**Use Cases**

**for**

**GoldFolks**

**Version 1.1 approved**

**Prepared by**

Ng Chi Hui

Lionel Teo Zhi Neng

Zachary Varella Lee Zheyu

**Nanyang Technological University**

**Created on 27/08/21**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Ng Chi Hui,  Zachary Varella Lee Zheyu,  Lionel Teo Zhi Neng | 27/08/21 | Initial Draft | 0.10 |
| Lionel Teo Zhi Neng | 28/09/21 | Updated use cases 3.7, 3.8, and 3.9 |  |
| Ng Chi Hui,  Zachary Varella Lee Zheyu,  Lionel Teo Zhi Neng | 04/09/21 | Amended use case descriptions to match SRS | 1.0 |
| Ng Chi Hui,  Zachary Varella Lee Zheyu,  Lionel Teo Zhi Neng | 11/09/21 | Change of Format | 1.1 |

**Use Case List**

|  |  |
| --- | --- |
| ***Primary Actor*** | ***Use Cases*** |
| User (Initiating), Database (Supporting) | 1.1 – Login Screen |
| - | 1.2 – Create Account |
| - | 1.3 – Forget Password |
| User (Initiating) | 1.4 – Select Service |
| User (Initiating) | 2.1 – Edit Reminder |
| Reminder Database (Supporting) | 2.2 – Delete Record |
| Reminder Database (Supporting) | 2.3 – Create New Record |
| Reminder Database (Supporting) | 2.4 – Change Record |
| User (Initiating) | 2.5 – Acknowledge Reminder |
| Reminder Database (Supporting) | 2.6 – Push Notification |
| User (Initiating) | 3.1 – Select Cognitive Game to Play |
| - | 3.2 – Launch Mental Math Game |
| - | 3.3 – Generate Math Questions & Answers |
| - | 3.4 – Check Answer for Mental Math |
| - | 3.5 – View Mental Math Tutorial |
| Leaderboard Database (Supporting) | 3.6 – View Mental Math Leaderboard |
| - | 3.7 – Launch Simon Says Game |
| - | 3.8 – Check Pattern for Simon Says |
| - | 3.9 – Create Pattern |
| - | 3.10 – View Simon Says Tutorial |
| Leaderboard Database (Supporting) | 3.11 – View Simon Says Leaderboard |
| User (Initiating) | 4.1 – Select Exercise Video |
| YouTube API (Supporting) | 4.2 – Load Exercise Video |

**1. General Use Case Diagram**

1. Login Screen

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.1 | | |
| Use Case Name: | Login | | |
| Created By: | Zachary Varella Lee Zheyu | Last Updated By: | Ng Chi Hui |
| Date Created: | 27/08/2021 | Date Last Updated: | 11/09/2021 |

|  |  |
| --- | --- |
| Actor: | User (Initiating), Database (Supporting) |
| Description: | The user must be able to login to use the app. |
| Trigger | The user opens the app for the first time. |
| Preconditions: | 1. The system must be active. |
| Postconditions: | 1. The user has logged in to the app. |
| Normal Flow: | 1. The user starts the application for the first time. 2. The system will launch the ‘Login Screen’ with fields for ‘Username’ and ‘Password’, and three buttons for ‘Login’, ‘Create Account’, and ‘Forget Password’. 3. The user will enter their username in the ‘Username’ field. 4. The user will enter their password in the ‘Password’ field. 5. The user taps the ‘Login’ button. 6. The system queries the database for the username and retrieves the encoded password. 7. The system checks the entered password with the appropriate encryption. 8. The system logs the user in and enters the main view. |
| Alternative Flows: | AF-S7: Wrong Password   1. An error message is displayed, saying “Please check your username/password!” 2. The app clears the password field. 3. The user acknowledges the error message. 4. The use case returns to step 4.   AF-S3: Create Account   1. The user taps the ‘Create Account’ button. 2. The use case uses the extended Use Case 1.2 ‘Create Account’.   AF-S3: Forget Password   1. The user taps the ‘Forget Password’ button. 2. The use case uses the extended Use Case 1.3 ‘Forget Password’ |
| Exceptions: | EX-1: Unable to connect to database   1. The system displays a message “Connection cannot be established to the server. Please try again!” 2. The user acknowledges the message. 3. The use case returns to step 2. |
| Includes: | Use Case 1.2: “Create Account”  Use Case 1.3: “Forget Password” |
| Priority: | High |
| Frequency of Use: | Low - When the user initially opens the app. |
| Business Rules: | Each device can only have 1 user logged in at all times |
| Special Requirements: | SR-1: The system should check the username/password within 2 seconds. |
| Assumptions: | - |
| Notes and Issues: | - |

