COMSATS University Islamabad (CUI)

Atom – Brain-Computer Interfacing using Electroencephalography

By

Student Name 1	**CIIT/FA16-BCS-054/ISB**
Student Name 2	**CIIT/FA16-BCS-108/ISB**

Supervisor

Dr. Yasir Faheem

Co-Supervisor

N/A

Bachelor of Science in Computer Science (2016-2020)

The candidate(s) confirm(s) that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.

MUHAMMAD FAIZAN BADAR KINZA ARSHAD

COMSATS University Islamabad (CUI)

Atom – Brain-Computer Interfacing using Electroencephalography

A project presented to
COMSATS University Islamabad

In partial fulfillment
of the requirement for the degree of

Bachelor of Science in Computer Science (2016-2020)

By

Student Name 1	**CIIT/FA16-BCS-054/ISB**
Student Name 2	**CIIT/FA16-BCS-108/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.

Muhammad Faizan Badar **Kinza Arshad**

MUHAMMAD FAIZAN BADAR KINZA ARSHAD

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

CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (CS) “Project title” was developed by **MUHAMMAD FAIZAN BADAR (CIIT/FA16-BCS-054)** and **KINZA ARSHAD (CIIT/FA16-BCS-108)** under the supervision of DR. YASIR FAHEEM and that in his 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

Progressively as generations evolve into the 21st century world, where technology has proliferated into every activity of individuals. The effects of this have been of different natures and varying magnitudes, one of which is a decline in attention spans. Social networking platforms requiring no physical or mental effort from individuals that can perform basic technological interfacing, continue to grow and attract more and more newer users towards becoming part of the victims of this “social addiction”. People spend most of their time scrolling through these applications, absorbing all sorts of entertainment content. Despite the fact that amongst these use cases, exist some with positive connotations of technological advances. 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 [2] and the control that the big organizations have over the human mind. And so is the importance of CT using certain BCIs [3].

BCI or Brain-Computer Interface, 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. 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. 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.

So, in nutshell, our system is a BCI that detects the mental state of the user while he or she is using the provided utilities and from the detected mental state makes the prediction with some confidence that whether the user is paying attention or has lost focus. The BCI will achieve this using the hardware provisions of an EEG system, a headset, to gather brain wave readings and judge the state by using pre-programmed experience and a learnt model from previous examples. EEG defines its readings to be of five types of waves
and if a specific part or cortex of the human brain is targeted, depending on the requirement and the nature of the part of brain, then while performing the activity any change in the mental state can be detected by different fluctuations in all these different types of waves. The command center is located in the frontal lobe which is also responsible for the conscious thought and voluntary movement. Although targeting one cortex limits us from the other aspects of the mental state detection such as emotion recognition, social involvement, the human vision. This also limits the overall accuracy due to lower spatial averaging, upon which the entire system is based.

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. Yasir Faheem. Without his personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to him for his 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.

Muhammad Faizan Badar Kinza Arshad

MUHAMMAD FAIZAN BADAR KINZA ARSHAD

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

ABBREVIATIONS

Abbr.	Description
CT	Cognitive Training
BCI	Brain-Computer Interfacing
EEG	Electroencephalography

TABLE OF CONTENTS

TABLE OF FIGURES

1 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.

1.1 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.

1.2 Relevance to Course Modules

- Machine Learning
- EEG Analysis /pattern Recognition
- Android Development
- Game design and development

1.3 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.

1.4 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.

1.5 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. A base module has been developed which we can re-implement in a more reasonable way [1]. Using a fine tuned, tech-equipped system and minimalistic, likely to be usable interface is a way of developing a product.

1.6 Methodology and Software Lifecycle for this Project

The software development lifecycle model that was incorporated was an improvisation over the waterfall model without the iterations limited to successive steps. Instead of the standard steps of a waterfall model, there were 4 main divisions in the steps for the model: the first divisional phase comprised of scope formation and a preamble setting for the requirements, the second reiterated and refined the preamble into well-formed requirements and presented the general architectural and functional design both at unit and system level, the third presented testing propositions in a documented format and the fourth relied totally on the evaluation and examination of the implementation. The development and practical application were underlying features throughout the model whereby each step involved some form of coding with sufficient increment to the system development. Development was assisted by the distributing the entire development framework using the Object-Oriented Design methodology.

1.6.1 Rationale behind the Selected Methodology

Entities that had sufficient self-identifying behavior and attributes were assigned the designation of being Objects whereby the composed units in particular groupable contexts were noted by Classes. This helped in translating the design into implementation using provisional language capabilities of Java and C# in a linear way without any extrapolation of design requirements necessary.

The following section explores the problem area that Atom targets to reveal, identify and eliminate from the ecosystem of all the engineering tools and utilities.

2 Problem Definition

2.1 Problem Statement

With the advent of millennialism, a decline in attention spans and a proliferation of abnormal dopamine cycle requirements, 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 brainchildren 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 [2] and the control that the big organizations have over the human mind. And so is the importance of CT using certain BCIs [3]. 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.

2.2 Deliverables and Development Requirements

The deliverables as stated earlier in the software development lifecycle, spanned over four phases: the first divisional phase comprised of scope formation and a preamble setting for the requirements, the second reiterated and refined the preamble into well-formed requirements and presented the general architectural and functional design both at unit and system level, the third presented testing propositions in a documented format and the fourth relied totally on the evaluation and examination of the implementation.

2.3 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.

3 Requirement Analysis

This section presents the well-formed system requirements from the preliminary steps in the software lifecycle. These requirements are presented in the form of use case models, documented use cases and functional level requirements.

3.1 Use Cases Diagram(s)

The use case diagrams representing the actors and their particular use cases in the context of modules are presented below:

