****

**COMSATS University Islamabad (CUI)**

**Project Name**

***By***

**Student Name 1 CIIT/SP09-BCS-xxx/ISB**

**Student Name 2 CIIT/SP09-BCS-xxx/ISB**

***Supervisor*Supervisor Name**

***Co-Supervisor*C0-Supervisor Name**

***Bachelor of Science in Computer Science (20xx-20xx)***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

****

**COMSATS University Islamabad (CUI)**

**Project Name**

**A project presented to**

**COMSATS University Islamabad**

**In partial fulfillment**

**of the requirement for the degree of**

***Bachelor of Science in Computer Science (20xx-20xx)***

**By**

**Student Name 1 CIIT/SP09-BCS-xxx/ISB**

**Student Name 2 CIIT/SP09-BCS-xxx/ISB**

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

|  |  |  |
| --- | --- | --- |
| Student Name1 | Student Name2 | Student Name3 |
| --------------------------- | --------------------------- | --------------------------- |

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (CS) “Project title” was developed by   
**STUDENT 1 NAME (CIIT/FAXX-BCS-000)** and **STUDENT 2 NAME (CIIT/FAXX-BCS-000)** under the supervision of “SUPERVISOR NAME” and co supervisor “CO-SUPERVISOR NAME” and that in (their/his/her) opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences.

---------------------------------------

**Supervisor**

---------------------------------------

**Co-Supervisor**

---------------------------------------

**External Examiner**

---------------------------------------

**Head of Department**

**(Department of Computer Science)**

**Executive Summary**

In public places, there is often a need for monitoring people and different activities going on, which can be referred later for many reasons including security. Appointing humans for this task involves many problems such as increased employee hiring, accuracy problem, trust, no proof for later use, and also the fact that a human can remember things till a certain time limit. Talking about the current security system, they use dumb still cameras with a continuous recording facility irrespective of the fact that any event may happen or not. Moreover, they are usually pointing at a specific user defined location so more than one cameras are required to cover the entire region.

To prevent all these problems from prevailing, the CSCS is developed. It is a surveillance system, which provides solution to many of these problems. It is a stand-alone application which doesn’t require any computer to operate. It monitors different situations using a camera which is able to rotate intelligently based on sensor messages and captures the scene in the form of video or photos later reference as well.

**C**ustomizable **S**urveillance **C**ontrol **S**ystem **(CSCS)** is a surveillance system that can be assigned a sensor type as in our case a heat sensor is used, it works accordingly, rotates the camera upon event detection and perform user defined actions like capturing video and stores them, for the future use.

It is an embedded system consisting of Linux fox kit with embedded a running server application also a camera, USB storage device and a sensor node base station is attached with fox kit. LAN communication is used by user to download the videos and to operate the system manually.

**Acknowledgement**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Dr. Majid Iqbal Khan” and our Co-Supervisor “Mr. Mukhtar Azeem”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

|  |  |  |
| --- | --- | --- |
| Student Name1 | Student Name2 | Student Name3 |
| --------------------------- | --------------------------- | --------------------------- |

**Abbreviations**

|  |  |
| --- | --- |
| **SRS** | Software Requirement Specification |
| **PC** | Personal Computer |
|  |  |
|  |  |
|  |  |

**Table of Contents**

[**1 Introduction** 10](#_Toc520773734)

[1.1 Brief Overview 10](#_Toc520773735)

[1.2 Relevance to Course Modules 10](#_Toc520773736)

[1.3 Project Background 10](#_Toc520773737)

[1.4 Literature Review 10](#_Toc520773738)

[1.5 Analysis from Literature Review 10](#_Toc520773739)

[1.6 Methodology and Software Lifecycle for this Project 10](#_Toc520773740)

[1.6.1 Rationale behind the Selected Methodology 10](#_Toc520773741)

[**2** **Problem Definition** 11](#_Toc520773742)

[2.1 Problem Statement 11](#_Toc520773745)

[2.2 Deliverables and Development Requirements 11](#_Toc520773746)

[2.3 Current Systems 11](#_Toc520773747)

[**3** **Requirement Analysis** 12](#_Toc520773748)

[3.1 Use Cases Diagram(s) 12](#_Toc520773750)

[3.2 Detailed Use Cases 12](#_Toc520773751)

[3.3 Functional Requirements 12](#_Toc520773752)

[3.4 Non-Functional Requirements 12](#_Toc520773753)

[**4 Design and Architecture** 13](#_Toc520773754)

[4.1 System Architecture 13](#_Toc520773759)

[4.2 Data Representation 13](#_Toc520773760)

[4.3 Process Flow/Representation 13](#_Toc520773761)

[4.4 Design Models 13](#_Toc520773762)

[**5** **Implementation** 14](#_Toc520773763)

[5.1 Algorithm 14](#_Toc520773765)

[5.2 External APIs 14](#_Toc520773766)

[5.3 User Interface 14](#_Toc520773767)

[**6 Testing and Evaluation** 15](#_Toc520773768)

[6.1 Manual Testing 15](#_Toc520773771)

[6.1.1 System testing 15](#_Toc520773772)

[6.1.2 Unit Testing 15](#_Toc520773773)

[6.1.3 Functional Testing 16](#_Toc520773774)

[6.1.4 Integration Testing 16](#_Toc520773775)

[6.2 Automated Testing: 17](#_Toc520773776)

[Tools used: 17](#_Toc520773777)

[**7** **Conclusion and Future Work** 18](#_Toc520773778)

[7.1 Conclusion 18](#_Toc520773780)

[7.2 Future Work 18](#_Toc520773781)

[**8** **References** 19](#_Toc520773782)

**List of Figures**

Fig 1.1 Block Diagram 8

Fig 2.1 Use Case Diagram 9

# 

# Introduction

A brain-computer interface provides control of the system to the user with minimal or no physical interaction at all with the system itself, or, gives the user the power of “virtual telekinesis”. In any BCI, the brain can be said to be the primary peripheral with the cortexes as further specialized sub-peripherals. The working of our brain relies on an interconnected neural network, and this network either receives signals from receptors over the entire body and processes these signals, and/or, originates new signals containing information to perform specific actions like walking, sneezing, sleeping, focusing, concentrating or even just thinking. These signals flow in the form of electrical conductions and EEG can detect the inputs, processes, and outputs to this natural neural network, which involve the firing of certain neurons, as spikes or other spatial variations in the graphs, and each type of variation corresponds to a specific action or state of the brain. Depending upon the type of the variation, the BCI can utilize the relevant information and provide function dependent interfacing. A BCI can be a reasonable platform standing on the shoulders of which the epidemic of dopamine infliction and addiction can be fought to normalize and lengthen attention spans.

## Brief Overview

To achieve self-awareness, the above proposed BCI can be used to make any user of the complementary application aware of the fact that he or she has lost focus during the indulgence of this certain task. Technically, this will be achieved by identification of a pattern difference in the Electroencephalograph of the user, which our system will learn by gathering experience and learning from previous examples to build and develop. A state in which the user has lost the focus, reinforcement of attention can be achieved in a multitude of ways.

* We lose our focus doing certain tasks, so if there’s monitor and check on this, we can bring forth a targeted solution
* Amongst all the victims of this mind wandering dilemma, the task of book reading is the most popular one. Almost all readers face this problem, and we can design our system to monitor our users and train them to pay more attention to this task at hand
* For the other population, which did not even get into reading, a more suitable way is to use entertainment in form games.

## Relevance to Course Modules

* Machine Learning
* EEG Analysis /pattern Recognition
* Android Development
* Game design and development
* Artificial Intelligence (Lexical Analysis/Natural Language Processing)

## Project Background

Brain-Computer Interface (BCI), also known as a neural-control interface, a mind-machine interface, etc., is the act of providing humans the access to the functionality, although not impartial, of computers, or any equally programmable and compatible device via the brain as the human input to this interface. Electroencephalography (EEG) is a medical non-invasive electrophysiological monitoring technique used to graph electrical brain waves, with its current usage in the domain of targeted and isolated research pertaining to the neurological and cerebral side of anatomy. EEG appears to be a reasonable approximation towards a starting point into creating a seamless BCI. The fundamental idea is to use this EEG to develop a BCI that can target issues pertaining to human activities. To be more specific, we’re aiming at providing a solution to the epidemic of excessive phone usage, which is one of the fore comings of millennialism, enhancing the attention span to improve focus in daily activities such as reading and others with similar brain involvement. We’ve chosen to divide the entire methodology into two streams; entertainment incentivized training and specialized controlled training, for example by games or any activity with self-perpetuating interest, and, monitored and visualized brain functioning during attentive activities respectively.

## Literature Review

The first portable EEG headset was created in 2010 . After that this technology has been used in many ways for the betterment of humanity.

|  |  |  |
| --- | --- | --- |
| **Application Name** | **Weakness** | **Proposed Project Solution** |
| **TGAM Neurosky:**  an EEG sensor with limited electrodes | * Lacks detailed and informative brain reading EEG and a likeable * Lacks captivating and interesting entry points into using and reusing the system | * A more high-end sensor from another headset manufacturer * Developing a more likeable interface between the headset and the use |
| A Real-time EEG-based BCI System for Attention Recognition in Ubiquitous Environment Li et al.’15 | No such front-end at all | At least a sub-module requiring user to interface with the system. |

## Analysis from Literature Review

The general scheme of the above proposed methodology i.e. entertainment incentivized training and specialized controlled training, can achieve sufficient augmentation in the problem’s situation. Self-awareness is important to fight any problem of similar nature, because the cause and resolve should belong in the same environment for a victor to evolve.

Achieving self-awareness, and/or more ultimately the required effect for better trained and focused cognitive minds and less dopamine addicted individuals, a reasonably equipped BCI is important, given that it may gain as much public interest as required, and that it is seamless and efficient. Now, towards a more practical and specific perspective, previous modular implementations of the entire project can be held as research standpoints to merge top-of-the-line research and popular off-the-shelf production. Li et al.’15 in A Real-time EEG-based BCI System for Attention Recognition in Ubiquitous Environment developed a base module which we can re-implement in a more reasonable way. Using a fine tuned, tech-equipped system and minimalistic, likely to be usable interface is a way of developing a product.

## Methodology and Software Lifecycle for this Project

The methodology chosen for this project Object Oriented Programming.

### Rationale behind the Selected Methodology

As this project is made on unity and android , both of which support OOP .Thus the only methodology that could be chosen was OOP.

# Problem Definition



## Problem Statement

With the advent of millennialism, a decline in attention spans and dopamine infliction epidemic, have arisen problems that are at the core of society’s current situation. Effortless and seamless social networking platforms continue to grow and attract more and newer users towards in becoming part of the victims of social addiction. People spend most of their time scrolling through these applications looking for self-appreciation and self-appraisal, causing habitual time wastage. From a small part of the global community exists a population that use the brain-children of these tech giants for productive business activities and/or workflow collaborations. For others, scrolling through the updates and news of their social circle keeps them captivated and hypnotized. There have been many realizations about bad public interest and the control that the big organizations have over the human mind. And so is the importance of Cognitive Training (CT) using certain BCIs. Therefore, it begs the question, why isn’t a seamless and efficient input-output system, dependent on a BCI, a popularly approved or a heard off utility? Using the brain to provide specific actions or commands to the interface, considering the above mechanism of EEG, contains a vast array of problematic areas in aspects of existential measurability, signal-to-noise ratio, and seamlessness, i.e. EEG, which is held to be the center point of the BCI, lacks in providing clear distinction between inputs.

## Deliverables and Development Requirements

## Current System

* Python: python will be used as a programming language because most of machine learning libraries are in python e.g. keras , TensorFlow etc. .
* Unity: unity is well reputed for game development and works well in coherence with python.
* Iron Python: to create a link between unity and python and run python scripts directly from c# scripts in unity
* Android: mobile app is chosen for the ease of access and as mobile users are more, we can target a bigger percentage of the population.
* Emotive headset: provides better accuracy and wireless headset is easy to use. Also, they are recommended by others in field to be the best out of all the options.