1. Create Account

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.2 | | |
| Use Case Name: | Create Account | | |
| Created By: | Ng Chi Hui | Last Updated By: | Ng Chi Hui |
| Date Created: | 27/08/2021 | Date Last Updated: | 11/09/2021 |

|  |  |
| --- | --- |
| Actor: | - |
| Description: | The user must be able register for a new user account |
| Trigger: | The user opens the app for the first time and taps on the ‘Create Account’ button. |
| Preconditions: | 1. User must be on the ‘Login’ page. 2. User account must not already exist in the database (Firebase) 3. User’s email must have a valid email address. |
| Postconditions: | 1. User successfully registered for an account. 2. The system will redirect the user to the Login page for login upon successfully account creation. |
| Normal Flow: | 1. The user taps on the ‘Create Account’ button in the login UI. 2. The system shall load the registration UI for the users to input their information. 3. The user must input a valid email address, full name and password. 4. The user taps on the ‘Register’ button. 5. The system validates that the credentials entered are valid. 6. The system shall ensure that the registrant’s email does not exist in the current database. 7. The system must validate that the password keyed in the password fields are identical. 8. The system must ensure that the password fulfils that criterion of being at least 8 alphanumeric characters. 9. The system must display a dialog box, ‘Account created successfully’ upon successful account creation. 10. The system shall redirect the user back to the ‘Login Page’. 11. The system returns to the calling Use Case 1.1 ‘Login’. |
| Alternative Flows: | AF-S6: Email already exists in the database   1. System queries and finds a record with the same email address. 2. The app will display a message saying “Sorry, this email is already in use” 3. The user acknowledges the message. 4. The app returns to step 3.   AF-S5: Invalid credentials   1. The app will display a message saying “Please make sure you are not using any special characters in the username or password.” 2. The user acknowledges the message. 3. The app returns to step 3. |
| Exceptions: | - |
| Includes: | - |
| Priority: | High |
| Frequency of Use: | Low - Once per Lifetime |
| Business Rules: | Each email address can only be assigned to 1 user. |
| Special Requirements: | - |
| Assumptions: | - |
| Noes and Issues: | - |

1. Forget Password

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.3 | | |
| Use Case Name: | Select Service | | |
| Created By: | Ng Chi Hui | Last Updated By: | Ng Chi Hui |
| Date Created: | 27/08/21 | Date Last Updated: | 11/09/2021 |

|  |  |
| --- | --- |
| Actor: | - |
| Description: | The user will be able to get a new password if they forget their password. |
| Trigger: | The user opens the app and taps on the forget password button. |
| Preconditions: | 1. The user must have an existing account.   AND   1. The user must be in the ‘Login’ page. |
| Postconditions: | 1. The system successfully sends an email with the new password. |
| Normal Flow: | 1. The user taps on the ‘Forget Password’ button in the Login UI. 2. The system shall load the Forget Password UI for the user to input their email address. 3. The user shall input their email address. 4. The user must tap the ‘Confirm’ button, 5. The system shall validate that the email address entered exists in the database. 6. The system must send the user an email with a link to reset their password. 7. The user clicks on the link on their mobile device to launch the app. 8. The app redirects to the ‘Reset Password’ website with a field for new password. 9. Upon tapping on the ‘Confirm’ button, the user shall be redirected back to the login page. |
| Alternative Flows: | AF-8: Email Address does not exist   1. The email is not sent. 2. The system returns to the login page. |
| Exceptions: | - |
| Includes: | - |
| Priority: | Low |
| Frequency of Use: | Low - the user will rarely forget their own password. |
| Business Rules: | - |
| Special Requirements: | The email containing the new created password shall reach the user in less than 10 minutes. |
| Assumptions: | - |
| Notes and Issues: | - |

1. Select Service

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.4 | | |
| Use Case Name: | Select Service | | |
| Created By: | Zachary Varella Lee Zheyu | Last Updated By: | Ng Chi Hui |
| Date Created: | 04/09/21 | Date Last Updated: | 11/09/2021 |

|  |  |
| --- | --- |
| Actor: | User (Initiating) |
| Description: | The user will be able to select a service by tapping a tab on the bottom of the screen. |
| Trigger: | The user logs into the app or returns to the main page. |
| Preconditions: | 1. The user must have logged in. |
| Postconditions: | 1. The screen should switch to the appropriate service. |
| Normal Flow: | 1. The system loads the ‘Home Screen’ upon start up. 2. The app will display four buttons in a list view - “Medication Reminders”, “Cognitive Games”, “Exercise Videos”, and “Settings”. 3. The user taps on “Cognitive Games” to go to the Games Screen (Use Case Description Set 3) 4. The user taps on “Exercise Videos” to go the Videos Screen (Use Case Description Set 4) 5. The user taps on “Medication Reminders” to return to the Reminders Screen “Use Case Description Set 2) |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Priority: | High |
| Frequency of Use: | Every time the user accesses a different functionality. |
| Business Rules: | - |
| Special Requirements: | SR-1: The app should switch between screens within 0.5 seconds. |
| Assumptions: | - |
| Notes and Issues: | - |