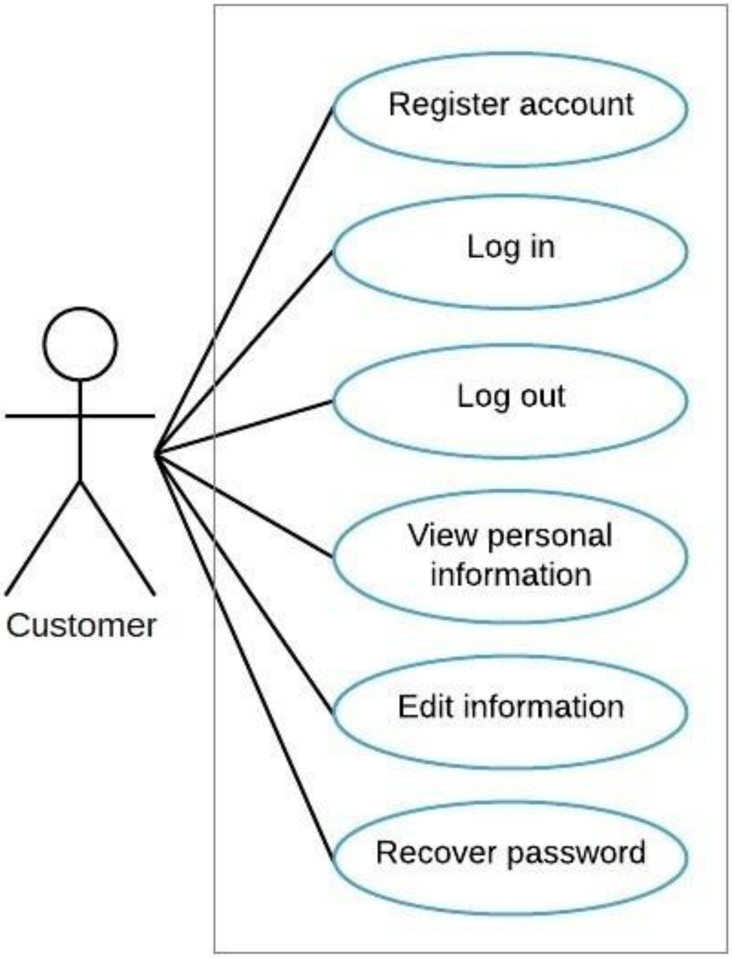


Figure 1 Use case module for Account Handling module

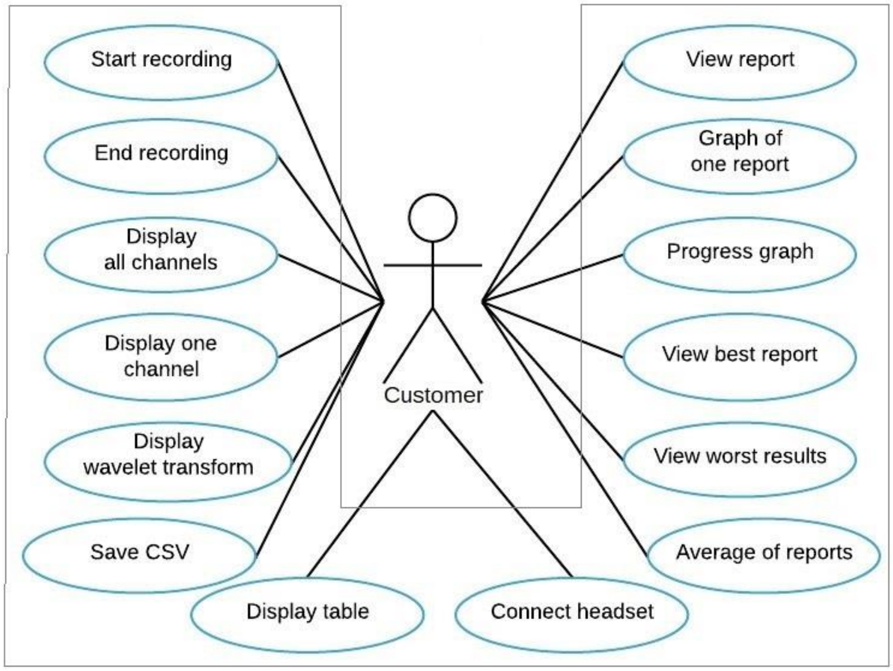


Figure 2 Use case model for User Analytics and Statistics module

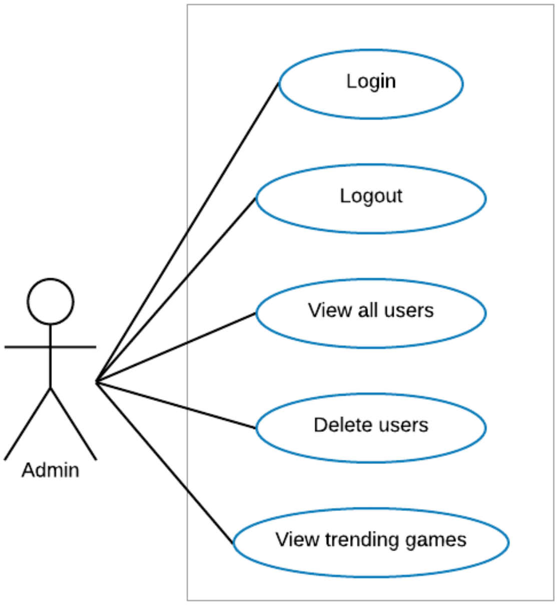


Figure 3 Use case model for Admin functionality

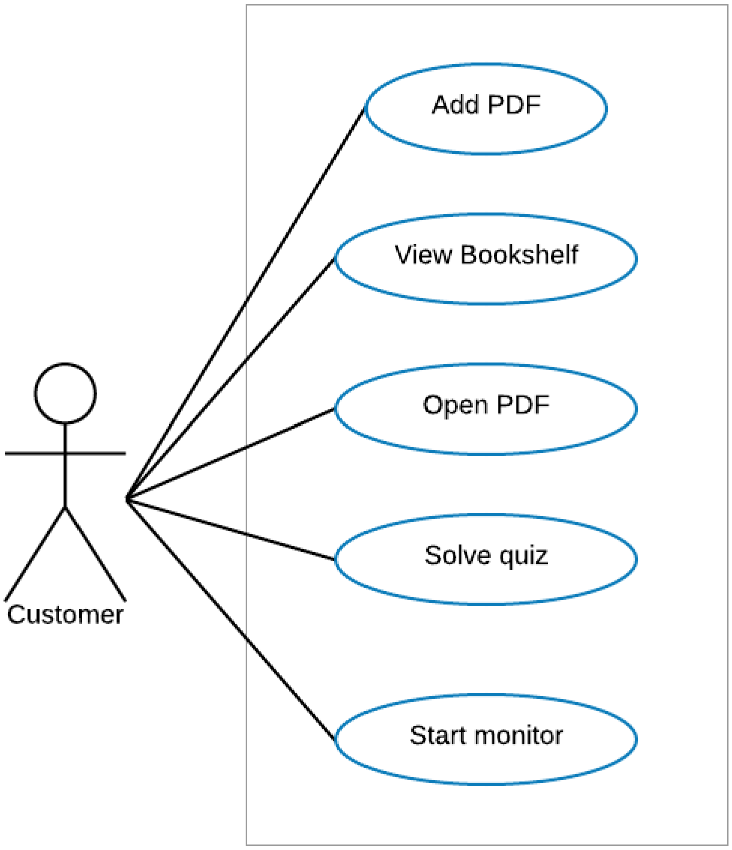


Figure 4 Use case model for Specialized Control Training module

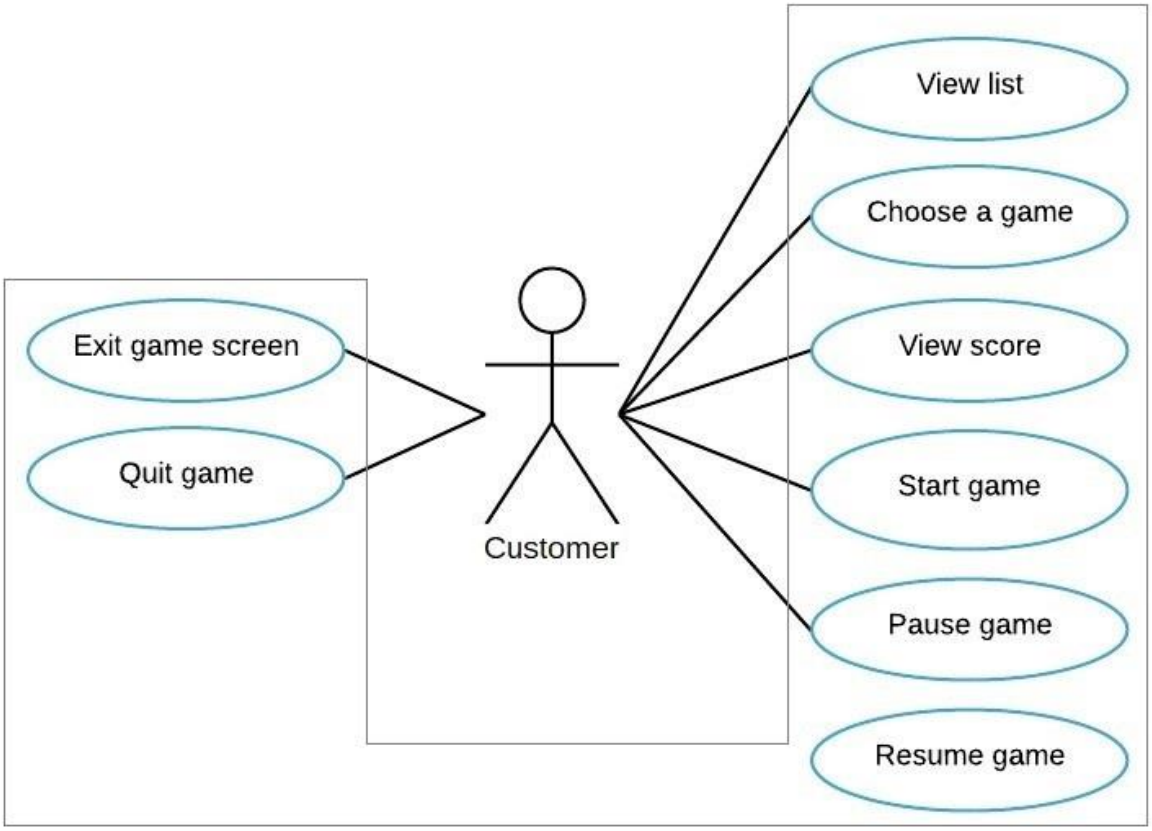


Figure 5 Use case model for Entertainment Incentivized Training module

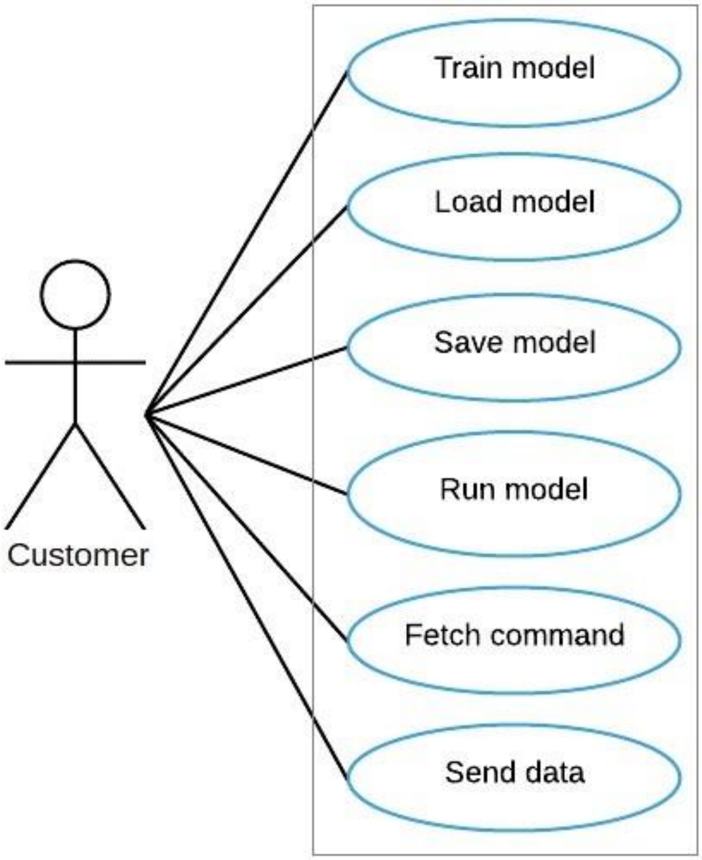


Figure 6 Use case model for EEG Feature Extraction module

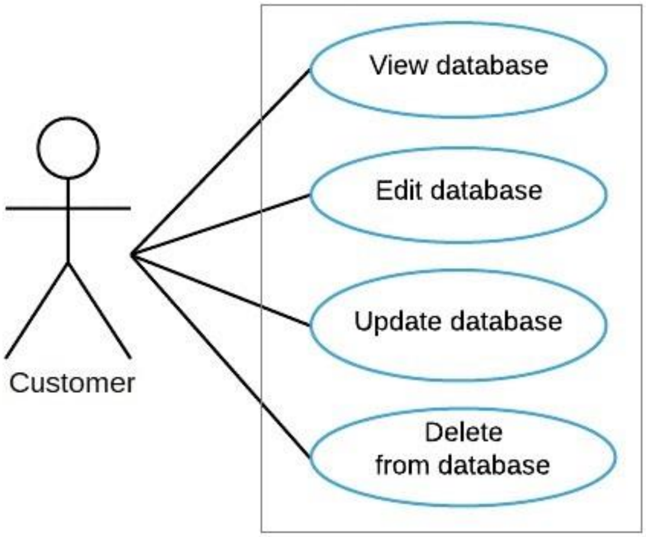


Figure 7 Use case model for Database Handling module

3.2 Use case descriptions

3.2.1 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

3.2.2 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

3.2.3 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

3.2.4 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

3.2.5 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

3.2.6 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

3.3 Functional Requirements

The functional level requirements with their source and rationale and the actual descriptions of the requirements are below:

3.3.1 Account Handling Functional Requirements

Identifier	1.1.1
Title	Registration parameter acquisition
Requirement	The system shall be able to acquire the required parameters from the user
Source	Developer
Rationale	Registration of new users
Dependencies	None
Priority	High

Identifier	1.1.2
Title	Registration database querying
Requirement	The system shall be able to query the database
Source	Developer
Rationale	So that entered parameters can be checked and validation, or, a new record can be created
Dependencies	None
Priority	High

Identifier	1.1.3
Title	Registration request validation
Requirement	The system shall be able to validate a registration request
Source	Developer
Rationale	All parameters collected after functional processing of the FR1.1 should be correct before they are entered into the system
Dependencies	FR1.1.2
Priority	High