# Requirement Analysis



## Use Cases Diagram(s)https://documents.lucidchart.com/documents/7b79830e-432b-433e-aada-be039d3c39a1/pages/0_0?a=1966&x=22&y=147&w=836&h=726&store=1&accept=image%2F*&auth=LCA%209eb5da79c522693a1260b758912a50a872ff2922-ts%3D1570112907

## Figure 1 Use case diagram for Account Handling module

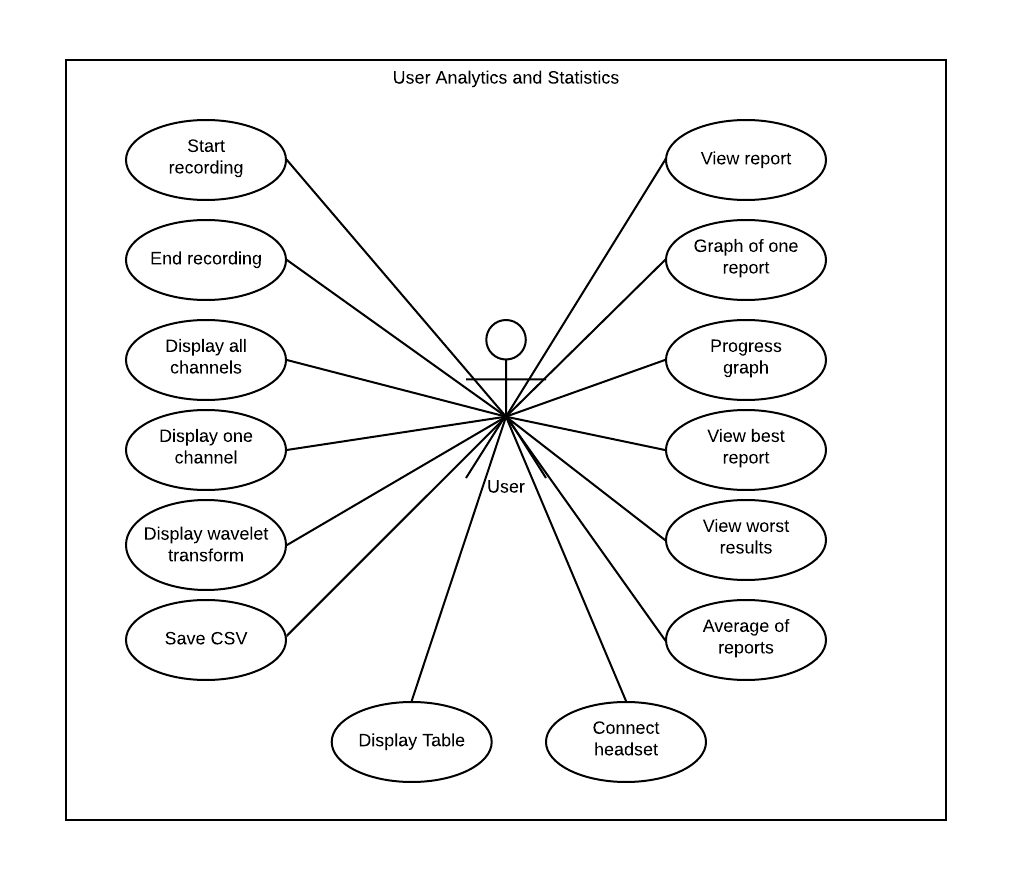


Figure 2 Use case diagram for User Analytics and Statistics

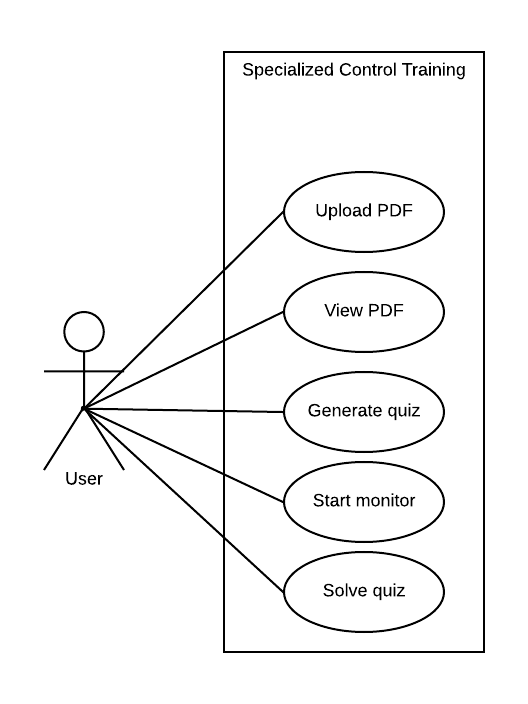


Figure 3 Use case diagram for Specialized Control Training

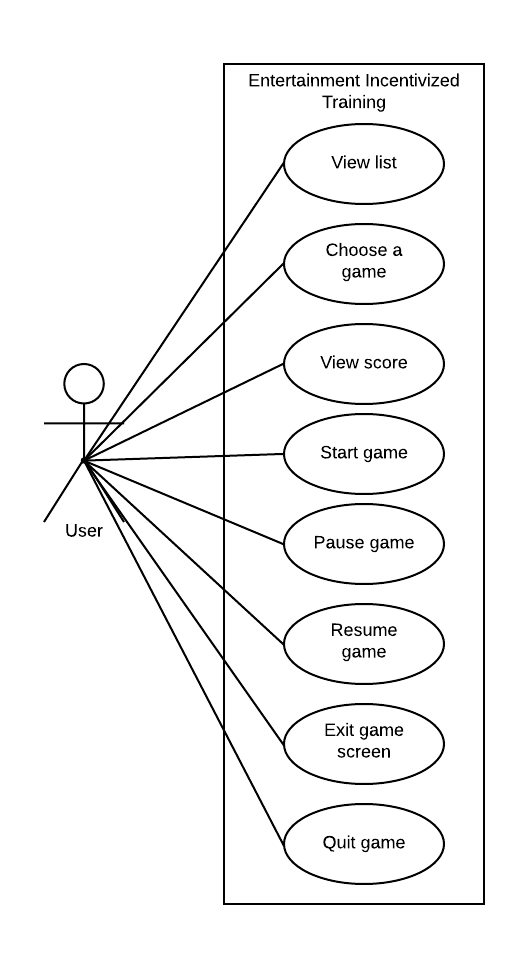


Figure 4 Use case diagram for Entertainment Incentivized Training

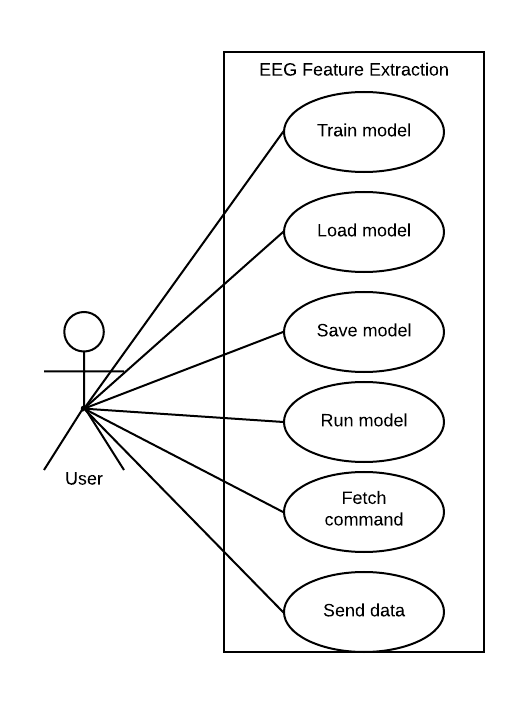


Figure 5 Use case diagram for EEG Feature Extraction

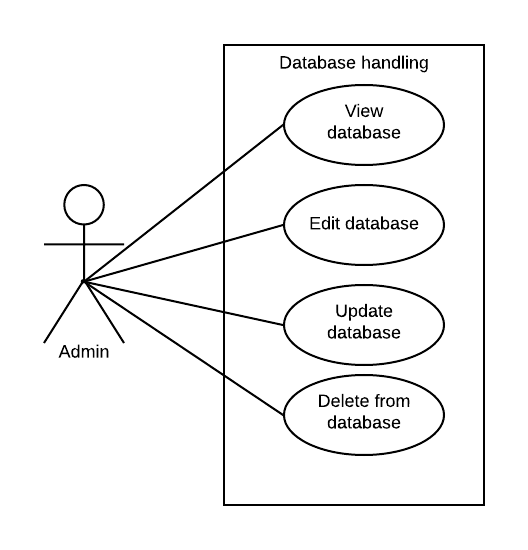


Figure 6 Use case diagram for Database Handling

## Detailed Use Case

## Module 1: Account Handling

**Table 1**

|  |  |
| --- | --- |
| Use Case ID: | UC-1 |
| Use Case Name: | Register Account |
| Actors | Primary actor : user |
| Description: | This registers the user account in the database so that all the data can be placed at one place regarding that user and to ensure privacy of users. |
| Trigger: | User/primary actor initiates this usecase to register his/her account. This use case will be triggered by a button on the first page at the bottom right which will take the user to a form containing empty fields to fill in their information |
| Preconditions: | As registration is one of the first tasks when the application loads so it doesn’t have preconditions other than the application data must be installed and a working internet connection should be provided. |
| Postconditions: | Success: In this case the account is registered and now user can start using the app and find all his data in his account.  Failure: In this case the account will not be registered am user will be notified. |
| Normal Flow: | * User has to enter his/her name * Enter user name * Enter password * Click sign-up * Wait for the confirmation notification |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Username exists * Database connection failed |
| Business Rules | N/A |
| Assumptions: | * Account does not already exist |

**Table 2**

|  |  |
| --- | --- |
| Use Case ID: | UC-2 |
| Use Case Name: | Log-in |
| Actors | Primary actor : user |
| Description: | This logs-in the user account , fetches the account data from the database which can be viewed and edited |
| Trigger: | User/primary actor initiates this usecase to log-in his/her previously registered account. This use case will be triggered by a button on the first page at the center which will take the user to a form containing empty fields to fill in their username and password. |
| Preconditions: | * Registered account * Stable Internet connection * Database connection |
| Postconditions: | Success: In this case the account is logged-in and now user can start using the app.  Failure: In this case the account will not log-in and the user will be notified. |
| Normal Flow: | * User has to enter his/her name * Enter user name * Enter password * Click sign-in * Wait for the confirmation notification |
| Alternative Flows: | N/A |
| Exceptions: | * Unstable internet connection * Username wrong * Password wrong * Database connection failed |
| Business Rules | N/A |
| Assumptions: | * Account is registered in the database * Password provided against the username is correct |

**Table 3**

|  |  |
| --- | --- |
| Use Case ID: | UC-3 |
| Use Case Name: | Log-out |
| Actors | Primary actor : user |
| Description: | This logs-out the user account so it cannot be accessed by someone else . |
| Trigger: | User/primary actor initiates this usecase to log-out his/her previously logged-in account. This use case will be triggered by a button on the home page at the top right corner which will take the user back to the log-in page. |
| Preconditions: | * Logged-in account * Stable Internet connection |
| Postconditions: | Success: in this case the account will be logged-out  Failure: In this case the account will not be logged-out and the user will be notified. |
| Normal Flow: | * User has to click on the profile picture on the top right * Click then button sign-out * Wait for the confirmation notification |
| Alternative Flows: | * Click settings * Click log-out at the bottom of the drop-down menu |
| Exceptions: | * Internet connection unstable |
| Business Rules | N/A |
| Assumptions: | * Account signed-in |

**Table 4**

|  |  |
| --- | --- |
| Use Case ID: | UC-4 |
| Use Case Name: | View personal information |
| Actors | Primary actor : user |
| Description: | This opens the settings page which has all the information the user entered about themselves e.g profile picture, name etc |
| Trigger: | User/primary actor initiates this usecase to view his/her previously entered information . This use case will be triggered by a button on the home page which will take the user to the settings page. |
| Preconditions: | * Logged-in account * Stable Internet connection |
| Postconditions: | Success: In this case the information is displayed on the settings page .  Failure: In this case the settings page will not open and user will be notified. |
| Normal Flow: | * Log-in account * Click on profile pic at the top right * Click on view timeline |
| Alternative Flows: | * Click settings * Click about me in the drop-down menu |
| Exceptions: | * Internet connection unstable |
| Business Rules | N/A |
| Assumptions: | * Account information was entered when account was registered |

**Table 5**

|  |  |
| --- | --- |
| Use Case ID: | UC-5 |
| Use Case Name: | Edit information |
| Actors | Primary actor : user |
| Description: | This opens a small window with input fields to edit the previous information |
| Trigger: | User/primary actor initiates this usecase to edit his/her previously given information . This use case will be triggered by a button on the settings page which will take the user to a small window from where he can change his personal information |
| Preconditions: | * Logged-in account * Stable Internet connection |
| Postconditions: | Success: In this case the information is edited  Failure: In this case the information will not be edited and user will be notified. |
| Normal Flow: | * Log-in account * Click profile pic * Click view timeline * Click button edit beside the information that needs to be edited e.g. name etc. * Enter new information * Click done * Wait for notification for confirmation |
| Alternative Flows: | * Log-in account * Open settings * Click about me * Click button edit beside the information that needs to be edited e.g. name etc. * Enter new information * Click done * Wait for notification for confirmation |
| Exceptions: | * Internet connection unstable * Database connection failed |
| Business Rules | N/A |
| Assumptions: | * Account logged-in |

**Table 6**

|  |  |
| --- | --- |
| Use Case ID: | UC-6 |
| Use Case Name: | Recover password |
| Actors | Primary actor : user |
| Description: | This opens the recover password page and asks the user if they wan to reset the password. |
| Trigger: | User/primary actor initiates this usecase to reset his/her password. This use case will be triggered by a button on the log-in page which will take the user to the recover password page from where he can reset the password which will be sent to the recovery email . |
| Preconditions: | * Stable Internet connection |
| Postconditions: | Success: In this case the password is reset and sent to the recovery email.  Failure: In this case the password will not reset |
| Normal Flow: | * Click forgot password button * Get the new password from recovery email * Sign-in with new password |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Account not registered * Recovery email not found |
| Business Rules | N/A |
| Assumptions: | * Account registered * Recovery email provided * Recovery email exists |

**Table 7**

|  |  |
| --- | --- |
| Use Case ID: | UC-7 |
| Use Case Name: | Password-check |
| Actors | System triggers this usecase . |
| Description: | This checks if the password is correct then the user logs-in. |
| Trigger: | This is in initiated when the user logs-in. This use case will be triggered by the sign-in use case which will check the user-password . |
| Preconditions: | * Log-in * Stable Internet connection |
| Postconditions: | Success: password is right. Account is opened  Failure: In this case the password is wrong and generates a notification |
| Normal Flow: | * Enter email * Enter password * Click sign-in * Wait for confirmation notification |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Password wrong |
| Business Rules | N/A |
| Assumptions: | * Account registered * Password correct |