**2. Medication Reminder Use Case Description**

1. Edit Reminder

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.1 | | |
| Use Case Name: | Edit Reminder | | |
| Created By: | Zachary Varella Lee Zheyu | Last Updated By: | Ng Chi Hui |
| Date Created: | 27/08/21 | Date Last Updated: | 11/09/2021 |

|  |  |
| --- | --- |
| Actor: | User (Initiating) |
| Description: | Upon opening the reminder screen, the user will be able to edit the timing and frequency of reminders, create new reminders, and delete reminders. |
| Trigger: | The user taps on the ‘Reminder Button’ in the reminder screen. |
| Preconditions: | 1. The user must be on the Reminder Screen   AND   1. The user must have allowed push notifications |
| Postconditions: | 1. Reminders must be changed according to user’s actions.   AND   1. Reminder must be saved in the database. |
| Normal Flow: | 1. The user must tap the reminder button. 2. The user must click on edit reminders to allow editing of reminders. 3. The user may tap the ‘Edit Reminder’ button to call the included use case ‘Change Record’. 4. The user may tap the ‘Create Reminder’ button to call the included use case Create New Record’. 5. The user may tap the ‘Delete Reminder’ button to call the included use case Delete Record’. |
| Alternative Flows: | AF-S1: The user has not allowed push notifications   1. The app will prompt users to edit the notification settings in their phone settings. 2. The app returns to step 1. |
| Exceptions: | - |
| Includes: | Use Case 2.2: “Delete Record”  Use Case 2.3: “Create New Record”  Use Case 2.4: “Change Record” |
| Priority: | Medium |
| Frequency of Use: | When the user wants to edit his/her reminder |
| Business Rules: | All updates must be reflected in both offline and online database. |
| Special Requirements: | SR-1: The reminder screen should load within 1 second of tapping the button. |
| Assumptions: | - |
| Notes and Issues: | - |

1. Delete Record

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.2 | | |
| Use Case Name: | Delete Record | | |
| Created By: | Ng Chi Hui | Last Updated By: | Ng Chi Hui |
| Date Created: | 27/08/21 | Date Last Updated: | 11/09/2021 |

|  |  |
| --- | --- |
| Actor: | Reminder Database (Supporting) |
| Description: | The database would carry out the removal of the record. |
| Trigger: | Triggered by the calling use case ‘Edit Reminder’ when the user taps on the ‘Delete Reminder’. |
| Preconditions: | 1. The reminder record must first be found before it can be removed.   AND   1. Connection to the database must be established. |
| Postconditions: | 1. The reminder database should be updated with the chosen record being removed.   AND   1. The updated data must be returned to the calling use case and reflected in the app. |
| Normal Flow: | 1. The calling use case ‘Edit Reminder’ will inform the system when the user taps on the ‘Delete Reminder’ button. 2. The system to delete the selected record in the database. 3. Upon successful deletion of the record, the app returns to the calling use case. |
| Alternative Flows: | - |
| Exceptions: | EX-1: Deletion of Reminder record fails   1. An error notification will pop-up, saying “An error has occurred. Please try removing again.” 2. The user acknowledges the notification. 3. The app returns to the ‘Reminders’ main screen. |
| Includes: | - |
| Priority: | Medium |
| Frequency of Use: | Low Frequency – Once set up properly, the user would most likely continue using the same medication reminders. The medication reminders would most likely change once every few months and/or when the prescription changes. |
| Business Rules: | - |
| Special Requirements: | SR-1: The reminder screen with the updated information should load within 1 second of tapping the button. |
| Assumptions: | - |
| Notes and Issues: | - |

1. Create New Record

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.3 | | |
| Use Case Name: | Create New Record | | |
| Created By: | Zachary Varella Lee Zheyu | Last Updated By: | Ng Chi Hui |
| Date Created: | 27/08/21 | Date Last Updated: | 11/09/2021 |

