Exceptions Handling: Not enough users have used the application to create the leaderboard.		
Technology / Data Variations: Technology: CSS, HTML, JS, Java, PHP, MySQL String: highestPointsUsers[], totalExperiencePoints[], Int: correctAnswers, incorrectAnswers, userExp, testExp, heartCount Boolean: isLoggedIn, gameOver, hasHearts		

USE CASE NO.10	Participating Actor(s): Website User	Sheet 10 of 17
Use Case Names: View concerts		
Description: User views an available list of concerts		
Entry Condition(s): User has logged into system and is on the main page		
Flow of Events: <ol style="list-style-type: none"> (1) User selects the View concerts option. (2) The system displays a list of concerts a date, time, and performers information (3) User selects the desired concert (4) The system displays a video screen of the concert, the option to 'dim the lights', program notes, and a counter for other people watching at the same time. 		
Exit Condition(s): User has successfully watched a concert		
Alternate Flows / Exceptions Handling: If the concert has is not on at that time, the system will supply how much time is left until the correct start time		
Technology / Data Variations: Technology: CSS, HTML, JS, Java, PHP, MySQL String: concertDateAndTime[[]], concertType, concertArtist Int: viewerCounter Boolean: concertOn		

USE CASE NO.11	Participating Actor(s): Website User	Sheet 11 of 12
Use Case Names: Verify account		
Description: User confirms email address		
Entry Condition(s): User has logged into system and is on the main page		
Flow of Events: <ol style="list-style-type: none"> (1) After registering, system sends the user an email to validate their account (2) User opens email account and clicks link on email address. (3) The system redirects the user back to the website and displays a call to action button, asking to confirm account. (4) The user clicks this button. (5) The system records that the user is now confirmed and stores their status in the users database. 		
Exit Condition(s): User is now confirmed and has a valid email address		
Alternate Flows / Exceptions Handling: The user did not supply a correct email address and no link was issue to the user's email address.		
Technology / Data Variations: Technology: CSS, HTML, JS, Java, PHP, MySQL String: userEmail Boolean: isValidated		

USE CASE NO.12	Participating Actor(s): Administrator	Sheet 12 of 12
Use Case Names: Manage data in database		
Description: System administrator reviews user data in database		
Entry Condition(s): System administrator has accessed the mySQL database and can see all available options.		
Flow of Events: <ol style="list-style-type: none"> (1) The system presents all the database tables that can be managed: usersFlaggedForDeletion, users, concerts, administrators, userDetails (2) System administrator selects the correct database table and interacts with the data. 		
Exit Condition(s): Administrator has correctly viewed/reviewed/modified the data		
Alternate Flows / Exceptions Handling:		
Technology / Data Variations: Technology: PHP, mySQL		