**Table 8**

|  |  |
| --- | --- |
| Use Case ID: | UC-8 |
| Use Case Name: | Log-in(admin) |
| Actors | Primary actor : user |
| Description: | This usecase starts when email and password are entered to log in to the admin account. |
| Trigger: | admin actor initiates this usecase to open their account. This use case will be triggered by a button log-In. |
| Preconditions: | * Stable Internet connection * Headset connected * Registered account * Database connection |
| Postconditions: | Success: In this case the account opens and the admin can start using it.  Failure: In this case the monitor will not start. |
| Normal Flow: | * Enter name * Enter password * Click sign-in |
| Alternative Flows: | N/A |
| Exceptions: | * Unstable internet connection * Account not registered * Database connection failed * Password doesn’t match |
| Business Rules | N/A |
| Assumptions: | * Account registered |

**Table 9**

|  |  |
| --- | --- |
| Use Case ID: | UC-9 |
| Use Case Name: | Already-exists-check(admin) |
| Actors | System |
| Description: | This usecase starts when email and password are entered to sign-up . |
| Trigger: | System initiates this usecase to check if the username exists or not . This use case will be triggered by a button log-In. |
| Preconditions: | * Stable Internet connection * Database connection |
| Postconditions: | Success: In this case the account is registered  Failure: In this case the account will not be registered and a notification will be generated |
| Normal Flow: | * Enter name * Enter password * Click sign-in |
| Alternative Flows: | N/A |
| Exceptions: | * Unstable internet connection * Username already exists * Database connection failed |
| Business Rules | N/A |
| Assumptions: | * Username does not already exist |

#### Module 2: User Analytics and Statistics

**Table 20**

|  |  |
| --- | --- |
| Use Case ID: | UC-10 |
| Use Case Name: | View-report |
| Actors | Primary actor : user |
| Description: | This opens the weekly progress reports generated of the user |
| Trigger: | User/primary actor initiates this usecase to view his/her previously generated reports. This use case will be triggered by a button on the home page which will take the user to the reports page from where he can choose from a lost of reports |
| Preconditions: | * Logged-in account * Stable Internet connection * Reports present to view |
| Postconditions: | Success: In this case the chosen report is displayed.  Failure: In this case the report will not open |
| Normal Flow: | * Log-in account * Play a game or do the reading activity * Open user-analytics tab * Choose a report * Click button view report |
| Alternative Flows: | * Log-in account * Open user-analytics tab * Choose a report * Click button view report |
| Exceptions: | * Internet connection unstable * Report deleted from the database * Database connection failed |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Reports previously generated and saved |

**Table 31**

|  |  |
| --- | --- |
| Use Case ID: | UC-11 |
| Use Case Name: | Graph of one report |
| Actors | Primary actor : user |
| Description: | This usecase displays the graph of one week result. |
| Trigger: | User/primary actor initiates this usecase to view his/her previously generated reports in the form of graph. |
| Preconditions: | * Logged-in account * Stable Internet connection * Previously generated reports |
| Postconditions: | Success: In this case the chosen report is displayed in graphical form  Failure: In this case the report will not open. |
| Normal Flow: | * Log-in account * Play a game or do the reading activity * Open user-analytics tab * Choose a report * Click button view report * Click the graph tab at the bottom of the screen * Wait for confirmation notification. |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Database connection failure * Report deleted from the database |
| Business Rules | N/A |
| Assumptions: | * Logged-in account * Reports previously generated to view |

**Table 42**

|  |  |
| --- | --- |
| Use Case ID: | UC-12 |
| Use Case Name: | Progress graph |
| Actors | Primary actor : user |
| Description: | This opens the graph of all the previous reports to shoe the progress. |
| Trigger: | User/primary actor initiates this usecase to view his/her progress over the time . This use case will be triggered by a button on the home page which will take the user to the reports page and display the progress report. |
| Preconditions: | * Logged-in account * Stable Internet connection * Previously generated reports. |
| Postconditions: | Success: In this case the progress report is displayed.  Failure: In this case the report will not open. |
| Normal Flow: | * Log-in account * Play a game or do the reading activity * Open user-analytics tab * Click button progress report * Wait for confirmation notification. |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Report deleted from the database * No reports present to display the progress report * Database connection failed |
| Business Rules | N/A |
| Assumptions: | * Logged-in account * Previously generated reports |

**Table 53**

|  |  |
| --- | --- |
| Use Case ID: | UC-13 |
| Use Case Name: | View best report |
| Actors | Primary actor : user |
| Description: | This finds the best report out of all present reports and displays that in table format. |
| Trigger: | User/primary actor initiates this usecase to view his/her best out of previously generated reports. This use case will be triggered by a button on the home page which will take the user to the best report in tabular form. |
| Preconditions: | * Logged-in account * Stable Internet connection * Previously generated reports |
| Postconditions: | Success: In this case the best report is displayed.  Failure: In this case the report will not open. |
| Normal Flow: | * Log-in account * Play a game or do the reading activity * Open user-analytics tab * Click button view best report * Wait for confirmation notification. |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * No reports present * Database connection failed |
| Business Rules | N/A |
| Assumptions: | * Logged-in account * Previously generated reports |

**Table 64**

|  |  |
| --- | --- |
| Use Case ID: | UC-14 |
| Use Case Name: | View worst result |
| Actors | Primary actor : user |
| Description: | This opens the worst result out of all the previously generated reports |
| Trigger: | User/primary actor initiates this usecase to view his/her previously generated worst report. This use case will be triggered by a button on the home page which will take the user to the reports page from where he will press the worst report button. |
| Preconditions: | * Logged-in account * Stable Internet connection * Previously generated reports |
| Postconditions: | Success: In this case the worst report is displayed.  Failure: In this case the report will not open |
| Normal Flow: | * Log-in account * Play a game or do the reading activity * Open user-analytics tab * Click button view worst report * Wait for confirmation notification. |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * No reports present * Database connection failed |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Reports previously generated |

**Table 75**

|  |  |
| --- | --- |
| Use Case ID: | UC-15 |
| Use Case Name: | Average of reports |
| Actors | Primary actor : user |
| Description: | This opens the average of weekly reports generated of the user |
| Trigger: | User/primary actor initiates this usecase to view his/her average of all the previously generated reports. This use case will be triggered by a button on the home page which will take the user to the reports page from where he will press the average of reports button |
| Preconditions: | * Logged-in account * Stable Internet connection * Previously generated reports |
| Postconditions: | Success: In this case the average of previously generated reports is displayed.  Failure: In this case the report will not open and user will be notified. |
| Normal Flow: | * Log-in account * Play a game or do the reading activity * Open user-analytics tab * Click button view worst report * Wait for confirmation notification. |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * No reports present * Database connection failed |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Reports previously generated |

**Table 86**

|  |  |
| --- | --- |
| Use Case ID: | UC-16 |
| Use Case Name: | Start recording |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button record is pressed from recording page . |
| Trigger: | User/primary actor initiates this usecase to record his/her brainwaves in the csv format . This use case will be triggered by a button record on the recording page. |
| Preconditions: | * Logged-in account * Stable Internet connection * Headset integration page opened * Headset connected |
| Postconditions: | Success:  Failure: |
| Normal Flow: | * Log-in account * Click button record on the home page * Click start recording on recording-page |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Headset not connected |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Recording page opened |

**Table 97**

|  |  |
| --- | --- |
| Use Case ID: | UC-17 |
| Use Case Name: | End recording |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button end is pressed from headset integration page is |
| Trigger: | User/primary actor initiates this usecase to stop recording his/her brainwaves. This use case will be triggered by a button end on the headset integration page. |
| Preconditions: | * Logged-in account * Stable Internet connection * Headset integration page opened * Headset connected |
| Postconditions: | Success: In this case the brainwaves will stop recording  Failure: In this case the brainwaves will not stop recording. |
| Normal Flow: | * Log-in account * Click button record on the home page * Click start recording on recording-page * Click on end recording |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Headset not connected |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Recording page opened * Recording started |

**Table 108**

|  |  |
| --- | --- |
| Use Case ID: | UC-18 |
| Use Case Name: | Display all channels |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button display channels is pressed from headset integration page. |
| Trigger: | User/primary actor initiates this usecase to view his/her brainwaves in the raw form. This use case will be triggered by a button display all channels on the headset integration page |
| Preconditions: | * Logged-in account * Stable Internet connection * Headset integration page opened * Headset connected * Previous recordings present |
| Postconditions: | Success: In this case the brainwaves will be displayed in the raw form  Failure: In this case the the brainwaves will not be displayed |
| Normal Flow: | * Log-in account * Choose a previous report * Click on display all channels |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Headset not connected * Previous reports not present |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Recording page opened * Reports present to display |

**Table 119**

|  |  |
| --- | --- |
| Use Case ID: | UC-19 |
| Use Case Name: | Display one channel |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button display channel is pressed from headset integration page. |
| Trigger: | User/primary actor initiates this usecase to view his/her brainwaves in the raw form from one channel. This use case will be triggered by a button present under all channels on the headset integration page. |
| Preconditions: | * Logged-in account * Stable Internet connection * Headset integration page opened * Headset connected |
| Postconditions: | Success: In this case the brainwaves will be displayed in the raw form. From the single channel that the user chose  Failure: In this case the the brainwaves will not be displayed |
| Normal Flow: | * Log-in account * Choose a report * Click display one channel |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Headset not connected * No reports present to display |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Recording page opened * Reports present |

**Table 20**

|  |  |
| --- | --- |
| Use Case ID: | UC-20 |
| Use Case Name: | Display wavelet transform |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button wavelet-transform is pressed from headset integration page |
| Trigger: | User/primary actor initiates this usecase to view his/her brainwaves after the application of wavelet transform on them. This use case will be triggered by a button present under each channel on the headset integration page. |
| Preconditions: | * Logged-in account * Stable Internet connection * Headset integration page opened * Headset connected |
| Postconditions: | Success: In this case the brainwaves will be displayed in the wavelet transform form.  Failure: In this case the the brainwaves will not be displayed |
| Normal Flow: | * Log-in account * Choose report * Click on wavelet transform |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Headset not connected * No reports present to display |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Recording page opened * Reports present |

**Table 21**

|  |  |
| --- | --- |
| Use Case ID: | UC-21 |
| Use Case Name: | Save csv |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button save is pressed from headset integration page |
| Trigger: | User/primary actor initiates this usecase to save his/her brainwaves in the table/csv form. This use case will be triggered by a button save on the headset integration page. |
| Preconditions: | * Logged-in account * Stable Internet connection * Headset integration page opened * Headset connected |
| Postconditions: | Success: In this case the brainwaves will be saved in the table form  Failure: In this case the the brainwaves will not be saved |
| Normal Flow: | * Log-in account * Click button record on the home page * Click start recording on recording-page * Click on end recording * Click on save |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Headset not connected |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Recording page opened |

**Table 22**

|  |  |
| --- | --- |
| Use Case ID: | UC-22 |
| Use Case Name: | Display table |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button display table is pressed from headset integration page. |
| Trigger: | User/primary actor initiates this usecase to view his/her brainwaves in the table form. This use case will be triggered by a button display table on the headset integration page. |
| Preconditions: | * Logged-in account * Stable Internet connection * Headset integration page opened * Headset connected |
| Postconditions: | Success: In this case the brainwaves will be displayed in the table form  Failure: In this case the the brainwaves will not be displayed |
| Normal Flow: | * Log-in account * Choose report * Click display table |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Headset not connected * No reports present |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Recording page opened * Reports present |

**Table 23**

|  |  |
| --- | --- |
| Use Case ID: | UC-23 |
| Use Case Name: | Connect headset |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button connect is pressed from the focus re-enforcement page |
| Trigger: | User/primary actor initiates this usecase to connect the headset. This use case will be triggered by a button connect on the focus reinforcement page . |
| Preconditions: | * Logged-in account * Stable Internet connection * Focus re-enforcement page opened |
| Postconditions: | Success: In this case the book headset will be connected.  Failure: In this case the headset will not be connected. |
| Normal Flow: | * Log-in account * Open user-analytics page * Click connect headset |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Headset not in range * Headset not charged |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Reading exercise page opened * Pdf’s present to read |

#### Module 3: Specialized Control Training

**Table 24**

|  |  |
| --- | --- |
| Use Case ID: | UC-24 |
| Use Case Name: | Upload pdf |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button upload is pressed. |
| Trigger: | User/primary actor initiates this usecase to upload a book . This use case will be triggered by a button upload on the focus reinforcement page . |
| Preconditions: | * Logged-in account * Stable Internet connection * Headset connected * Focus re-enforcement page opened |
| Postconditions: | Success: In this case the book chosen will be uploaded.  Failure: In this case the book will not be uploaded. |
| Normal Flow: | * Log-in account * Choose reading exercise button * Click on upload button * Choose a pdf * Click open |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Headset not connected * No books present |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Reading exercise page opened * Pdf’s present to read |

**Table 25**

|  |  |
| --- | --- |
| Use Case ID: | UC-25 |
| Use Case Name: | View pdf |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button view is pressed on the focus re-enforcement page |
| Trigger: | User/primary actor initiates this usecase to view the pdf he uploaded. This use case will be triggered by a button view on the focus reinforcement page .to view the pdf |
| Preconditions: | * Logged-in account * Stable Internet connection * Headset connected * Focus re-enforcement page opened |
| Postconditions: | Success: In this case the book chosen will be displayed  Failure: In this case the book will not be displayed |
| Normal Flow: | * Log-in account * Choose reading exercise button * Click on upload button * Choose a pdf * Click open * Choose one pdf from uploaded pdf * Click view |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Headset not connected * No books present |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Reading exercise page opened * Pdf’s present to read |