|  |  |
| --- | --- |
| Actor: | Reminder Database (Supporting) |
| Description: | The system will add a new reminder to the Reminder Database according to the user’s input. |
| Trigger: | Triggered by the calling use case ‘Edit Reminder’ when the user taps on the ‘Create Reminder’ button. |
| Preconditions: | 1. The reminder record must exist in the database.   AND   1. Connection to the database must be established. |
| Postconditions: | 1. The chosen reminder record is updated and saved in the database. |
| Normal Flow: | 1. The calling use case ‘Edit Reminder’ will inform the system when the user taps on the ‘Create Reminder’ button. 2. The screen will show a menu pop-up containing editable fields for ‘Name’, ‘Description’, ‘Frequency’, and ‘Time’, a cancel button and a ‘Done’ button. 3. Upon selecting a field, a keyboard will appear allowing the user to type the new field information. 4. The user taps the ‘Done’ button. 5. The app uses a SQL query to insert the record accordingly. 6. The pop-up closes. 7. The app returns to the previous screen. |
| Alternative Flows: | AF-S3: The user taps the ‘cancel’ button   1. The pop-up closes. 2. The app returns to step 7. |
| Exceptions: | EX-1: Connection to the database is lost.   1. Error pops up saying “An error occurred. Information could not be saved.” 2. The app returns to the previous screen.   EX-2: Record could not be saved.   1. Error pops up saying “Reminder was not saved. Please try again.” 2. The app returns to the previous screen. |
| Includes: | - |
| Priority: | Low |
| Frequency of Use: | Low Frequency – Once set up properly, the user would most likely continue using the same medication reminders. The medication reminders would most likely change once every few months and/or when the prescription changes. |
| Business Rules: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

1. Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.4 | | |
| Use Case Name: | Change Record | | |
| Created By: | Zachary Varella Lee Zheyu | Last Updated By: | Ng Chi Hui |
| Date Created: | 27/08/21 | Date Last Updated: | 11/09/2021 |

|  |  |
| --- | --- |
| Actor: | Reminder Database (Supporting) |
| Description: | The system will modify the selected reminder record in the database according to the specifications applied by the user. |
| Trigger: | Triggered by the calling use case ‘Edit Reminder’ when the user taps on the ‘Edit Reminder’ button. |
| Preconditions: | 1. The reminder record must exist in the database.   AND   1. Connection to the database must be established. |
| Postconditions: | 1. The chosen reminder record is updated and saved in the database. |
| Normal Flow: | 1. The calling use case ‘Edit Reminder’ will inform the system when the user taps on the ‘Edit Reminder’ button. 2. The screen will show a menu pop-up containing editable fields for ‘Name’, ‘Description’, ‘Frequency’, and ‘Time’, a cancel button and a ‘Done’ button. 3. Upon selecting a field, a keyboard will appear allowing the user to type the new field information. 4. The user taps the ‘Done’ button. 5. The app uses a SQL query to change the record accordingly. 6. The pop-up closes. 7. The app returns to the previous screen. |
| Alternative Flows: | AF-S3: The user taps the ‘cancel’ button   1. The pop-up closes. 2. The app returns to step 7. |
| Exceptions: | EX-1: Connection to the database is lost.   1. Error pops up saying “An error occurred. Information could not be saved.” 2. The app returns to the previous screen.   EX-2: Record could not be saved.   1. Error pops up saying “Reminder was not saved. Please try again.” 2. The app returns to the previous screen. |
| Includes: | - |
| Priority: | Low |
| Frequency of Use: | Low Frequency – Once set up properly, the user would most likely continue using the same medication reminders. The medication reminders would most likely change once every few months and/or when the prescription changes. |
| Business Rules: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

1. Acknowledge Reminder

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.5 | | |
| Use Case Name: | Acknowledge Reminder | | |
| Created By: | Zachary Varella Lee Zheyu | Last Updated By: | Zachary Varella Lee Zheyu |
| Date Created: | 27/08/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | User (Initiating) |
| Description: | The user acknowledges the reminder notification by tapping on it or swiping it. |
| Trigger: | Triggered by the user tapping on the system reminder. |
| Preconditions: | 1. It is the correct time for the notification |
| Postconditions: | 1. The notification is acknowledged. |
| Normal Flow: | 1. The use case “Push Notification” is called, sending a medicine reminder notification to the user’s device. 2. The user taps on the notification. 3. The app will launch. 4. The app navigates to the Reminder screen for the correct medicine pertaining to the medicine reminder. |
| Alternative Flows: | AF-S2: The user swipes away the notification.   1. The user swipes away the notification by dragging it to the right or to the left. 2. The notification is dismissed. |
| Exceptions: | - |
| Includes: | - |
| Priority: | Medium |
| Frequency of Use: | Medium Frequency – The notification must pop up on a regular schedule. Estimated frequency is once a day. |
| Business Rules: | - |
| Special Requirements: | SR-1: The app should launch and navigate to the reminder screen within 2 seconds. |
| Assumptions: | - |
| Notes and Issues: | - |