Identifier	1.1.4
Title	Completed registration execution
Requirement	The system shall be able to make an entry for the new user
Source	Developer
Rationale	So that the user might be able to login in the system at the next time
Dependencies	FR1.1.3
Priority	High

Identifier	1.1.5
Title	Incomplete registration notification
Requirement	The system shall be able to notify the user that the registration request was incomplete or inaccurate
Source	Developer
Rationale	The user might be able to correct or submit the correct updated request
Dependencies	FR1.1.3

Priority	Medium

Identifier	1.2.1
Title	Login parameter acquisition
Requirement	The system shall be able to acquire parameters to login into the system from the user
Source	Developer
Rationale	The provision of the ability to login into the system
Dependencies	None
Priority	High

Identifier	1.2.2
Title	Login database querying
Requirement	The system shall be able to query the database to search for the user in the system
Source	Developer
Rationale	The entered parameters can be validated and verified
Dependencies	None
Priority	High

Identifier	1.2.3
Title	Login parameter validation
Requirement	The system shall be able to validate the entered parameters for login
Source	Developer
Rationale	Users that belong in the system should be able to access the functionality
Dependencies	FR1.2.2
Priority	High

Identifier	1.2.4
Title	Completed login execution
Requirement	The system shall login the user to the system
Source	Developer
Rationale	So that the user should be able to open access to the entire functionality of the app
Dependencies	None
Priority	High

Identifier	1.2.5
Title	Incomplete login notification
Requirement	The system shall be able to notify the user that the login request was incomplete
Source	Developer
Rationale	Correct parameter should only guarantee entrance into the system
Dependencies	None

Priority	Low

Identifier	1.3.1
Title	User log in/log out status access
Requirement	The system shall be able to get access to the current status of the user
Source	Developer
Rationale	Whether the user is logged in or logged out will act as a firewall or a valve to a lot of other functions
Dependencies	None
Priority	High

Identifier	1.3.2
Title	Logout request receival
Requirement	The system shall receive a request to log user out of it
Source	Developer
Rationale	Restricting access to unauthorized personnel from accessing the application
Dependencies	FR1.3.1
Priority	High

Identifier	1.4.1
Title	Categorical information access
Requirement	The system shall be able to retrieve all the information registered by the user
Source	Developer
Rationale	So that the system might be able to view it
Dependencies	None
Priority	Medium

Identifier	1.4.2
Title	User information display
Requirement	The system shall layout or display the user relevant information available to the system
Source	Developer
Rationale	Any system should be accessible regarding user's data access
Dependencies	FR1.4.1
Priority	Low

Identifier	1.5.1
Title	User data manipulation in database
Requirement	The system shall be able to modify existing information about the user in the database
Source	Developer
Rationale	In case of any mistake or error while record insertion, the user data should not lose integrity

Dependencies	None
Priority	Medium

Identifier	1.5.2
Title	Acquisition of changes
Requirement	The system shall be able to acquire the changes made to information by the user
Source	Developer
Rationale	To submit a edit information request
Dependencies	None
Priority	Medium

Identifier	1.5.3
Title	Validation of changes
Requirement	The system shall be able to validate and verify the correctness of all changes
Source	Developer
Rationale	To eliminate the possibility of losing data integrity and causing invalid records to exist in the database
Dependencies	None
Priority	Medium

Identifier	1.5.4
Title	Feedback on information change result
Requirement	The system shall notify the user if the changes have been made or the changes have been dropped due to some error
Source	Developer
Rationale	Reducing the gulf of evaluation in the use case of edit information
Dependencies	None
Priority	Medium

Identifier	1.6.1
Title	Forgot password request
Requirement	The system shall provide the user with the ability to submit a password forgotten request
Source	Developer
Rationale	The user does not get blocked from the system permanently
Dependencies	None
Priority	Medium

Identifier	1.6.2
Title	Database querying for user email
Requirement	The system shall be able to query the database with user email and/or password
Source	Developer

Rationale	To check whether the user’s email exists in the system or not, and/or change the password
Dependencies	None
Priority	Medium

Identifier	1.6.3
Title	Form generation for password reset
Requirement	The system shall be able to generate a link containing the password reset form
Source	Developer
Rationale	To let the user change the password but not review the old one
Dependencies	None
Priority	Medium

Identifier	1.6.4
Title	Automatic forgot password emailing
Requirement	The system shall be able to issue an email consisting of the password resetting instructions
Source	Developer
Rationale	The user may begin with the password resetting procedure or be notified that no such email exists in the account directory
Dependencies	None
Priority	Medium

Identifier	1.6.5
Title	Execution of password reset
Requirement	The system shall be able to change the password in the database upon the submission of a valid request
Source	Developer
Rationale	The user may resume access to the functionality of the entire system or the user may be notified of an error in the request submission
Dependencies	None
Priority	Medium

Identifier	1.8.1
Title	Admin login
Requirement	The admin of the system should be able to login towards a different view
Source	Developer
Rationale	To restrict elevated access to only the admins of the system
Dependencies	None
Priority	Low

Identifier	1.8.2

Title	Alternate route to admin login
Requirement	The system shall provide a different route to login as an Admin
Source	Developer
Rationale	To require different parameters for the admin login before granting elevated access
Dependencies	None
Priority	

3.3.2 User Analytics and Statistics Functional Requirements

Identifier	2.10.1
Title	Saved background activity monitoring
Requirement	The system shall record all neural activity (focus/attention graphs) over every task performed
Source	Developer
Rationale	This recorded activity can be viewed by the user
Dependencies	None
Priority	High

Identifier	2.10.2
Title	Report acquisition from storage
Requirement	The system shall acquire either a desired one, or all of the reports from their storage place
Source	Developer
Rationale	These acquired reports will then be viewed
Dependencies	None
Priority	High

Identifier	2.10.3
Title	Report graphical display
Requirement	The numerical data collected over a certain period of time can be plotted by the system using a graph
Source	Developer
Rationale	The user can note the trend over a select period of time
Dependencies	FRs 2.10.1 & 2.10.2
Priority	High

Identifier	2.10.4
Title	Statistical inference from data
Requirement	The system shall be able to make certain deductions and inferences using statistical and logical reasoning
Source	Developer

Rationale	The user can take advantage of accurate logical conclusion rather than developing crude ones
Dependencies	None
Priority	Low

Identifier	2.10.5
Title	Report tabular display
Requirement	The system should be able to display conclusions and inferences drawn from data in a tabular or any textual format
Source	Developer
Rationale	The user can do a brief qualitative analysis
Dependencies	FRs 2.10.1, 2.10.2 & 2.10.4
Priority	High

Identifier	2.13.1
Title	Criteria based report selection
Requirement	The system shall be able to select a report on certain reprogrammable criterion
Source	Developer
Rationale	To give a more depth progress analysis of the system
Dependencies	None
Priority	Low