**Table 26**

|  |  |
| --- | --- |
| Use Case ID: | UC-26 |
| Use Case Name: | Generate quiz |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button take quiz is pressed and the system then generates a quiz from the pdf user was reading. |
| Trigger: | User/primary actor initiates this usecase to take a quiz. This use case will be triggered by a button take quiz on the focus reinforcement page |
| Preconditions: | * Logged-in account * Stable Internet connection * Headset connected * Focus re-enforcement page opened * Pdf opened |
| Postconditions: | Success: In this case a quiz will be generated.  Failure: In this case a quiz will be generated. |
| Normal Flow: | * Log-in account * Choose reading exercise button * Click on upload button * Choose a pdf * Click open * Click generate quiz |
| Alternative Flows: | This usecase can be started by the system when the focus level drops below threshold |
| Exceptions: | * Internet connection unstable * Headset not connected * No books present * Book not opened |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Reading exercise page opened * Pdf’s present to read * Book opened |

**Table 27**

|  |  |
| --- | --- |
| Use Case ID: | UC-27 |
| Use Case Name: | Start monitor |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button record is pressed from reading exercise page is |
| Trigger: | User/primary actor initiates this usecase to record his/her brainwaves in the csv format . This use case will be triggered by a button record on the reading exercise page. |
| Preconditions: | * Logged-in account * Stable Internet connection * Headset integration page opened * Headset connected |
| Postconditions: | Success: In this case the brainwaves will start recording  Failure: In this case the the brainwaves will not record |
| Normal Flow: | * Log-in account * Choose reading exercise button * Click on upload button * Choose a pdf * Click open * Click start monitor |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Headset not connected * No books present |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Reading exercise page opened * Pdf’s present to read |

**Table 28**

|  |  |
| --- | --- |
| Use Case ID: | UC-28 |
| Use Case Name: | Solve quiz |
| Actors | Primary actor : user |
| Description: | This usecase starts when the solve is pressed from the pop-up window that appears after generating the quiz |
| Trigger: | User/primary actor initiates this usecase to solve the generated quiz. This use case will be triggered by a button solve on the pop-up window. |
| Preconditions: | * Logged-in account * Stable Internet connection * Headset connected * Focus re-enforcement page opened * Uploaded pdf * Opened pdf * Generated quiz |
| Postconditions: | Success: In this case the generated quiz will be opened.  Failure: In this case the quiz will not be opened. |
| Normal Flow: | * Log-in account * Choose reading exercise button * Click on upload button * Choose a pdf * Click open * Click generate quiz * Click solve quiz |
| Alternative Flows: | Click solve quiz when prompted by the system to take quiz |
| Exceptions: | * Internet connection unstable * Headset not connected * No books present |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Reading exercise page opened * Pdf’s present to read |

**Module 4: Entertainment Incentivized Training**

**Table 29**

|  |  |
| --- | --- |
| Use Case ID: | UC-29 |
| Use Case Name: | View list |
| Actors | Primary actor : user |
| Description: | This opens the list of games to choose from. |
| Trigger: | User/primary actor initiates this usecase to view the list o games to play. This use case will be triggered by a button on the game page where a list of all the games will be present . |
| Preconditions: | * Logged-in account * Stable Internet connection |
| Postconditions: | Success: In this case a list of all games is displayed  Failure: In this case the page will not load and following can be the cause |
| Normal Flow: | * Log-in account * Click button games on the home page |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable |
| Business Rules | N/A |
| Assumptions: | * Logged-in account |

**Table 30**

|  |  |
| --- | --- |
| Use Case ID: | UC-30 |
| Use Case Name: | Choose a game |
| Actors | Primary actor : user |
| Description: | This opens the chosen game and you can start playing. |
| Trigger: | User/primary actor initiates this usecase to play the game of his/her choice. This use case will be triggered by a play button on the game page under the game preview picture from the list of all the games on the game page. |
| Preconditions: | * Logged-in account * Stable Internet connection |
| Postconditions: | Success: In this case the game starts and user can start playing..  Failure: In this case the game will not load. |
| Normal Flow: | * Log-in account * Click button games on the home page * Choose a game |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable |
| Business Rules | N/A |
| Assumptions: | * Log-in account |

**Table 31**

|  |  |
| --- | --- |
| Use Case ID: | UC-31 |
| Use Case Name: | View score |
| Actors | Primary actor : user |
| Description: | This opens the score of the previous game . |
| Trigger: | User/primary actor initiates this usecase to view the score of previous game . This use case will be triggered by a button inside the game where score of previous game will be present. |
| Preconditions: | * Logged-in account * Stable Internet connection * Game played before to generate score |
| Postconditions: | Success: In this case the score of previous try will be displayed  Failure: In this case the score will not be displayed |
| Normal Flow: | * Log-in account * Click button games on the home page * Choose a game * Click view score on the game’s home page |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * No scores to view * Database connection failed |
| Business Rules | N/A |
| Assumptions: | * Logged-in account * Game’s home page opened * Scores of previous games present. |

**Table 32**

|  |  |
| --- | --- |
| Use Case ID: | UC-32 |
| Use Case Name: | Start game |
| Actors | Primary actor : user |
| Description: | This usecase starts the game which was chosen. |
| Trigger: | User/primary actor initiates this usecase to play the game. This use case will be triggered by a button on the game menu . |
| Preconditions: | * Logged-in account * Stable Internet connection |
| Postconditions: | Success: In this case the game starts and you find yourself in the level .  Failure: In this case the level will not load. |
| Normal Flow: | * Log-in account * Click button games on the home page * Choose a game * Click view score on the game’s home page * Click start game button |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Game started |

**Table 33**

|  |  |
| --- | --- |
| Use Case ID: | UC-33 |
| Use Case Name: | Pause game |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button pause is pressed while the game is playing. |
| Trigger: | User/primary actor initiates this usecase to pause the game. This use case will be triggered by the button “pause” at the top right of the screen while the game is being played. |
| Preconditions: | * Logged-in account * Stable Internet connection * Game being played |
| Postconditions: | Success: In this case the game will pause in whatever state it is  Failure: In this case the level will not pause. |
| Normal Flow: | * Log-in account * Click button games on the home page * Choose a game * Click view score on the game’s home page * Click start game button * Click pause button |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Game started |

**Table 34**

|  |  |
| --- | --- |
| Use Case ID: | UC-34 |
| Use Case Name: | Resume game |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button “Resume” is pressed while the game is paused. |
| Trigger: | User/primary actor initiates this usecase to restart the game from the point he/she paused it. This use case will be triggered by a button “Resume” at the center of the page while the game is paused. |
| Preconditions: | * Logged-in account * Stable Internet connection * Game being played * Game paused state |
| Postconditions: | Success: In this case the game will resume in whatever state it was before being paused.  Failure: In this case the level will not resume from the paused state . |
| Normal Flow: | * Log-in account * Click button games on the home page * Choose a game * Click view score on the game’s home page * Click start game button * Click pause button * Click resume button |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Game started * Game paused |

**Table 35**

|  |  |
| --- | --- |
| Use Case ID: | UC-35 |
| Use Case Name: | Exit game screen |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button “back” is pressed while the game is playing. |
| Trigger: | User/primary actor initiates this usecase to EXIT the gamescreen. This use case will be triggered by a button “back” while the game is being played |
| Preconditions: | * Logged-in account * Stable Internet connection * Game being played |
| Postconditions: | Success: In this case the game will EXIT to the game menu in whatever state it is  Failure: In this case the gamescreen will not exit. |
| Normal Flow: | * Log-in account * Click button games on the home page * Choose a game * Click view score on the game’s home page * Click start game button * Click back button |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Game started |

**Table 36**

|  |  |
| --- | --- |
| Use Case ID: | UC-36 |
| Use Case Name: | Quit game |
| Actors | Primary actor : user |
| Description: | This usecase starts when the button quit from the game menu is pressed. |
| Trigger: | User/primary actor initiates this usecase to quit the game and go back to the gamesList to choose another. This use case will be triggered by a button quit on the game menu. |
| Preconditions: | * Logged-in account * Stable Internet connection * Game menu opened |
| Postconditions: | Success: In this case the user will be taken back to the gamePage  Failure: In this case the gamePage will not be opened |
| Normal Flow: | * Log-in account * Click button games on the home page * Choose a game * Click quit button |
| Alternative Flows: | * Log-in account * Click button games on the home page * Choose a game * Click start game button * Click back button * Click quit button |
| Exceptions: | * Internet connection unstable |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Game started |

#### Module 5: EEG Feature Extraction

**Table 37**

|  |  |
| --- | --- |
| Use Case ID: | UC-37 |
| Use Case Name: | Train model |
| Actors | System |
| Description: | This usecase starts this usecase to tarin the moel with the data recorded during the game by the user. |
| Trigger: | System initiates this usecase to train the model on the data provided by the user while the user is playing any game. It is triggered byt the place of user at a specific part of the game. |
| Preconditions: | * Logged-in account * Stable Internet connection * Game started |
| Postconditions: | Success: In this case the system will train the model successfully.  Failure: In this case the training process will somehow be interrupted. |
| Normal Flow: | * Log-in account * Click button games on the home page * Choose a game |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Training process interrupted * Game not started |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Game started |

**Table 38**

|  |  |
| --- | --- |
| Use Case ID: | UC-38 |
| Use Case Name: | Load model |
| Actors | System |
| Description: | This usecase starts this use case to load a pre-trained model to use . |
| Trigger: | System initiates this usecase to load a pre-trained model while the game is being played to classify the commands using the data sent by the user. This is triggered by the progression of game to a specific point . |
| Preconditions: | * Logged-in account * Stable Internet connection * Game started |
| Postconditions: | Success: In this case the system will load the model successfully.  Failure: In this case the loading process will somehow be interrupted. |
| Normal Flow: | * Log-in account * Click button games on the home page * Choose a game * Start game |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Training process interrupted * Game not started |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Game started |

**Table 39**

|  |  |
| --- | --- |
| Use Case ID: | UC-39 |
| Use Case Name: | Save model |
| Actors | System |
| Description: | This usecase starts after the model has been trained . The trained model is then saved to be used later. |
| Trigger: | System initiates this usecase to save the model after it has been trained on the data provided by the user while the user is playing any game. This usecase is triggered after the usecase train model. |
| Preconditions: | * Logged-in account * Stable Internet connection * Game started * Model trained |
| Postconditions: | Success: In this case the system will save the trained model successfully.  Failure: In this case the saving process will somehow be interrupted. |
| Normal Flow: | * Log-in account * Click button games on the home page * Choose a game * Train model |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Training process interrupted * Game not started |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Game started |

**Table 40**

|  |  |
| --- | --- |
| Use Case ID: | UC-40 |
| Use Case Name: | Run model |
| Actors | System |
| Description: | This usecase starts when the user runs a pre-trained model to classify the commands while the game is being played. |
| Trigger: | System initiates this usecase to run the pre-trained model on the data provided by the user while the user is playing any game to classify the commands. This is being checked in the loop while the game is being played. |
| Preconditions: | * Logged-in account * Stable Internet connection * Game started |
| Postconditions: | Success: In this case the system will run the model successfully.  Failure: In this case the running process will somehow be interrupted. |
| Normal Flow: | * Log-in account * Click button games on the home page * Choose a game |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Training process interrupted * Game not started |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Game started |

**Table 41**

|  |  |
| --- | --- |
| Use Case ID: | UC-41 |
| Use Case Name: | Fetch command |
| Actors | System |
| Description: | This usecase starts after the run model usecase to fetch the answer generated by the model. |
| Trigger: | System initiates this usecase to fetch the answer of the classifier to give input for the game on the data provided by the user while the user is playing any game. |
| Preconditions: | * Logged-in account * Stable Internet connection * Game started |
| Postconditions: | Success: In this case the system will fetch the command successfully.  Failure: In this case the command will not be fetched. |
| Normal Flow: | * Log-in account * Click button games on the home page * Choose a game |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Training process interrupted * Running model process interrupted * Game not started |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Game started |

**Table 42**

|  |  |
| --- | --- |
| Use Case ID: | UC-42 |
| Use Case Name: | Send data |
| Actors | System |
| Description: | This usecase takes the input from the user and takes it to the python files to be classified by the classifier. |
| Trigger: | System initiates this usecase to classify the data collected from the user while the user is playing any game. This is triggered in the loop to get input for the game . |
| Preconditions: | * Logged-in account * Stable Internet connection * Game started |
| Postconditions: | Success: In this case the data will be sent to python classifier successfully.  Failure: In this case the sending data process will somehow be interrupted. |
| Normal Flow: | * Log-in account * Click button games on the home page * Choose a game |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Training process interrupted * Game not started |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Game started |

#### Module 6: Database Handling

**Table 43**

|  |  |
| --- | --- |
| Use Case ID: | UC-43 |
| Use Case Name: | View database |
| Actors | Admin |
| Description: | This usecase lets the admin view the database which contains the data of all the users,their scores and their reports . |
| Trigger: | System initiates this usecase to view the database. This is triggered by a button on the admin account. Then the admin can choose which table to view . |
| Preconditions: | * Logged-in account * Stable Internet connection |
| Postconditions: | Success: In this case the admin will be able to access the database and view it successfully.  Failure: In this case the data will not be provided and a notification will inform the admin as to why the data is not available |
| Normal Flow: | * Log-in account * Click view database * Choose table |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Access denied * Database connection failed |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Access allowed |