1. Push Notification

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.6 | | |
| Use Case Name: | Push Notification | | |
| Created By: | Zachary Varella Lee Zheyu | Last Updated By: | Zachary Varella Lee Zheyu |
| Date Created: | 27/08/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | Reminder Database (Supporting) |
| Description: | The system will send a notification in the background when it is the time for the user’s medicine, as according to the medicine reminder record in the database. |
| Trigger: | The system’s internal reminder handling will call the use case when it is time for the reminder to be pushed. |
| Preconditions: | 1. The reminder record must exist in the database.   AND   1. Background notifications are enabled. |
| Postconditions: | 1. The push notification is sent to the user’s device at the appropriate time. |
| Normal Flow: | 1. The system checks the system clock against the reminder database. 2. When it is the correct time for a medicine reminder, the reminder is activated and sent to the user’s device. 3. The user can use the Use Case “Acknowledge Reminder” to acknowledge and view the Reminder screen. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | Use Case 2.5: “Acknowledge Reminder” |
| Priority: | Medium |
| Frequency of Use: | Medium Frequency – The notification must pop up on a regular schedule. |
| Business Rules: | - |
| Special Requirements: | SR-1: Reminders must not be more than 30 seconds off from the time they are supposed to be sent. |
| Assumptions: | - |
| Notes and Issues: | - |

**3. Cognitive Game Games Case Description**

1. Select Cognitive Game to Play

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.1 | | |
| Use Case Name: | Select Cognitive Game to Play | | |
| Created By: | Ng Chi Hui | Last Updated By: | Ng Chi Hui |
| Date Created: | 27/08/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | User (Initiating) |
| Description: | Upon opening of the games screen, the user selects the desired game they would like to play. |
| Trigger: | Triggered by the user selecting the ‘Cognitive Games’ button on the Home Screen. |
| Preconditions: | 1. The user must be on the cognitive games screen. |
| Postconditions: | 1. The system successfully loads the correct game according to the user’s actions. |
| Normal Flow: | 1. The user taps the ‘Cognitive Games’ button from the main screen. 2. The app will navigate to the ‘Cognitive Games’ screen. 3. The app displays a grid view of the games including the game icon and name of ‘Mental Math’ and ‘Simon Says’. 4. If the user taps on the ‘Mental Maths’ button, the included use case “Mental Math Game” will be called to load the selected game page. 5. If the user taps on the “Simon Says” button, the included use case “Simon Says Game” will be called to load the selected game page. |
| Alternative Flows: | AF-S3: The user taps the ‘Back’ button.   1. The app will return to the ‘Main’ screen. |
| Exceptions: | - |
| Includes: | Use Case 3.2: “Launch Mental Math Game”  Use Case 3.3: “Generate Math Question and Answers”  Use Case 3.4: “Check Answer for Mental Math”  Use Case 3.5: “Launch Simon Says Game”  Use Case 3.6: “Create Pattern”  Use Case 3.7: “Check Pattern for Simon Says” |
| Priority: | High |
| Frequency of Use: | High Frequency – This page would have to be accessed every time the user wants to play a game. |
| Business Rules: | - |
| Special Requirements: | SR-1: The game screen should load within 1 second of tapping the games button. |
| Assumptions: | - |
| Notes and Issues: | - |

1. Launch Mental Math Game

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.2 | | |
| Use Case Name: | Launch Mental Math Game | | |
| Created By: | Ng Chi Hui | Last Updated By: | Zachary Varella Lee Zheyu |
| Date Created: | 27/08/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | - |
| Description: | User successfully starts and enters the game |
| Trigger: | The user presses the ‘Mental Math’ button on the Game Screen. |
| Preconditions: | 1. The user must have selected the ‘Mental Math’ game. |
| Postconditions: | 1. The game is loaded successfully.   AND   1. The user can interact with the game. |
| Normal Flow: | 1. The game is chosen by the calling use case ‘Select Game to Play’. 2. The user may tap the tutorial button to use the extended use case ‘View Mental Math Tutorial’. 3. The user may tap the leaderboard button to use the extended use case ‘View Mental Math Leaderboard’. 4. The user must tap the play button to start the game. 5. The game play screen is displayed and loaded. 6. The system will generate a queue of math questions using the use case ‘Generate Math Questions and Answers’. 7. A timer of 1 minute is started. 8. The app displays the math question from the queue. 9. The user types in the answer. 10. The app checks the answer using the included use case ‘Check Answer for Mental Math’. 11. Steps 7-9 are repeated until the time limit runs out. 12. At the end of the game, the system will display the game statistics. |
| Alternative Flows: | AF-S4: The user exits the game early before the start of the game.   1. The user taps on the back button. 2. The app returns to the ‘Games’ screen.   AF-S6: The user exits the game amid game play.   1. The user taps on back button which exits the game upon confirmation. 2. The app returns to step 4. |
| Exceptions: | - |
| Includes: | Use Case 3.3: “Generate Math Questions and Answers”  Use Case 3.4: “Check Answer for Mental Math”  Use Case 3.5: “View Mental Math Tutorial”  Use Case 3.6: “View Mental Math Leaderboard” |
| Priority: | Medium |
| Frequency of Use: | Medium Frequency – This feature would be accessed whenever the user wants to play the mental math game. |
| Business Rules: | - |
| Special Requirements: | SR-1: The game play screen should load within 10s second of tapping the play button. |
| Assumptions: | - |
| Notes and Issues: | - |