Identifier	2.13.2
Title	Greatest average attention report
Requirement	The system selects the report with the highest overall attention average per given time
Source	Developer
Rationale	Improving the progress analysis
Dependencies	FR 2.13.1
Priority	Low

Identifier	2.14.1
Title	Lowest average attention report
Requirement	The system selects the report with the lowest overall attention average per given time
Source	Developer
Rationale	Improving the progress analysis
Dependencies	None
Priority	Low

Identifier	2.15.1
Title	Summation of results
Requirement	The system shall be able to summarize the different types of result forming a consensus of the results

Source	Developer
Rationale	One crude summarized result will be quicker to conclude from
Dependencies	None
Priority	Low

Identifier	2.16.1
Title	Live data access from headset
Requirement	The system shall be able to get access to the live data from the headset
Source	Developer
Rationale	To inform user of how EEG recording is used in the application
Dependencies	None
Priority	Low

Identifier	2.16.2
Title	Live animated plotting of results
Requirement	The system shall be able to plot all values attained by the headset in real time
Source	Developer
Rationale	To give a more real time simulation to the user
Dependencies	None
Priority	Low

Identifier	2.16.3
Title	Begin recording of neural activity
Requirement	The system shall be informable on when to begin the recording
Source	Developer
Rationale	So that the recording session may start
Dependencies	FRs 2.16.1 & 2.16.2
Priority	Low

Identifier	2.17.1
Title	End recording of neural activity
Requirement	The system shall be informable on when to end the recording
Source	Developer
Rationale	So that the recording can be ended
Dependencies	None
Priority	Low

Identifier	2.18.1
Title	Multi-channel data plotting
Requirement	The system shall be to plot data from multiple channels
Source	Developer
Rationale	To give the technical user group a much more detailed insight into the EEG recording
Dependencies	FRs 2.16.1 & 2.16.2

Priority	Low

Identifier	2.19.1
Title	Single channel isolated plotting
Requirement	The system shall be able to plot data from a single selected channel
Source	Developer
Rationale	Give a choice to the different types of user
Dependencies	2.16.1 & 2.16.2
Priority	Low

Identifier	2.20.1
Title	Wavelet-transform output
Requirement	The system shall be able to plot wavelet-transformed data
Source	Developer
Rationale	This algorithm gives a cleaner and more conclusive output
Dependencies	None
Priority	Low

Identifier	2.21.1
Title	Storage of recording file
Requirement	The system shall be able to save both the user recorded activity and the automatically collected activity
Source	Developer
Rationale	To later use and view them
Dependencies	FRs 2.16.1 & 2.16.2
Priority	Low

Identifier	2.23.1
Title	Scanning for headsets
Requirement	The system should be able to scan nearby devices for a headset
Source	Developer
Rationale	So that the headset can be connected
Dependencies	None
Priority	High

Identifier	2.23.2
Title	Connection request
Requirement	The system should be able to connect to a nearby EEG headset
Source	Developer
Rationale	So as to get access to the core functionality
Dependencies	None
Priority	High

3.3.3 Specialized Control Training Functional Requirements

Identifier	3.24.1
Title	Add new book
Requirement	The system shall allow the user to add new book(s) to the bookshelf
Source	Developer
Rationale	To arrange all users book and display them within the application
Dependencies	None
Priority	High

Identifier	3.25.1
Title	Book reading
Requirement	The system shall allow any added book to be read
Source	Developer
Rationale	While the user performs this task, the application can monitor the background activity of the user
Dependencies	None
Priority	High

Identifier	3.25.2
Title	Changing the view mode the book reader
Requirement	The system shall be available multiple view modes in the book reader
Source	Developer
Rationale	The user reads the book in their preferred reading mode - Continuous - One page - Double Page
Dependencies	None
Priority	High

Identifier	3.25.3
Title	Changing the swipe gesture for page change
Requirement	The system should provide multiple alternatives for page changing
Source	Developer
Rationale	The user changes the pages while reading according to their preference: - Right to Left; Next page - Left to Right; Next page - Up to down; Next page - Down to up; Next page
Dependencies	None
Priority	High

Identifier	3.25.4

Title	Moving to another point in the book
Requirement	The system shall allow the user to scroll through or move to a certain referenced page in the book
Source	Developer
Rationale	A natural function during the design of most book readers
Dependencies	None
Priority	High

Identifier	3.25.5
Title	Changing the day mode of the reader
Requirement	The system shall allow the user to change the reading theme mode
Source	Developer
Rationale	Providing readers ease at day and night: - Day mode - Night mode
Dependencies	None
Priority	High

Identifier	3.26.1
Title	Semantic tree buildup
Requirement	The system should be able to build a semantic tree of any grammatically correct text
Source	Developer
Rationale	This semantic tree will be used to make questions for the user
Dependencies	None
Priority	Medium

Identifier	3.26.2
Title	Questions elicitation
Requirement	The system shall be able to elicit text relevant questions
Source	Developer
Rationale	So that a quiz or a comprehension can be made
Dependencies	None
Priority	Medium

Identifier	3.28.1
Title	Solvable and Gradeable Quiz
Requirement	The system shall display a solvable and gradable quiz
Source	Developer
Rationale	The user may solve the quiz for reinforcing the user’s attention
Dependencies	None
Priority	Medium

Identifier	4.32.1

Title	Background monitoring module
Requirement	Whilst the playing of any game, the background monitoring module should be active
Source	Developer
Rationale	The neural activity can be recorded and displayed later on
Dependencies	None
Priority	High

Identifier	4.32.2
Title	Game Engine Executor
Requirement	The games installed should be provided with their corresponding engine (Unity) to execute in
Source	Developer
Rationale	So that the user may play the games
Dependencies	None
Priority	High

Identifier	4.32.3
Title	Realtime EEG signal processing
Requirement	With the headset connected, the system should be able to fetch live EEG data from the headset and process it
Source	Developer
Rationale	This EEG data will contain the command and the controls for the immersive gaming
Dependencies	None
Priority	High

Identifier	4.32.4
Title	Brain state feature acquisition
Requirement	The system shall be able to extract or obtain the features of the brain state especially focus level or any other motor actions
Source	Developer
Rationale	These will be the primary inputs to the controller of the game
Dependencies	4.32.3
Priority	High

3.3.4 Database Handling Functional Requirements

Identifier	6.43.1
Title	Hosted database
Requirement	There should be a database pre-hosted
Source	Developer
Rationale	The basic purpose of storing data

Dependencies	None
Priority	High

Identifier	6.43.2
Title	Database connector
Requirement	The system should retort to a connector to connect to this hosted database
Source	Developer
Rationale	All transactions of database management should be held through this connector
Dependencies	FR6.43.1
Priority	High

Identifier	6.43.3
Title	Database selection
Requirement	The system should be able to select certain part of the database based on any combination of conditions
Source	Developer
Rationale	To view view-relevant attributes and records
Dependencies	FRs 6.43.1 & 6.43.2
Priority	High