**Table 44**

|  |  |
| --- | --- |
| Use Case ID: | UC-44 |
| Use Case Name: | Edit database |
| Actors | Admin |
| Description: | This usecase lets the admin edit the database which contains the data of all the users,their scores and their reports . |
| Trigger: | System initiates this usecase to edit the database. This is triggered by a button on the admin account. Then the admin can choose which table and value to edit. |
| Preconditions: | * Logged-in account * Stable Internet connection |
| Postconditions: | Success: In this case the admin will be able to access the database and edit it successfully.  Failure: In this case the data will not be provided and a notification will inform the admin as to why the data is not available |
| Normal Flow: | * Log-in account * Click view database * Choose table * Choose edit * enter the value to change and the new value in the pop-up window. |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Access denied * Database connection failed |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Access allowed |

**Table 45**

|  |  |
| --- | --- |
| Use Case ID: | UC-45 |
| Use Case Name: | Update database |
| Actors | Admin |
| Description: | This usecase lets the admin update the database which contains the data of all the users,their scores and their reports . |
| Trigger: | System initiates this usecase to update the database. This is triggered by a button on the admin account. Then the admin can choose which table an which value to update. |
| Preconditions: | * Logged-in account * Stable Internet connection |
| Postconditions: | Success: In this case the admin will be able to access the database and update it successfully.  Failure: In this case the data will not be provided and a notification will inform the admin as to why the data is not available |
| Normal Flow: | * Log-in account * Click view database * Choose table * Click update * Enter the cell to change and its new value in the pop-up window |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Access denied * Database connection failed |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Access allowed |

**Table 46**

|  |  |
| --- | --- |
| Use Case ID: | UC-46 |
| Use Case Name: | Delete from database |
| Actors | Admin |
| Description: | This usecase lets the admin delete from the database which contains the data of all the users,their scores and their reports . |
| Trigger: | System initiates this usecase to delete from the database. This is triggered by a button on the admin account. Then the admin can choose which table or row to delete. |
| Preconditions: | * Logged-in account * Stable Internet connection |
| Postconditions: | Success: In this case the admin will be able to access the database and delete from it successfully.  Failure: In this case the data will not be provided and a notification will inform the admin as to why the data is not available |
| Normal Flow: | * Log-in account * Click view database * Choose table * Choose row * Press button delete |
| Alternative Flows: | N/A |
| Exceptions: | * Internet connection unstable * Access denied * Database connection failed |
| Business Rules | N/A |
| Assumptions: | * Account logged-in * Access allowed |

## Functional Requirements

The Functional Requirements needs to be mentioned in this section.

## Non-Functional Requirements

Non-Functional requirements are required to be explicitly mention in this section.

# Design and Architecture



## System Architecture

[diagram + description]

## Data Representation

# Data design

{

{

**“$schema”** : “ “ ,

**“$id”** : ,

**“Title”** : “Users” ,

**“description”** : “all the accounts created”,

**“Type” : “object”,**

**“Properties” :**

**{**

“Username” : {

“Description” : “name of the user”,

“Type” : “String”

}

“Email” : {

“Description” : “email of the user”,

“Type” : “String”

}

“Password” : {

“Description” : “password for the authentication”,

“Type” : “varchar”

}

“Contact” : {

“Description” : “phone number of the user”,

“Type” : “num”

}

“profileImg” : {

“Description” : “picture of the user”,

“Type” : “Jpg , png”

}

**},**

**“Required” : [“username”, “email”, “password” , “contact” ]**

**}**

**{**

**“$schema” : ,**

**“$id” : ,**

**“Title” : “Admin” ,**

**“description” : “all the accounts of admins”,**

**“Type” : “object”,**

**“Properties” :{**

“AdminID” : {

“Description” : “Id assigned by the system for admin access”,

“Type” : “String”

}

“AdminPass” : {

“Description” : “password for admin authentication”,

“Type” : “varchar”

}

**},**

**“Required” : [“adminID”, “adminPass” ]**

**}**

**{**

**“$schema” : “ “ ,**

**“$id” : ,**

**“Title” : “Games” ,**

**“description” : “all the games in the application”,**

**“Type” : “object”,**

**“Properties” :{**

“gameName” : {

“Description” : “name of the game”,

“Type” : “string”

}

“gameID” : {

“Description” : “random ID assigned to the game”,

“Type” : “num”

}

**},**

**“Required” : [“gameName”, “gameID” ]**

**}**

**{**

**“$schema” : “ “ ,**

**“$id” : ,**

**“Title” : “Scores” ,**

**“description” : “scores recorded of all the users ”,**

**“Type” : “object”,**

**“Properties” :**

**{**

**“**Username” : {

“Description” : “name of the user”,

“Type” : “string”

}

“gameID” : {

“Description” : “ID of the game ”,

“Type” : “num”

}

“Score” : {

“Description” : “score of the user ”,

“Type” : “num”

}

**},**

**“Required” : [“username”, “gameID” , “score” ]**

**}**

**{**

**“$schema” : “ “ ,**

**“$id” : ,**

**“Title” : “Recordings” ,**

**“description” : “all the recording files”,**

**“Type” : “object”,**

**“Properties” :{**

“userName” : {

“Description” : “name of the user”,

“Type” : “string”

}

“recordingID : {

“Description” : “random ID assigned to the file by system”,

“Type” : “num”

}

“recordingLink” : {

“Description” : “link of the file uploaded in the database storage”,

“Type” : “string”