1. Generate Math Questions & Answers

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.3 | | |
| Use Case Name: | Generate Math Questions and Answers | | |
| Created By: | Ng Chi Hui | Last Updated By: | Ng Chi Hui |
| Date Created: | 27/08/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | - |
| Description: | Once the user has tapped the play button from the Mental Math Game page, the system will need to generate a queue of Maths answers and questions randomly. |
| Trigger: | When the user taps the ‘Play’ button, the system will automatically start to generate the math questions. |
| Preconditions: | 1. The user must have clicked on the play button of Mental Math Game.   AND   1. The user must be in the game play screen. |
| Postconditions: | 1. The appropriate questions should be loaded into the app to allow the users to view the questions to play.   AND   1. The correct answer for the respective question will needed to be generated for checking and points tabulation. |
| Normal Flow: | 1. The calling use case ‘Mental Math Game’ will inform the system when the user taps on the ‘Play’ button. 2. The system generates a series of maths questions. 3. The system generates a corresponding series of answers. 4. The system returns the generated questions to the calling use case. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Priority: | High |
| Frequency of Use: | High Frequency |
| Business Rules: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

1. Check Answer for Mental Math

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.4 | | |
| Use Case Name: | Check Answer for Mental Math | | |
| Created By: | Ng Chi Hui | Last Updated By: | Ng Chi Hui |
| Date Created: | 27/8/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | - |
| Description: | Once the user has inputted their answers, the system will check if the answers are right. |
| Trigger: | When the user submits their answer |
| Preconditions: | 1. The user must answer the displayed question in the game. |
| Postconditions: | 1. The system should reflect whether their answer is right or not and attach the points correspondingly. |
| Normal Flow: | 1. The calling use case ‘Mental Math Game’ invokes this use case. 2. The system will use the generated answer to check against the user’s answer. 3. The system returns to the previous use case ‘Mental Math Game’. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Priority: | High |
| Frequency of Use: | High Frequency |
| Business Rules: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

1. View Mental Math Tutorial

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.5 | | |
| Use Case Name: | View Mental Math Tutorial | | |
| Created By: | Zachary Varella Lee Zheyu | Last Updated By: | Zachary Varella Lee Zheyu |
| Date Created: | 27/8/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | - |
| Description: | The app will show a series of images detailing how to play the game. |
| Trigger: | When the user taps the ‘Tutorial’ button on the Mental Math screen. |
| Preconditions: | 1. The user is in the Mental Math game screen. |
| Postconditions: | 1. The tutorial has been exited and the app has returned to the game screen. |
| Normal Flow: | 1. The app shows the first screen explaining the rules of the game. 2. The app shows the second screen showing how to input the answer. 3. The app shows the third screen highlighting the timer. 4. The app shows the fourth screen showing how to use the statistics screen after playing. 5. The app returns to the original Mental Math game screen. |
| Alternative Flows: | AF-S1: The user presses the ‘Back’ button   1. The use case returns to step 5. |
| Exceptions: | - |
| Includes: | - |
| Priority: | Low |
| Frequency of Use: | Low Frequency |
| Business Rules: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

1. View Mental Math Leaderboard

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.6 | | |
| Use Case Name: | View Mental Math Leaderboard | | |
| Created By: | Zachary Varella Lee Zheyu | Last Updated By: | Zachary Varella Lee Zheyu |
| Date Created: | 04/09/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | Leaderboard Database (Supporting) |
| Description: | The app will show a series of images detailing how to play the game. |
| Trigger: | When the user taps the ‘Leaderboard’ button on the Mental Math screen. |
| Preconditions: | 1. The user has Internet connection. |
| Postconditions: | 1. The leaderboard is shown on the app. |
| Normal Flow: | 1. The use case is invoked by the calling use case ‘Mental Math Game’. 2. The app pulls records in order of highest score from the Leaderboard Database. 3. The app pulls the user’s top score from the Leaderboard Database. 4. The app will display the top scores as a list in order of highest score first. 5. The user can press the ‘Back’ button to return to the calling use case. |
| Alternative Flows: | AF-S2: No scores exist in the database.   1. The app will display the text ‘No scores yet!” in place of the list. 2. The use case returns to step 5. |
| Exceptions: | EX-1: The app cannot connect to the Leaderboard Database.   1. An error message is displayed, saying “Unable to connect to the server. Please try again!” 2. The user acknowledges the error message. 3. The application returns to the calling use case screen. |
| Includes: | - |
| Priority: | Low |
| Frequency of Use: | Low Frequency |
| Business Rules: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