Identifier	6.44.1
Title	Database insertion
Requirement	The system should be able to insert into the database
Source	Developer
Rationale	To make new users entities of the system
Dependencies	None
Priority	FRs 6.43.1 & 6.43.2

Identifier	6.45.1
Title	Database updation
Requirement	The system should be able to modify the database records based on any combination of conditions
Source	Developer
Rationale	To accommodate for any changes any system user wishes to make
Dependencies	FRs 6.43.1 & 6.43.2
Priority	High

Identifier	6.46.1
Title	Database deletion
Requirement	The system should be able to delete the database records based on any combination of conditions

Source	Developer
Rationale	To optimize storage by eliminating unnecessary data
Dependencies	FRs 6.43.1 & 6.43.2
Priority	High

3.4 Non-Functional Requirements

A non-functional requirement chart is provided:
While it is prudent to refrain from considering Organizational and External requirements, our focus is primarily on Product Requirements. We think that the other types of requirements need not be taken stress on because they'll dampen commercialization of the product by putting unnecessary constraints.

Speed	**Connection**	The headset connection should be under a reasonable time of less **4 seconds** (Comparing to normal Bluetooth and other connections)
	Data Acquisition	The real-time data feed should be seamless and continuous with at **1 Hz** of data sampling
	Interface	The interface animations and transitions should be **fluid** and **effortless**
Storage	**Database**	Although this lies under the shadow of constrained area of this project, but Firebase would be sufficient
	Local smartphone	The local space requirement should be kept under the order of **tens of Megabytes (< 90Mb)**
Ease of use	**Application**	The application interface should be **interactable**, **understandable**¸ **considerate** of all types of users, **accessible**, and **discoverable**
Portability	**Smartphone**	Portability is obvious
	Headset	- **Size:** The headset should be compact and durable - **Auxiliaries:** There should be no auxiliaries or supplementary connection equipment

4 Design and Architecture

This section discusses the major design decisions and their effects. The entire design is distributed on many levels to facilitate and aid development whereby abstraction level ranges from being totally functional to systemic level. All these design models and descriptions are provided below:

4.1 System Architecture

Since the primary choice to deploy Atom is to release an application or software on a platform which a mass majority of users are comfortable and familiar with, we've streamlined production on smartphone and due to technical barriers, the pre-dominant choice is Android. According to convention, Android targeted software are inherently followers of the Model-View-Controller paradigm, and in addition to that, the below diagram begins to concretize by providing further insight into each block. An implied improvisation on this architecture is the inclusion of the headset with the controller and not allocating another block for the peripheral to prevent breaking standard convention, although an accurate visualization might consider this as a separate block.

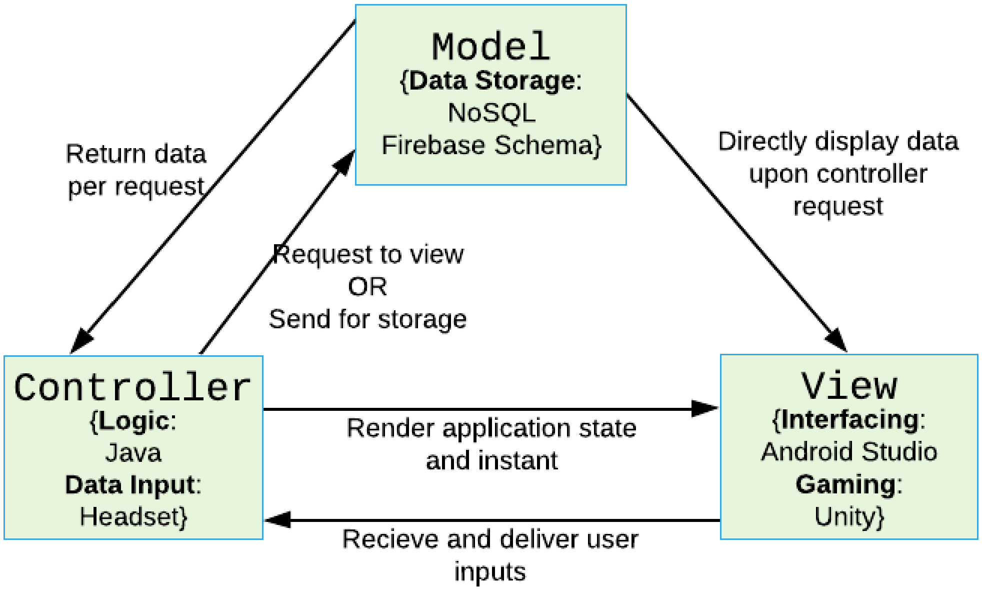


Figure 8 System architecture

4.2 Data Representation

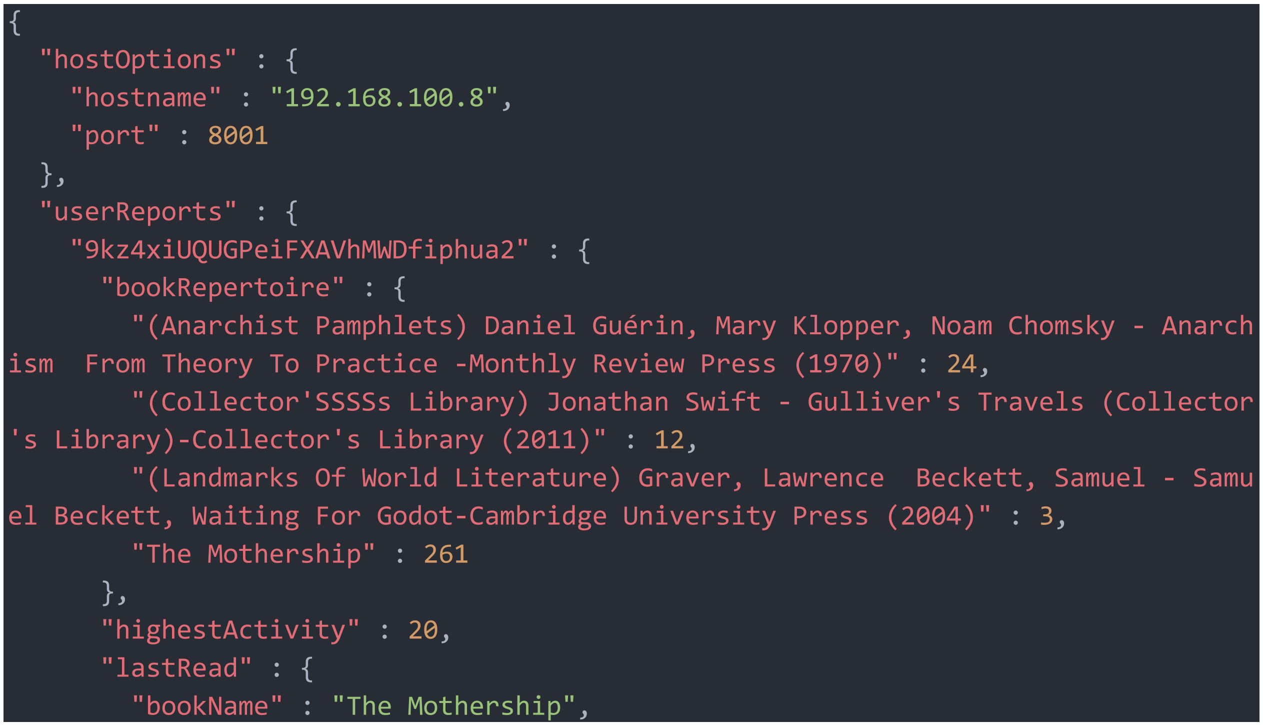
In consideration of all the models, descriptions, and diagram of Section 4; the representations of data exist in a versatile array. Often diagram discuss an overall flow, while some present forward actual storage mechanisms for the bytes and bits of the data.