}

**},**

**“Required” : [“username”, “recordingID” , “recordingLink” ]**

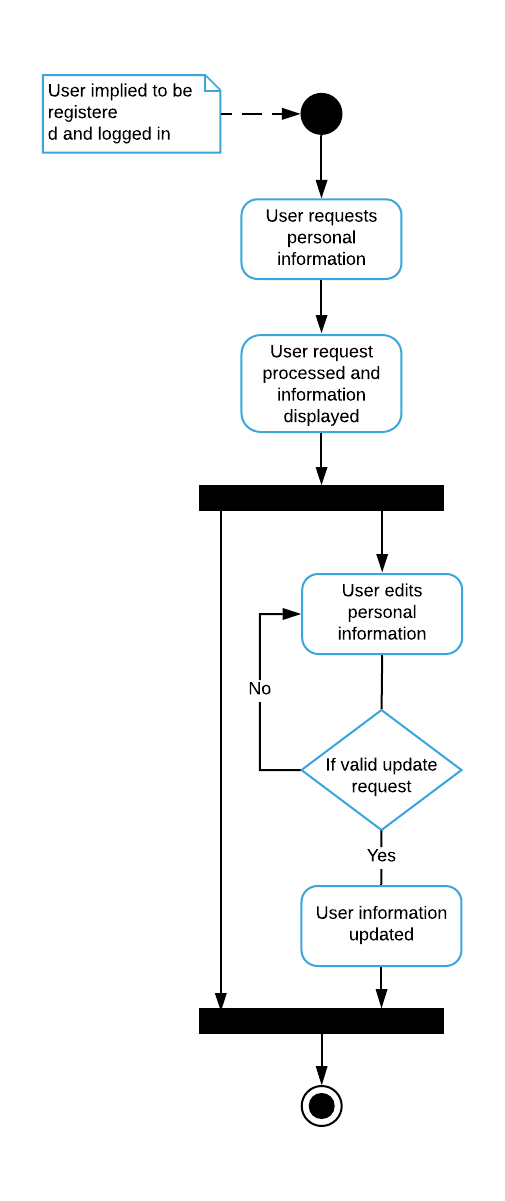
**}**

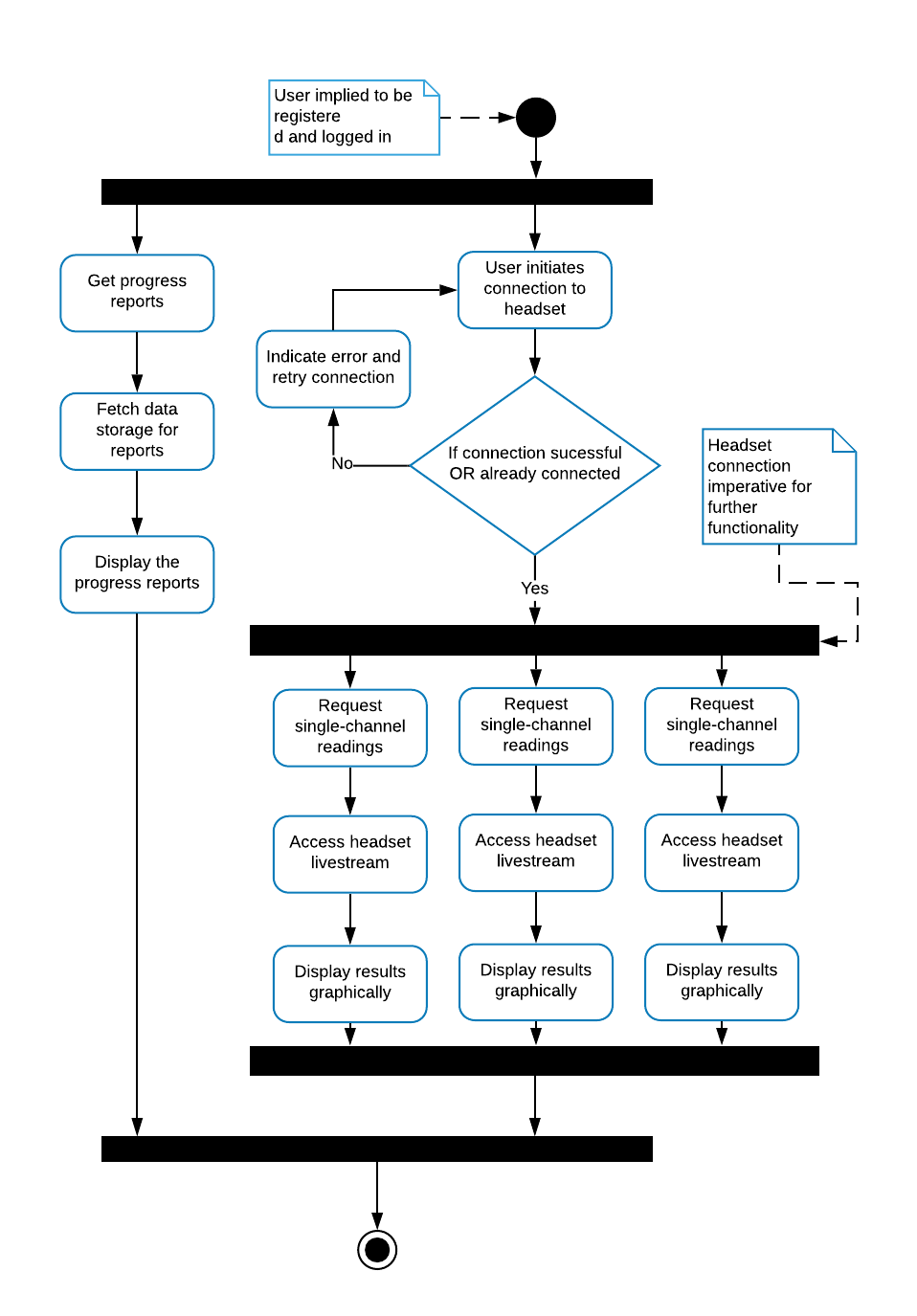
**}**

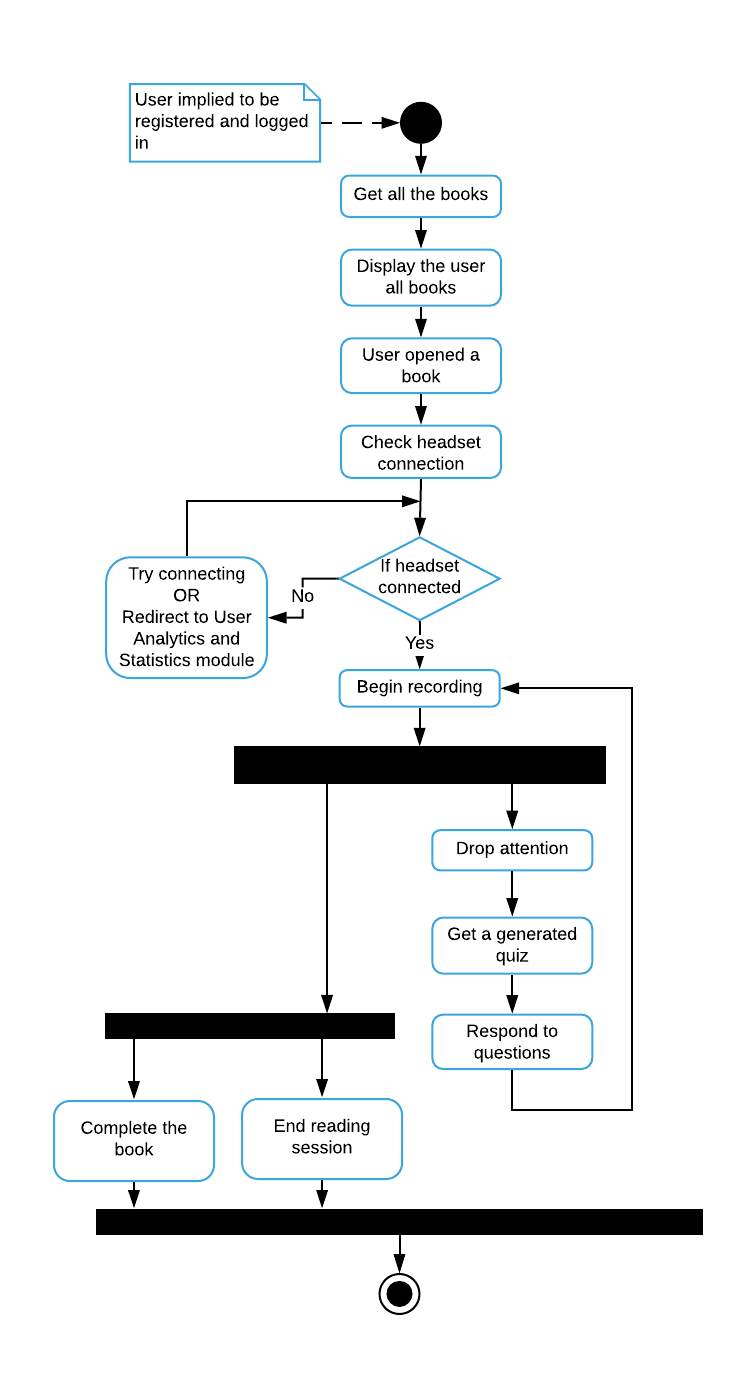
## Data dictionary

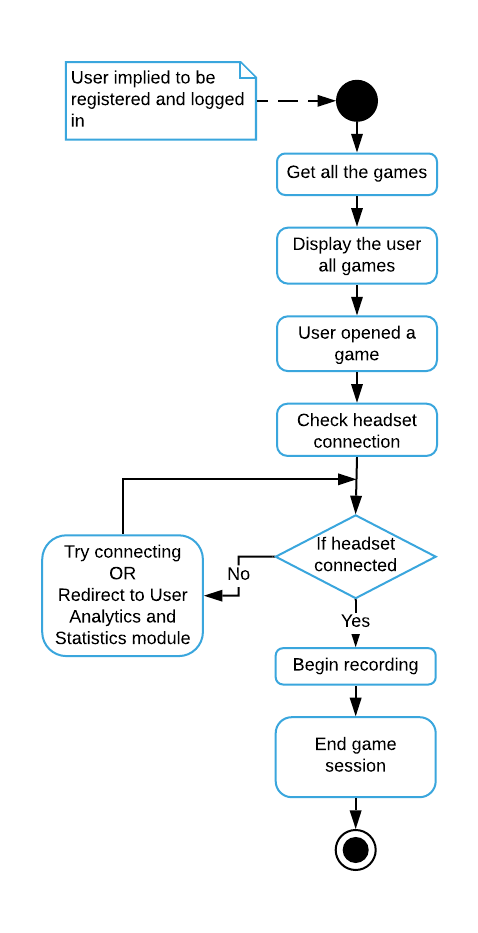
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FieldName** | **DataType** | **DataFormat** | **FieldSize** | **Description** | **Example** |
|  |  |  |  |  |  |
| username | string | text | Not specified | Full name of the user | “Kinza arshad” |
| email | string | text | Not specified | Email of the user | “maida@gmail” |
| password | VarChar | text | Not specified | Password for user authentication | “mustafa” |
| contact | num | text | Not specified | Phone number of the user | “03321576652” |
| profileImg | String | text | Not specified | Link of the profile uploaded on the database storage | “ “ |
| adminId | String | text | Not specified | Id assigned by the developers to the admins | “kinza@gmail” |
| adminPass | VarChar | text | Not specified | Admin password assigned to adminID for authentication | 13718847262” |
| gameName | String | text | Not specified | Name of all the games in the application | “ball\_jump” |
| gameID | num | text | Not specified | Game ID assigned to the game to identify it | “01’ |
| score | num | text | Not specified | Score to keep track of progress of the users | “20” |
| recordingID | num | text | Not specified | RandomID assigned by the system to the recording file | “0318487101” |
| recordingLink | String | text | Not specified | link of the file  In the storage | “ “ |

## Process Flow/Representation

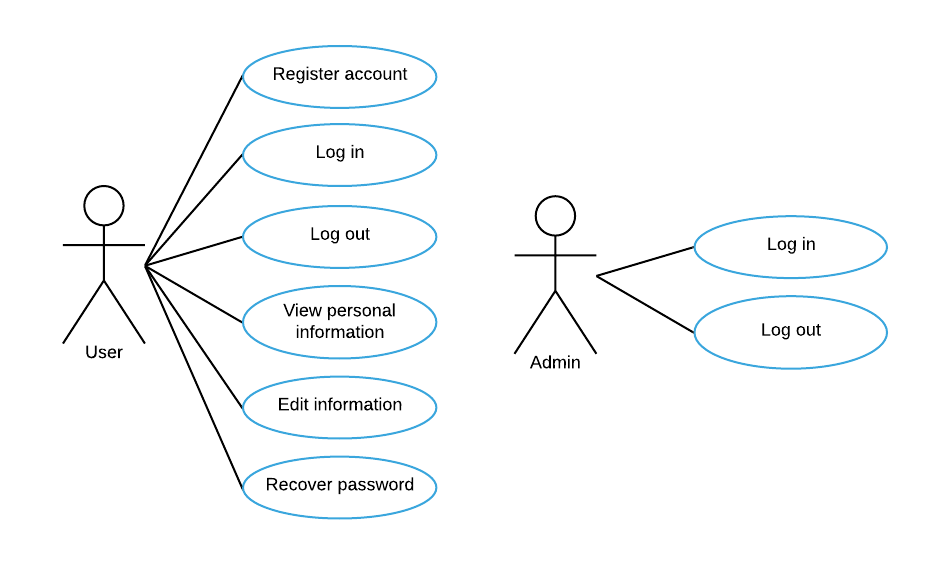


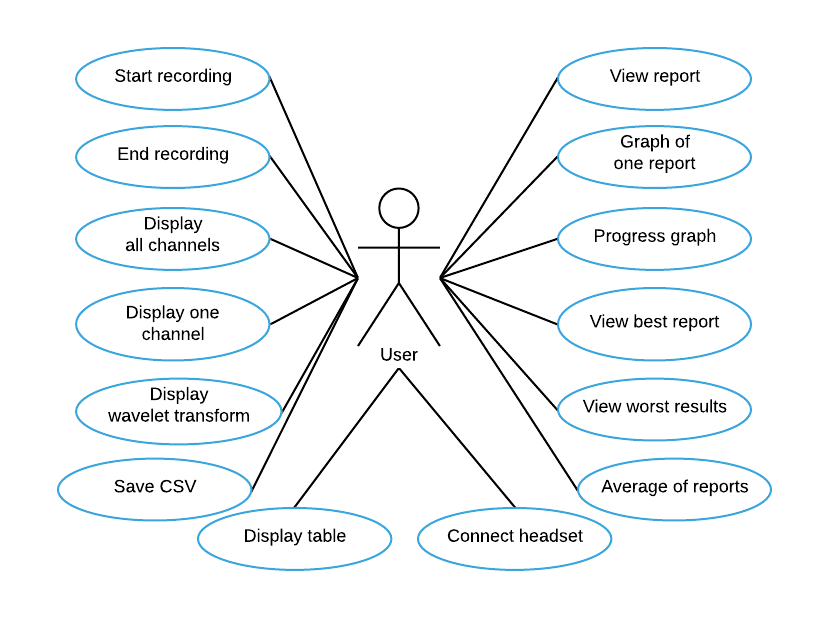


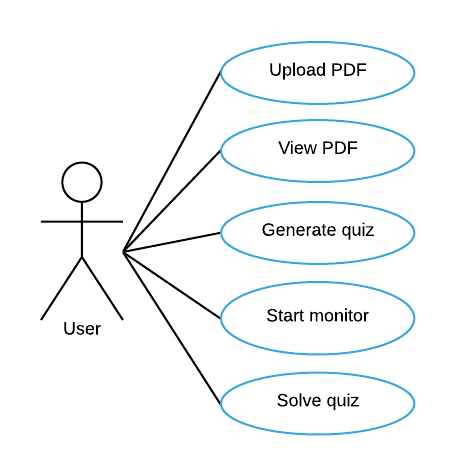


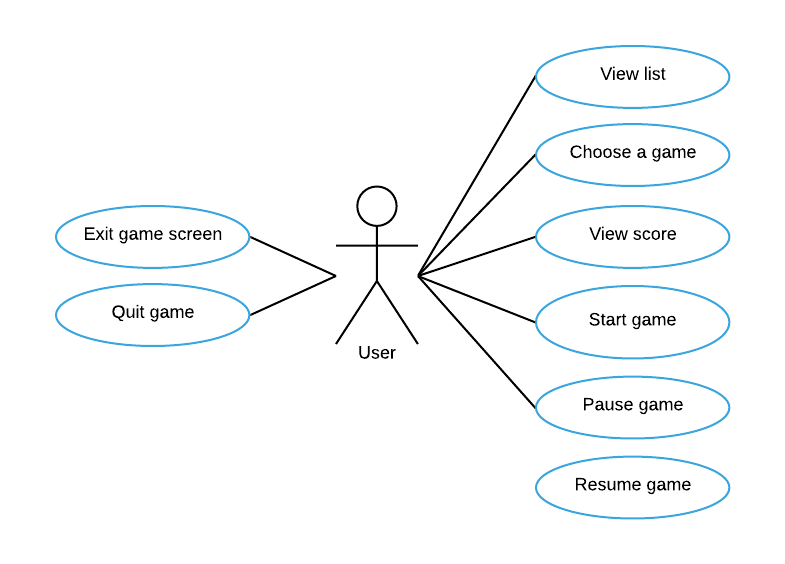


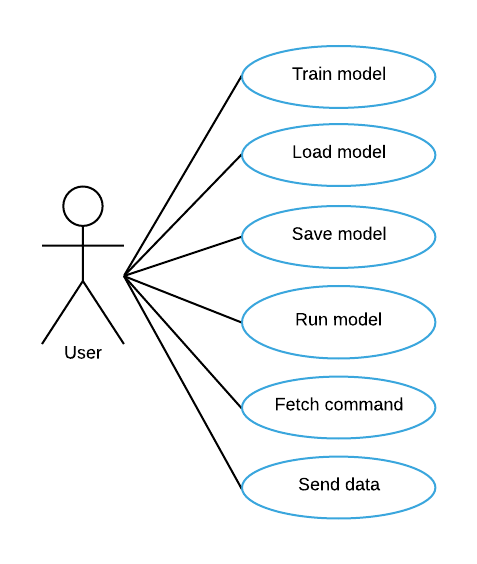
## Design Models

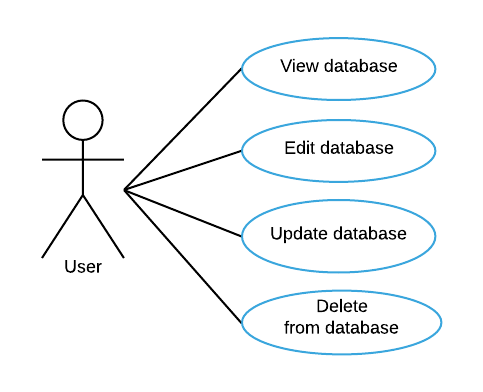












### Structural Diagrams

This section would present the static structure of the system, its parts on different abstraction and implementation levels, and how they are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts.