1. Launch Simon Says Game

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.7 | | |
| Use Case Name: | Simon Says Game | | |
| Created By: | Lionel Wong Zhi Neng | Last Updated By: | Zachary Varella Lee Zheyu |
| Date Created: | 27/8/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | - |
| Description: | User successfully starts and enters the game |
| Trigger: | When user taps on Simon Says application button at the Game Screen |
| Preconditions: | 1. The user must have selected the ‘Simon Says’ game. |
| Postconditions: | 1. The game is loaded successfully.   AND   1. The user can interact with the game. |
| Normal Flow: | 1. The game is chosen by the calling use case ‘Select Game to Play’. 2. The user may tap the tutorial button to use the extended use case ‘View Simon Says Tutorial’. 3. The user may tap the leaderboard button to use the extended use case ‘View Simon Says Leaderboard’ 4. The user must tap the play button to start the game. 5. The game play screen is displayed and loaded. 6. The system will press boxes on a 3 by 3 grid at random using the use case ‘Create Pattern’. 7. The user may tap on the boxes that follows the pattern that was shown. 8. Every time the user gets the pattern correct, the system will reuse the same pattern and add one more pressed box to the pattern. 9. Steps 6-7 are repeated until the user presses a box in the pattern that does not match the one created by the system. 10. At the end of the game, the system will display the game statistics. |
| Alternative Flows: | AF-S3: The user exits the game early before the start of the game.   1. The user taps on the back button. 2. The app returns to the ‘Games’ screen.   AF-S5: The user exits the game amid game play.   1. The user taps on back button which exits the game upon confirmation. 2. The app returns to step 3. |
| Exceptions: | - |
| Includes: | Use Case 3.8: “Check Pattern for Simon Says”  Use Case 3.9: “Create Pattern”  Use Case 3.10: “View Simon Says Tutorial”  Use Case 3.11: “View Simon Says Leaderboard” |
| Priority: | Medium |
| Frequency of Use: | Medium Frequency – This feature would be accessed whenever the user wants to play the mental math game. |
| Business Rules: | - |
| Special Requirements: | SR-1: The game play screen should load within 10s second of tapping the play button. |
| Assumptions: | - |
| Notes and Issues: | - |

1. Check Pattern for Simon Says

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.8 | | |
| Use Case Name: | Check Pattern for Simon Says | | |
| Created By: | Lionel Wong Zhi Neng | Last Updated By: | Lionel Wong Zhi Neng |
| Date Created: | 27/8/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | - |
| Description: | Once the user has inputted their pattern, the system will check if the patterns match. |
| Trigger: | When the user inputs the pattern answer. |
| Preconditions: | 1. The user must press the boxes to create a pattern in the game. |
| Postconditions: | 1. The system should reflect whether their answer is right or not and continue or end the game correspondingly. |
| Normal Flow: | 1. The calling use case ‘Simon Says Game’ invokes this use case. 2. The system will use the saved pattern to check against the user’s pattern input. 3. The system returns to the previous use case ‘Simon Says Game’. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Priority: | High |
| Frequency of Use: | High Frequency |
| Business Rules: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

1. Create Pattern

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.9 | | |
| Use Case Name: | Create Pattern | | |
| Created By: | Lionel Wong Zhi Neng | Last Updated By: | Lionel Wong Zhi Neng |
| Date Created: | 27/8/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | - |
| Description: | Once the user has tapped the play button from the Simon Says game page, the system will need to press boxes on a 3 by 3 grid at random. |
| Trigger: | When the user taps the ‘Play’ button, the system will automatically start to create the pattern. |
| Preconditions: | 1. The user must have clicked on the play button of Simons Says Game.   AND   1. The user must be in the game play screen. |
| Postconditions: | 1. The pattern should be loaded into the app to allow the users to view the questions to play.   AND   1. The pattern must be saved by the system to check against the user’s pattern. |
| Priority: | High |
| Frequency of Use: | High Frequency |
| Normal Flow: | 1. The calling use case ‘Simon Says Game’ will inform the system when the user taps on the ‘Play’ button. 2. The system generates a sequence of clicked boxes as a pattern. 3. The system returns the generated pattern to the calling use case. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

1. View Simon Says Tutorial

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.10 | | |
| Use Case Name: | View Simon Says Tutorial | | |
| Created By: | Zachary Varella Lee Zheyu | Last Updated By: | Zachary Varella Lee Zheyu |
| Date Created: | 27/8/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | - |
| Description: | The app will show a series of images detailing how to play the game. |
| Trigger: | When the user taps the ‘Tutorial’ button on the Simon Says screen. |
| Preconditions: | 1. The user is in the Simon Says game screen. |
| Postconditions: | 1. The tutorial has been exited and the app has returned to the game screen. |
| Priority: | Low |
| Frequency of Use: | Low Frequency |
| Normal Flow: | 1. The app shows the first screen explaining the rules of the game. 2. The app shows the second screen showing how to input the pattern. 3. The app shows the fourth screen showing how to use the statistics screen after playing. 4. The app returns to the original Mental Math game screen. |
| Alternative Flows: | AF-S1:   1. The user presses the ‘Back’ button 2. The use case returns to step 4. |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