4.3 Data design

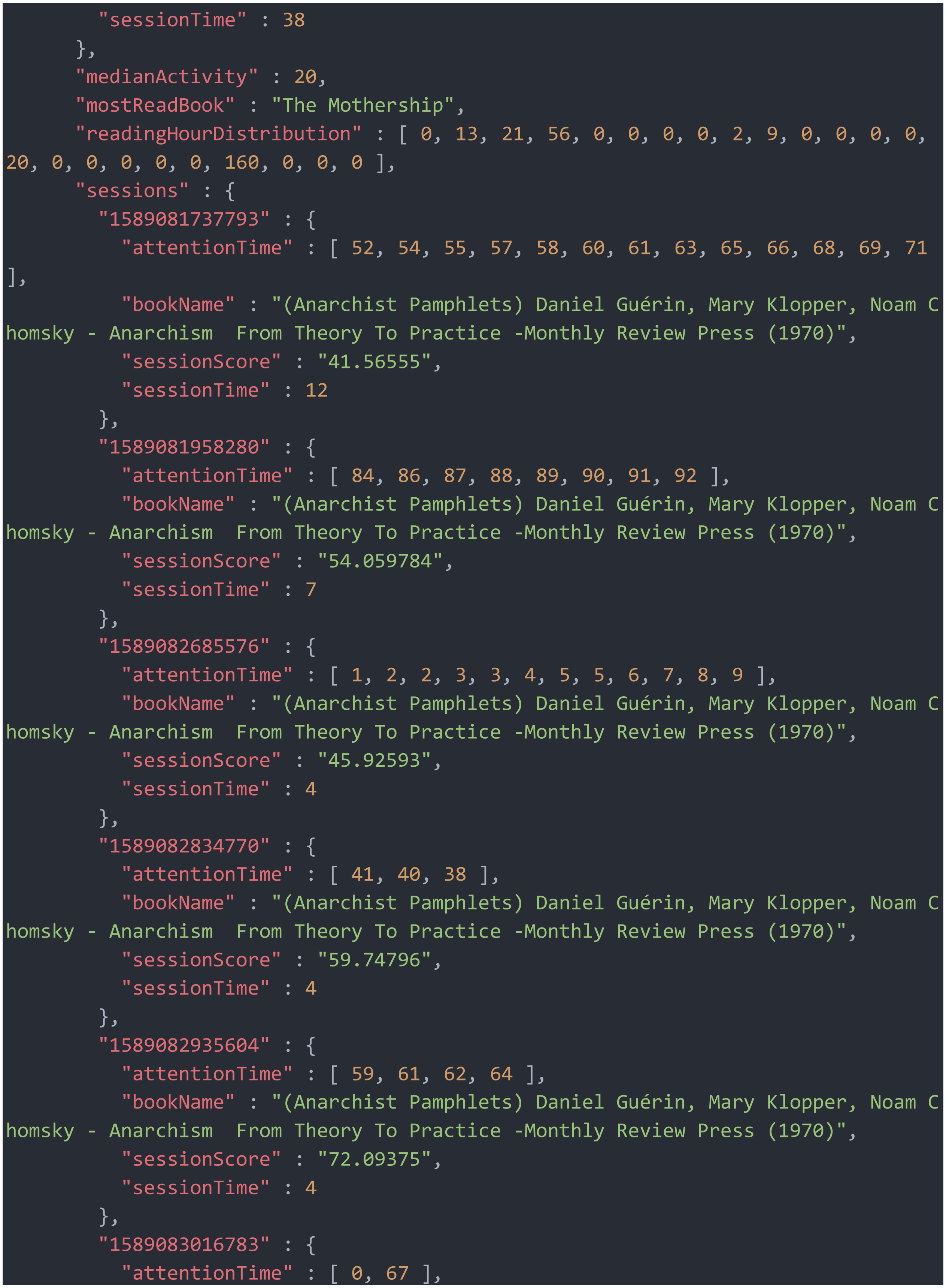
The schemas of data that are to be maintained for certain modules of the system:

```
{
        {
                “$schema” : “ “ ,
                “$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”
```