#### Class diagram



#### Object diagram

#### Component diagram

#### Package diagram

#### Deployment diagram

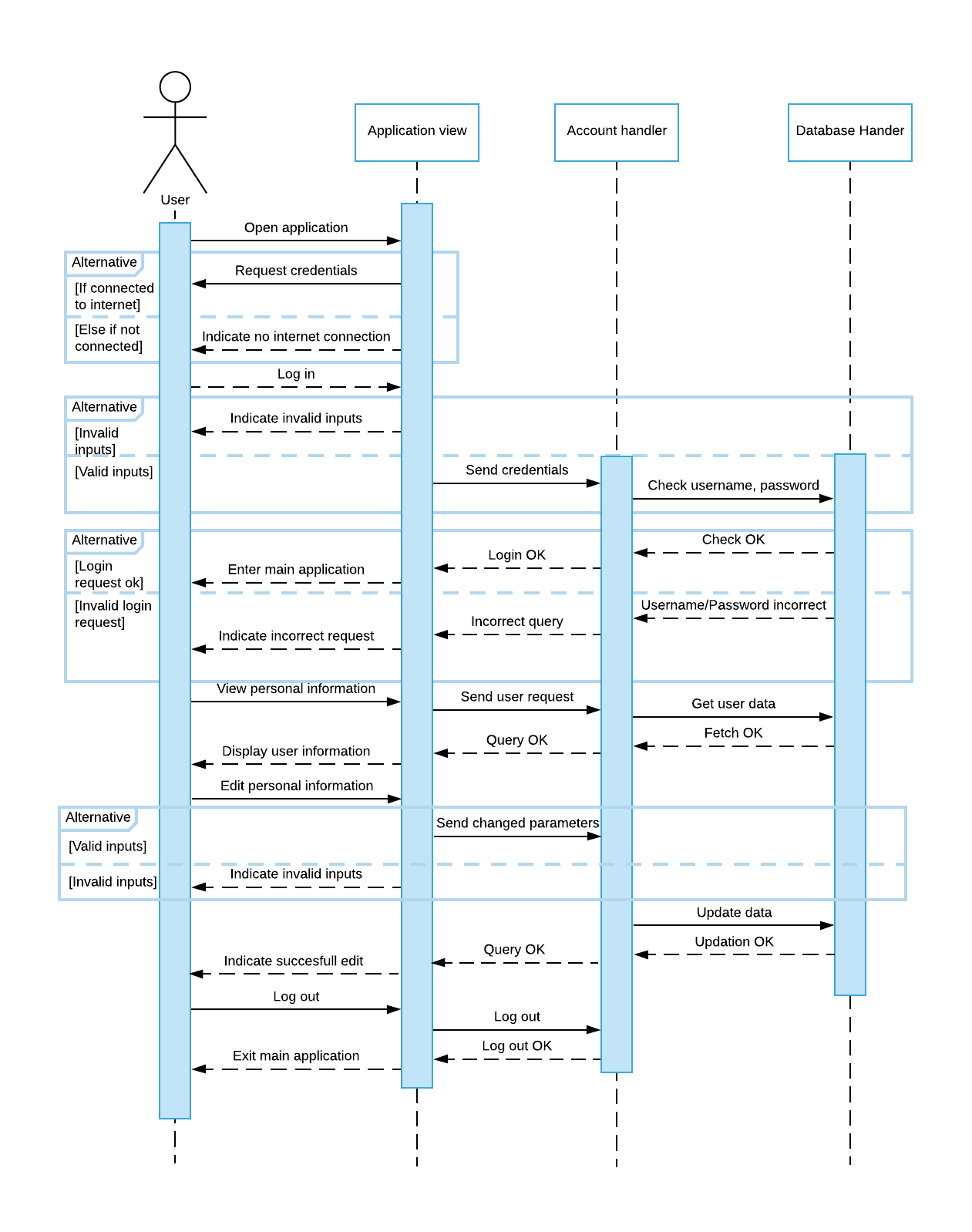
### Behavioral Diagrams

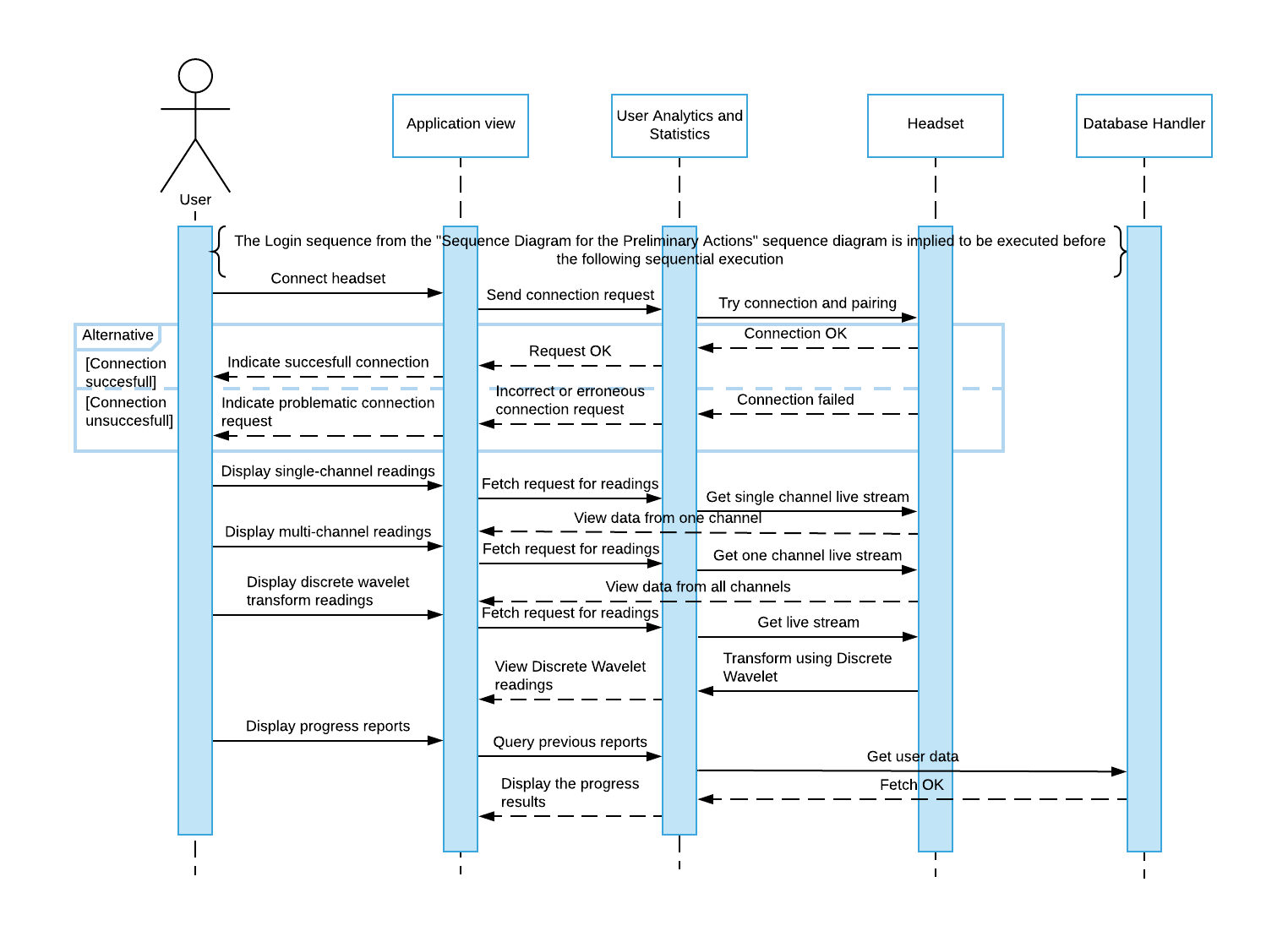
This section would present the behavior diagrams that show the dynamic behavior of the objects in a system, which can be described as a series of changes to the system over time.

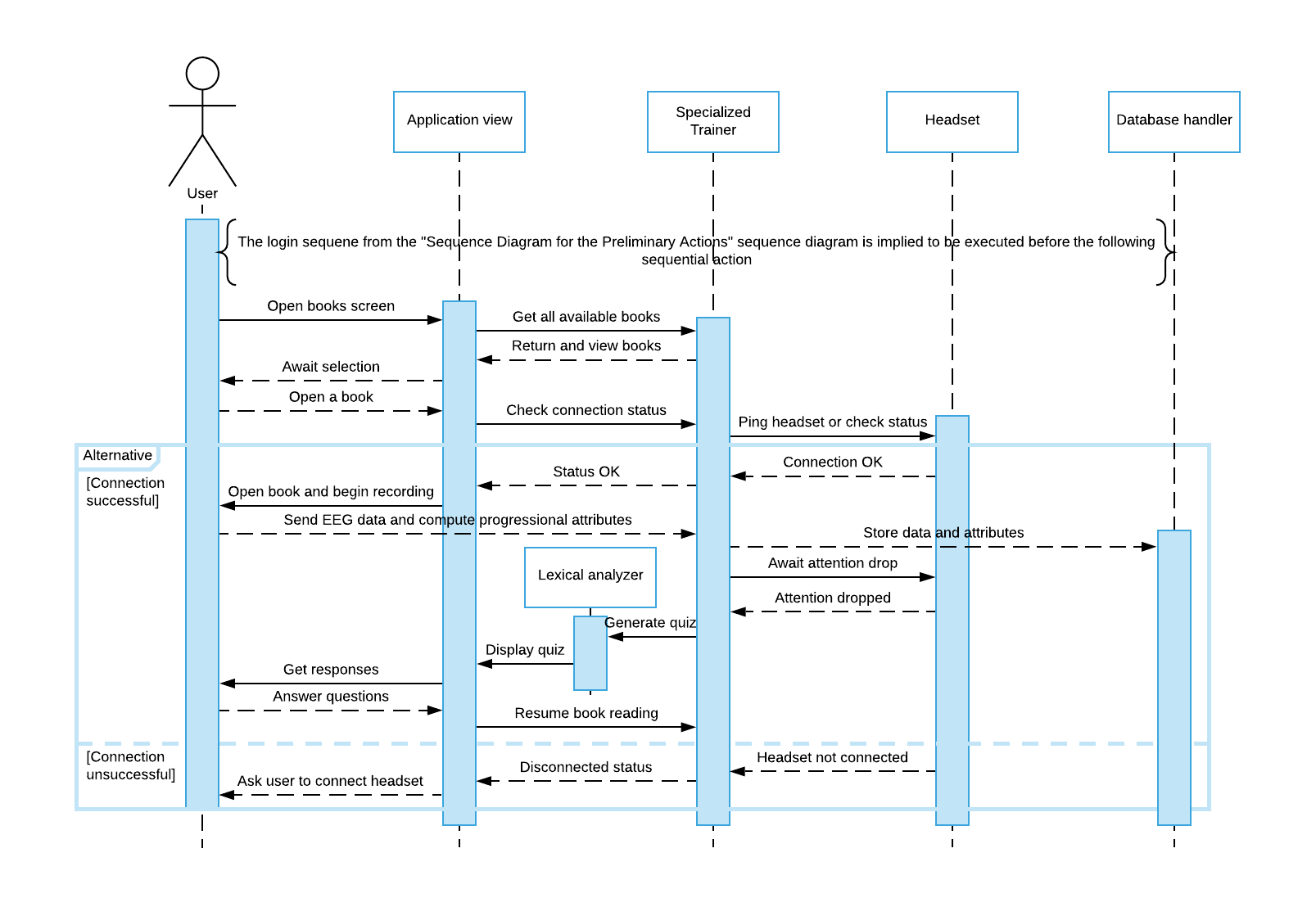
#### Activity diagram

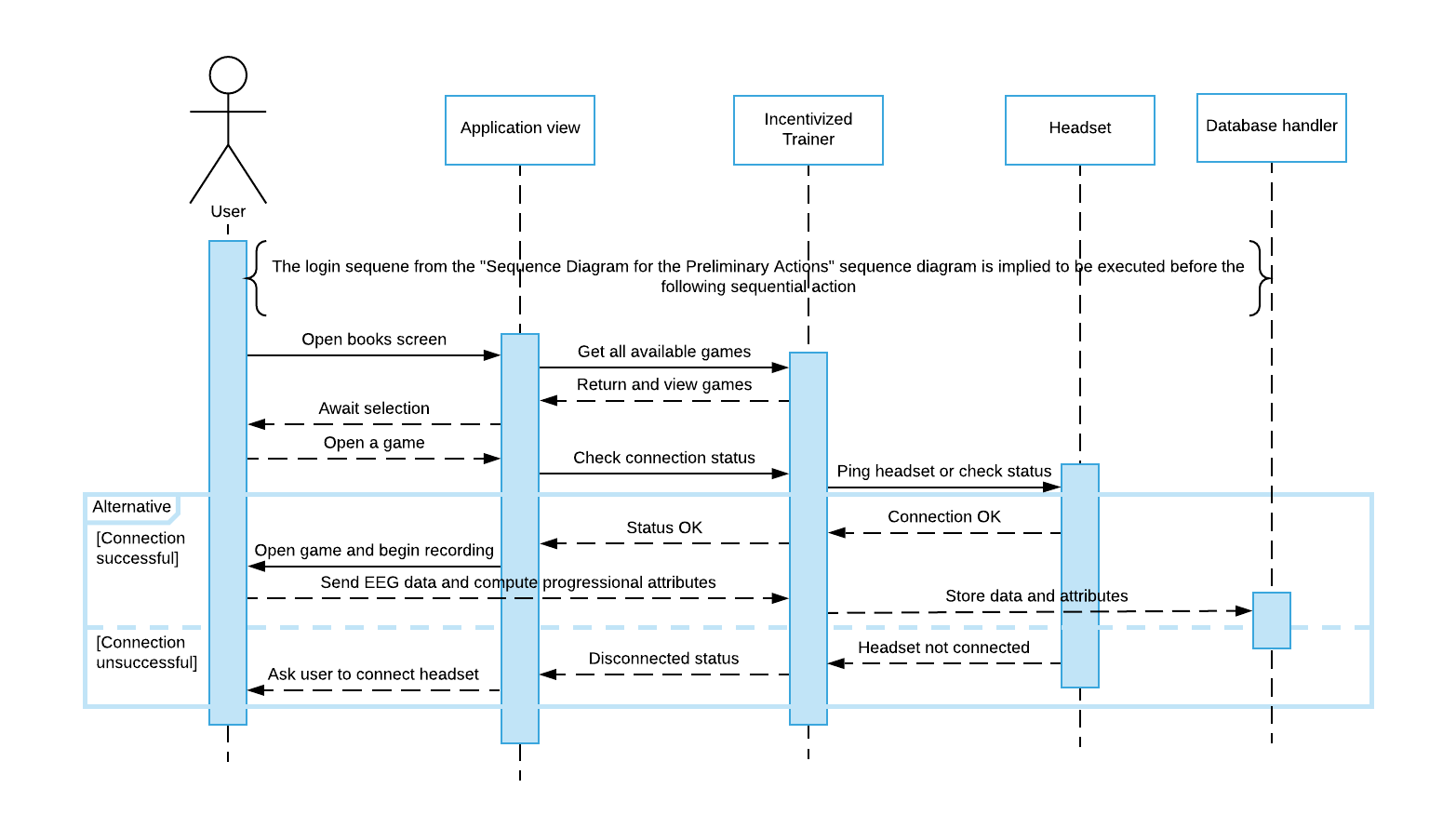
#### State machine diagram

#### Sequence diagram









#### Communication diagram

#### Interaction Overview diagram

#### Timing Overview diagram

Note this diagram can be used only for hardware related projects.

# Implementation

## Algorithm

Mention the algorithm(s) used in your project to get the work done with regards to major modules. Provide a pseudocode **OR** a natural language explanation regarding the functioning of main features. Be sure to use the correct syntax and semantics for algorithm representations.