1. View Simon Says Leaderboard

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.11 | | |
| Use Case Name: | View Simon Says Leaderboard | | |
| Created By: | Zachary Varella Lee Zheyu | Last Updated By: | Zachary Varella Lee Zheyu |
| Date Created: | 04/09/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | Leaderboard Database (Supporting) |
| Description: | The app will show a series of images detailing how to play the game. |
| Trigger: | When the user taps the ‘Leaderboard’ button on the Simon Says screen. |
| Preconditions: | 1. The user has Internet connection. |
| Postconditions: | 1. The leaderboard is shown on the app. |
| Priority: | Low |
| Frequency of Use: | Low Frequency |
| Normal Flow: | 1. The use case is invoked by the calling use case ‘Simon Says Game’. 2. The app pulls records in order of highest score from the Leaderboard Database. 3. The app pulls the user’s top score from the Leaderboard Database. 4. The app will display the top scores as a list in order of highest score first. 5. The user can press the ‘Back’ button to return to the calling use case. |
| Alternative Flows: | AF-S2: No scores exist in the database.   1. The app will display the text ‘No scores yet!” in place of the list. 2. The use case returns to step 5. |
| Exceptions: | EX-1: The app cannot connect to the Leaderboard Database.   1. An error message is displayed, saying “Unable to connect to the server. Please try again!” 2. The user acknowledges the error message. 3. The application returns to the calling use case screen. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

**4. Exercise Video Diagram Case Description**

1. Select Exercise Video

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4.1 | | |
| Use Case Name: | Select Exercise Video | | |
| Created By: | Ng Chi Hui | Last Updated By: | Ng Chi Hui |
| Date Created: | 27/8/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | User (Initiating) |
| Description: | Upon opening the exercise screen, the user selects desired exercise video they would like to play. |
| Trigger: | When the user selects an exercise video from the Exercise Videos list. |
| Preconditions: | 1. The user must be on the exercise screen. |
| Postconditions: | 1. The system successfully selects the correct exercise video according to the user’s actions. |
| Priority: | Low |
| Frequency of Use: | Medium Frequency - When the user wants to exercise |
| Normal Flow: | 1. The user is shown a list view of the available exercises. 2. The user will click on their preferred exercise that they would like to watch. 3. The included use case “Load Video” is called to load the selected exercise video. |
| Alternative Flows: | - |
| Exceptions: | - |
| Includes: | Use Case 4.2: “Load Video” |
| Special Requirements: | SR-1: The exercise screen should load within 1 second of tapping the button. |
| Assumptions: | - |
| Notes and Issues: | - |

1. Load Exercise Video

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4.2 | | |
| Use Case Name: | Load Exercise Video | | |
| Created By: | Ng Chi Hui | Last Updated By: | Ng Chi Hui |
| Date Created: | 27/8/21 | Date Last Updated: | 11/09/21 |

|  |  |
| --- | --- |
| Actor: | YouTube API (Supporting) |
| Description: | Once the user has tapped on an Exercise Video card from the Exercise page, the system will load the video. |
| Trigger: | When the system calls the YouTube API to load the video. |
| Preconditions: | 1. The user has selected an exercise to watch. |
| Postconditions: | 1. The selected video should be loaded into the app to allow the users to view the videos.   AND   1. The user can watch, pause and stop the exercise video. |
| Normal Flow: | 1. The use case is invoked by the calling use case ‘Select Exercise Video’ 2. The video is fetched using the YouTube API. 3. The video is loaded on to the user’s screen within the app. 4. Upon successful loading of the video, the video will play automatically. 5. The user presses the ‘back’ button to return to the previous screen. |
| Alternative Flows: | AF-S4: The user exits the video midway.   1. The user taps on the back button. 2. The system closes the video. 3. The app returns to the previous screen. |
| Exceptions: | EX-1: Video cannot be fetched from YouTube API   1. An error notification will appear, saying “Video could not be loaded.” 2. The user acknowledges the notification. 3. The system closes the notification. 4. The app returns to the previous screen. |
| Includes: | - |
| Priority: | Low |
| Frequency of Use: | Medium Frequency |
| Business Rules: | The YouTube API should not have more than 10,000 calls per day among all users. |
| Special Requirements: | SR-1: The exercise video should play within 10 seconds of tapping the play button |
| Assumptions: | - |
| Notes and Issues: | - |