## External APIs

**Table 5.1: Details of APIs used in the Project**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of API** | **Description of API** | **Purpose of Usage** | **List down the function/class name in which it is used** |
| **cortex** | This API is given by the Emotiv headset to access the data from the headset. | This API is used to locate and connect to the headset and then send data to the user as per subscription | It is used in the server file named index.js |
|  |  |  |  |

## User Interface

Details about user interface with descriptions will be presented in this section.

# Testing and Evaluation



## Manual Testing

### System Testing

Once the system has been successfully developed, testing has to be performed to ensure that the system working as intended. This is also to check that the system meets the requirements stated earlier. Besides that, system testing will help in finding the errors that may be hidden from the user. There are few types of testing which includes the unit testing, functional testing and integration testing. The testing must be completed before it is being deploy for user to use.

### Unit Testing

**Unit Testing 1:** Sign up

**Testing Objective:** To ensure the Sign up form is working properly.

**Test Case Id:** TC\_01

**Test Case Description:** Test the Sign up functionality.

**Test Scenario:**

**Table 1: Test Cases for Sign up form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify user Sign up after click on the ‘Sign up’ button on Sign up form with correct input data | Username:  03321576652  Password:  Stark506 | Successfully Account is created into the android application | Account is created successfully | Pass |
| 2. | Verify user Sign up after click on the ‘Sign up’ button on Sign up form with correct input data | Username:  0332157665245  Password:  123456 | Account not created in ATOM | Contact must be valid. | Fail |

**Unit Testing 1:** sign In

**Testing Objective:** To ensure the Sign In form is working properly.

**Test Case Id:** TC\_02

**Test Case Description:** Test the Sign in functionality.

**Test Scenario:**

**Table 1: Test Cases for Sign In form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify user Sign in after click on the ‘Sign in’ button on Sign in form with correct input data | Contact :  03321576652  Password:  Stark506 | Successfully Account is signed-in into the ARTINK website. | Sign in into the website successfully | Pass |
| 2. | Verify user Sign in after click on the ‘Sign in’ button on Sign in form with correct input data | Contact :  03321576652  Password:  1234 | Account is not signed-in. | Password incorrect | Fail |
| 3. | Verify user Sign in after click on the ‘Sign in’ button on Sign in form with correct input data | Contact:  03235065035  Password:  Ansa | Account not signed-in . | Contact not registered. | Fail |

**Unit Testing 1:** Sign out

**Testing Objective:** To ensure the Sign out is working properly.

**Test Case Id:** TC\_03

**Test Case Description:** Test the Sign out functionality.

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify user Sign out after click on the ‘Sign out’ button | User clicks on the Sign out button | Successfully Signed out . | Signed out successfully | Pass |

**Unit Testing 1:** Edit profile

**Testing Objective:** To ensure the Edit profile form is working properly.

**Test Case Id:** TC\_04

**Test Case Description:** Test the Edit profile functionality.

**Test Scenario:**

**Table 1: Test Cases for Edit profile form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify user Edit profile after click on the ‘Edit profile’ button on Edit profile form with correct input data | Contact :  03321576652  Username:  kinza  Password:  Stark506  Gender:  Female  Age :  21 | Successfully Profile is Edited . | Account is Edited successfully | Pass |
| 2. | Verify user Edit profile after click on the ‘Edit profile’ button on Edit profile form with correct input data | Contact :  Username:  kinza  Password:  Stark506  Gender:  Female  Age :  21 | Profile is not Edited | Contact is empty. | Fail |
| 4. | Verify user Edit profile after click on the ‘Edit profile’ button. | Contact :  03321576652  Username:  Password:  Stark506  Gender:  Female  Age :  21 | Profile is not Edited | Invalid username | Fail |

**Unit Testing 1:** view personal information

**Testing Objective:** To ensure the personal information Dashboard is working properly.

**Test Case Id:** TC\_06

**Test Case Description:** Test the personal information Dashboard functionality.

**Test Scenario:**

**Table 1: Test Cases for Customer Dashboard**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify personal information Dashboard is getting all the user information . | User slides the dashboard out . | Successfully personal information is retrieved from the cloud real-time. | Information is correct. | Pass |

**Unit Testing 1:** already exists check

**Testing Objective:** To ensure if the contact is already present.

**Test Case Id:** TC\_07

**Test Case Description:** Test the contact already exists check works.

**Test Scenario:**

**Table 1: Test Cases for Customer Dashboard**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify that profile is not created if the contact already exists. | Username:  03321576652 | Account not created. | Contact already exists. | Pass |

**Unit Testing 1:** contact validation.

**Testing Objective:** check if the contact validation from firebase works.

**Test Case Id:** TC\_08

**Test Case Description:** check if the code is generated to the contact number and checked properly.

**Test Scenario:**

**Table 1: Test Cases for Customer Dashboard**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify contact number with code generated | Code received on the message. | Successfully code verified and contact validated | Information is correct. | Pass |

**Unit Testing 1:** sign In(admin)

**Testing Objective:** To ensure the Sign In form is working properly.

**Test Case Id:** TC\_09

**Test Case Description:** Test the Sign in functionality.

**Test Scenario:**

**Table 1: Test Cases for Sign In form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify user Sign in after click on the ‘Sign in’ button on Sign in form with correct input data | Contact :  03321576652  Password:  Stark506 | Successfully Account is signed-in into the ARTINK website. | Sign in into the website successfully | Pass |
| 2. | Verify user Sign in after click on the ‘Sign in’ button on Sign in form with correct input data | Contact :  03321576652  Password:  1234 | Account is not signed-in. | Password incorrect | Fail |
| 3. | Verify user Sign in after click on the ‘Sign in’ button on Sign in form with correct input data | Contact:  03235065035  Password:  Ansa | Account not signed-in . | Contact not registered. | Fail |

**Unit Testing 1:** Sign out(admin)

**Testing Objective:** To ensure the Sign out is working properly.

**Test Case Id:** TC\_10

**Test Case Description:** Test the Sign out functionality.

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify user Sign out after click on the ‘Sign out’ button | User clicks on the Sign out button | Successfully Signed out . | Signed out successfully | Pass |

**Unit Testing 1: choose a game**

**Testing Objective:** To ensure that all the games are in the list.

**Test Case Id:** TC\_11

**Test Case Description:** Test the buttons functionality in the list.

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify all the buttons take to the corresponding games . | User clicks on the start buttons against different games. | Successfully opened the game and phone locked in landscape mode. | Game opens and phone locks in landscape mode | Pass |

**Unit Testing 1: view game-list**

**Testing Objective:** To ensure that all the games are in the list.

**Test Case Id:** TC\_12

**Test Case Description:** Test the scrollable functionality in the list.

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify the scrollable functionality in the list | User drags his/her finger on the screen to scroll through the list. | Successfully scrolled through the list | Scroll works and all the games are accessible. | Pass |

**Unit Testing 1: start game**

**Testing Objective:** To ensure that the game starts properly

**Test Case Id:** TC\_13

**Test Case Description:** game starts and scores are updated periodically

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify that the game starts properly | User clicks on the start buttons against different games. | Successfully opened the game and phone locked in landscape mode. | Game opens and phone locks in landscape mode | Pass |

**Unit Testing 1: quit game**

**Testing Objective:** To ensure that the game quits properly

**Test Case Id:** TC\_14

**Test Case Description:** game quits and takes the user back to the games-list

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify that the game quits properly | User clicks on the back button. | Successfully quits the game and takes the user back to the games-list. | Game quits and games-list is opened. | Pass |

**Unit Testing 1: exit game-screen**

**Testing Objective:** To ensure that the games/unity screen quits and takes us back to home page.

**Test Case Id:** TC\_15

**Test Case Description:** game-screen/unity quits and takes us to the home screen.

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify that the game-screen quits properly | User clicks on the back button. | Successfully quits the game and takes the user back to the home screen | Game quits and home-screen is opened. | Pass |

**Unit Testing 1: view scores**

**Testing Objective:** To ensure that the game-score updates periodically.

**Test Case Id:** TC\_16

**Test Case Description:** score is visible and updates periodically.

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify that the score updates properly. | User stays on the game and score is updated periodically. | Successfully updates the game score. | Updates the score periodically. | Pass |

**Unit Testing 1: pause game**

**Testing Objective:** To ensure that the game pauses properly.

**Test Case Id:** TC\_17

**Test Case Description:** game screen pauses and a smaller menu is opened.

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify that the game pauses properly. | User clicks on the pause button. | Successfully pauses the game and a smaller menu is opened. | Game pauses and a smaller menu is opened. | Pass |

**Unit Testing 1: resume game**

**Testing Objective:** To ensure that the game resumes properly.

**Test Case Id:** TC\_18

**Test Case Description:** game screen resumes.

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify that the game resumes properly. | User clicks on the resume button from the smaller menu. | Successfully resumes the game. | Game resumes. | Pass |

**Unit Testing 1: run model**

**Testing Objective:** To ensure EEG classification works properly.

**Test Case Id:** TC\_19

**Test Case Description:** EEG classification works and sends the commands to server.

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify that the EEG classification properly. | server calls the model periodically. | Successfully returns the commands. | Commands are returned to server. | Pass |

**Unit Testing 1: fetch command**

**Testing Objective:** To ensure unity gets the command.

**Test Case Id:** TC\_20

**Test Case Description:** unity fetches the commands from the server 50 times per second.

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify that the unity fetches commands periodically. | Unity calls the model 50 times per second. | Successfully fetches the commands from the server. | Commands are fetched from the server. | Pass |

**Unit Testing 1: save model**

**Testing Objective:** To ensure model is updated periodically.

**Test Case Id:** TC\_21

**Test Case Description:** EEG classification model is updated periodically.

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify that the EEG classification model works properly. | Unity updates the model. | Successfully updates the model. | model is updated to cortex profile | Pass |

**Unit Testing 1: load model**

**Testing Objective:** To ensure that the model is loaded from the cortex.

**Test Case Id:** TC\_22

**Test Case Description:** model loads , takes the EEG data from the headset and sends the classified command back to the server.

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify that the model loads properly. | Server loads the model periodically. | Successfully loads the model on cortex. | Model is loaded successfully. | Pass |

**Unit Testing 1: send data**

**Testing Objective:** To ensure the data from the headset is sent to cortex properly.

**Test Case Id:** TC\_23

**Test Case Description:** EEG data is sent from the headset to the cortex.

**Test Scenario:**

**Table 1: Test Cases for Sign out**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Verify that the EEG data from the headset is sent properly. | Server send the data to the cortex from the headset. | Successfully sends the data to the cortex. | Data is sent to the cortex from the headset. | Pass |

### Functional Testing

The functional testing will take place after the unit testing. In this functional testing, the functionality of each of the module is tested. This is to ensure that the system produced meets the specifications and requirements.

**Functional Testing 1:** Login with different roles

**Test Objective**: To ensure that the correct page with the correct navigation bar is loaded.

**Test Case Id:** BU\_003

**Test Case Description:**

**Test Scenario:**

**Table 6.3: Test Cases for Login with different Roles**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Login as a ‘FYP Committee’ member. | Username: L001  Password: 1234 | Main page for the FYP Committee member is loaded with the FYP Committee navigation bar | Login as a ‘FYP Committee’ member. | Pass |
| 2. |  |  |  |  |  |

### Integration Testing

**Table 6.4: Test Cases for Integration Testing of mention the Units Name**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail/Not Executed/ Suspended** |
| 1. | Login as “FYP Committee” member | Username: L001  Password: 1234 | Login successful and the FYP Committee page with its navigation bar is loaded and in the view profile page | As Expected | Pass |
| 2. | Upload student record for Project 1 | - | File successfully uploaded and return to the upload page. Student records are updated. | As Expected | Pass |
| 3. | View supervising student | - | The list of supervisees shown on the screen. | As Expected | Pass |
| 4. |  |  |  |  |  |

## Automated Testing:

This section will discuss the testing tools used to automatically test the targeted project.

**Table 6.5: Tools employed for Automated Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tool Name** | **Tool Description** | **Applied on [list of related Test Cases / FR / NFR]** | **Results** |
|  |  |  |  |
|  |  |  |  |

# Conclusion and Future Work



## Conclusion

Considering the presented problem and the solution, its fair to estimate the importance and relevance of this project, and within or even beyond the projected scope, this project will hold reasonable grounds for further research and development. Not only is this this beneficial for

the market at it will breed and environment of productivity but also a project worthy of investment and with potential to turn profit because we are attracting the general public, also in the process making them more productive in their society.

## Future Work

The results gained from this project can range from controlling a smart home to the entertainment industry and provide users a whole new gaming experience .It can be used to control robotic arms for cripples or a fully functional robot to do your bidding .

# References

References to any book, journal paper or website should properly be acknowledged. Please consistently follow the style. The following are few examples of different resources i.e. journal article, book, and website.

* 1. Lyda M.S. Lau, Jayne Curson, Richard Drew, Peter Dew and Christine Leigh, (1999), Use Of VSP Resource Rooms to Support Group Work in a Learning Environment, ACM 99, pp-2. **(Journal paper example)**
  2. Hideyuki Nakanishi, Chikara Yoshida, Toshikazu Nishmora and TuruIshada, (1996), FreeWalk: Supporting Casual Meetings in a Network, pp 308-314 **(paper on web)** http://www.acm.org/pubs/articles/proceedings/cscw/240080/p308-nakanishi.pdf
  3. Ali Behforooz& Frederick J.Hudson, (1996), Software Engineering Fundamentals, Oxford University Press. Chapter 8, pp255-235. **(book reference example)**
  4. Page Author, Page Title, http://www.bt.com/bttj/archive.htm, Last date accessed**. (web site)**