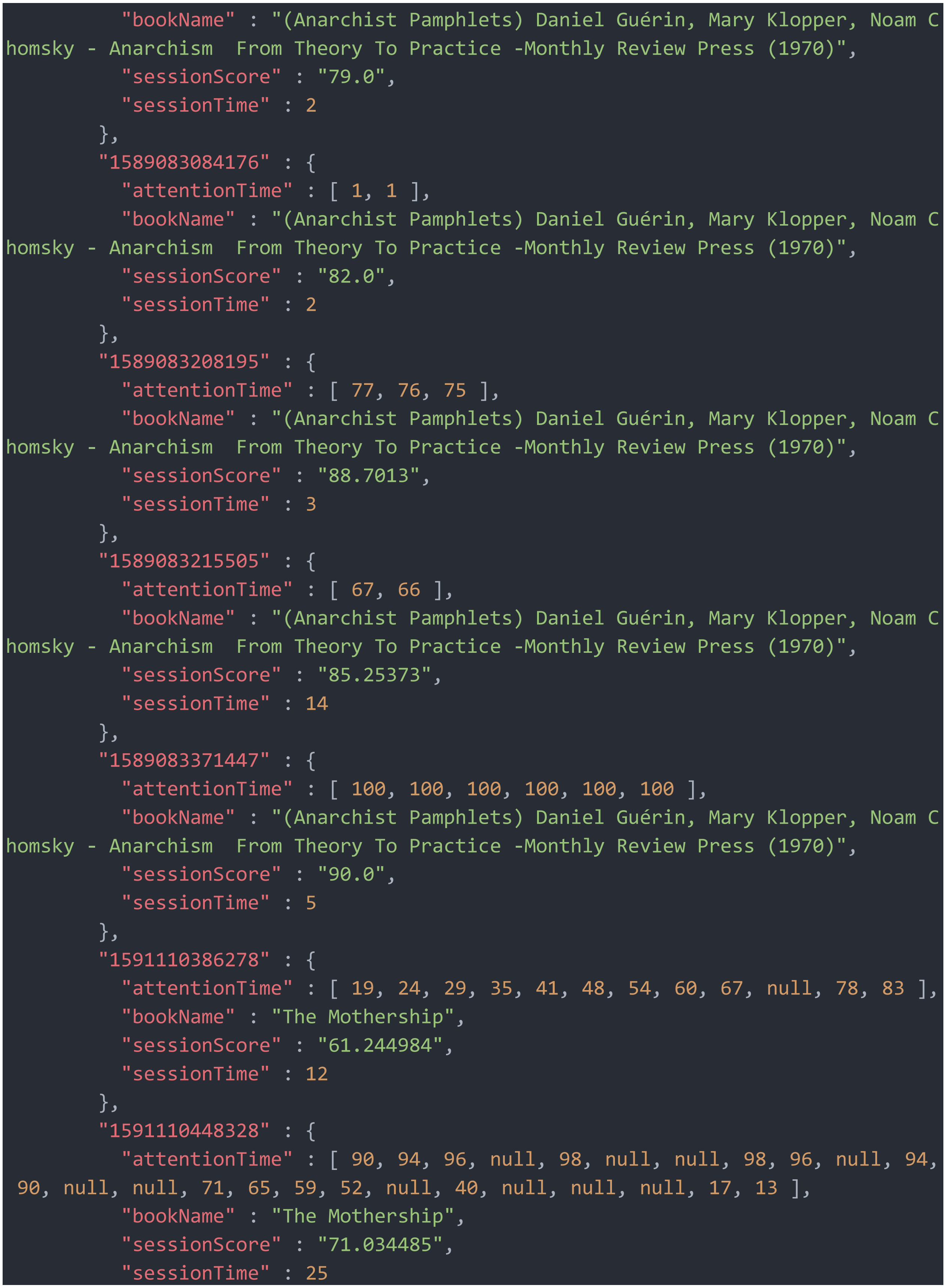
```
            }

    },
    “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” ]
    }
}
```



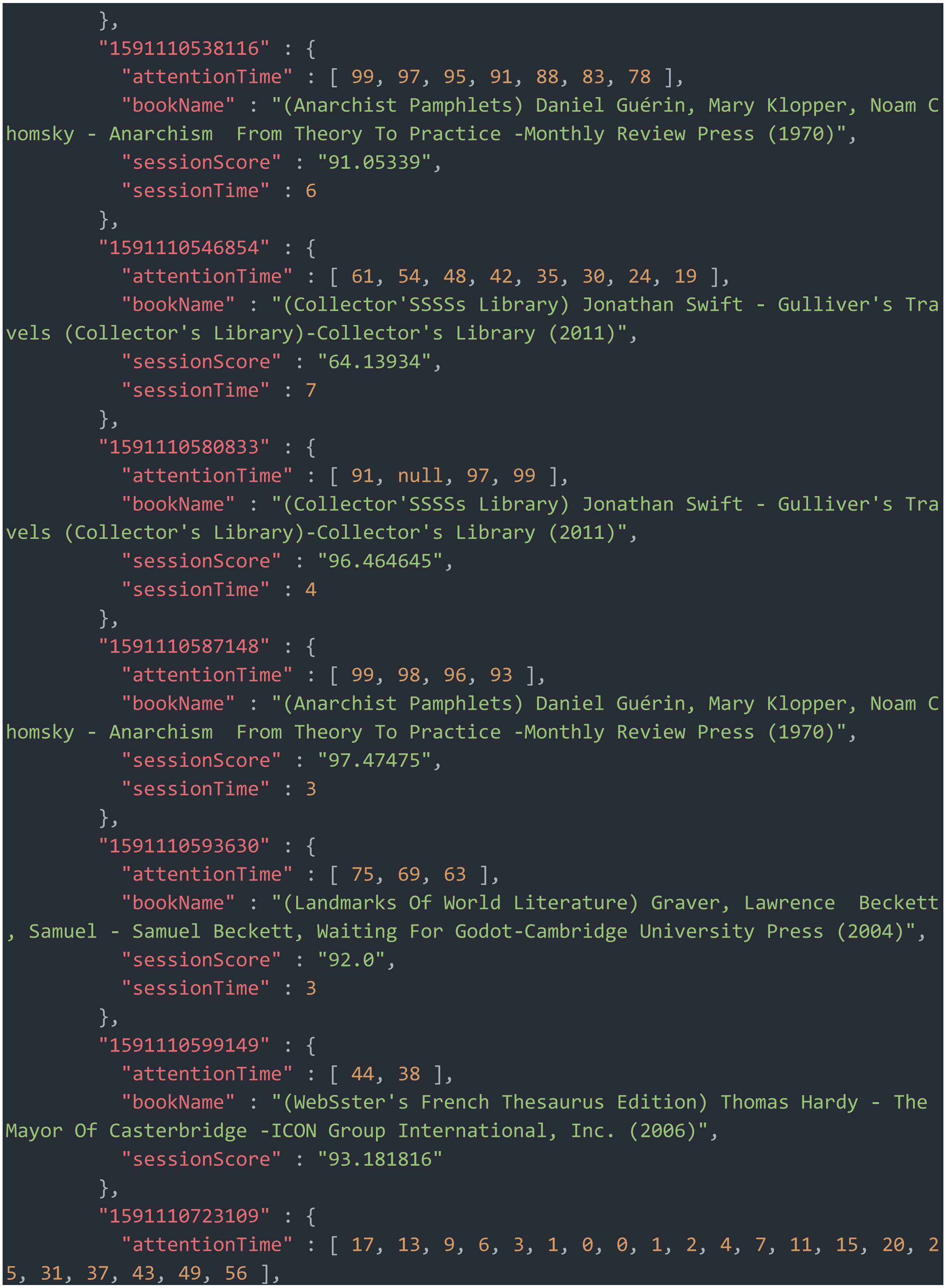
```
{
  "hostOptions" : {
    "hostname" : "192.168.100.8",
    "port" : 8001
  },
  "userReports" : {
    "9kz4xiUQUGPeiFXAVhMWDfiphua2" : {
      "bookRepertoire" : {
        "(Anarchist Pamphlets) Daniel Guérin, Mary Klopper, Noam Chomsky - Anarch
ism  From Theory To Practice -Monthly Review Press (1970)" : 24,
        "(Collector'SSSSs Library) Jonathan Swift - Gulliver's Travels (Collector
's Library)-Collector's Library (2011)" : 12,
        "(Landmarks Of World Literature) Graver, Lawrence  Beckett, Samuel - Samu
el Beckett, Waiting For Godot-Cambridge University Press (2004)" : 3,
        "The Mothership" : 261
      },
      "highestActivity" : 20,
      "lastRead" : {
        "bookName" : "The Mothership",
```



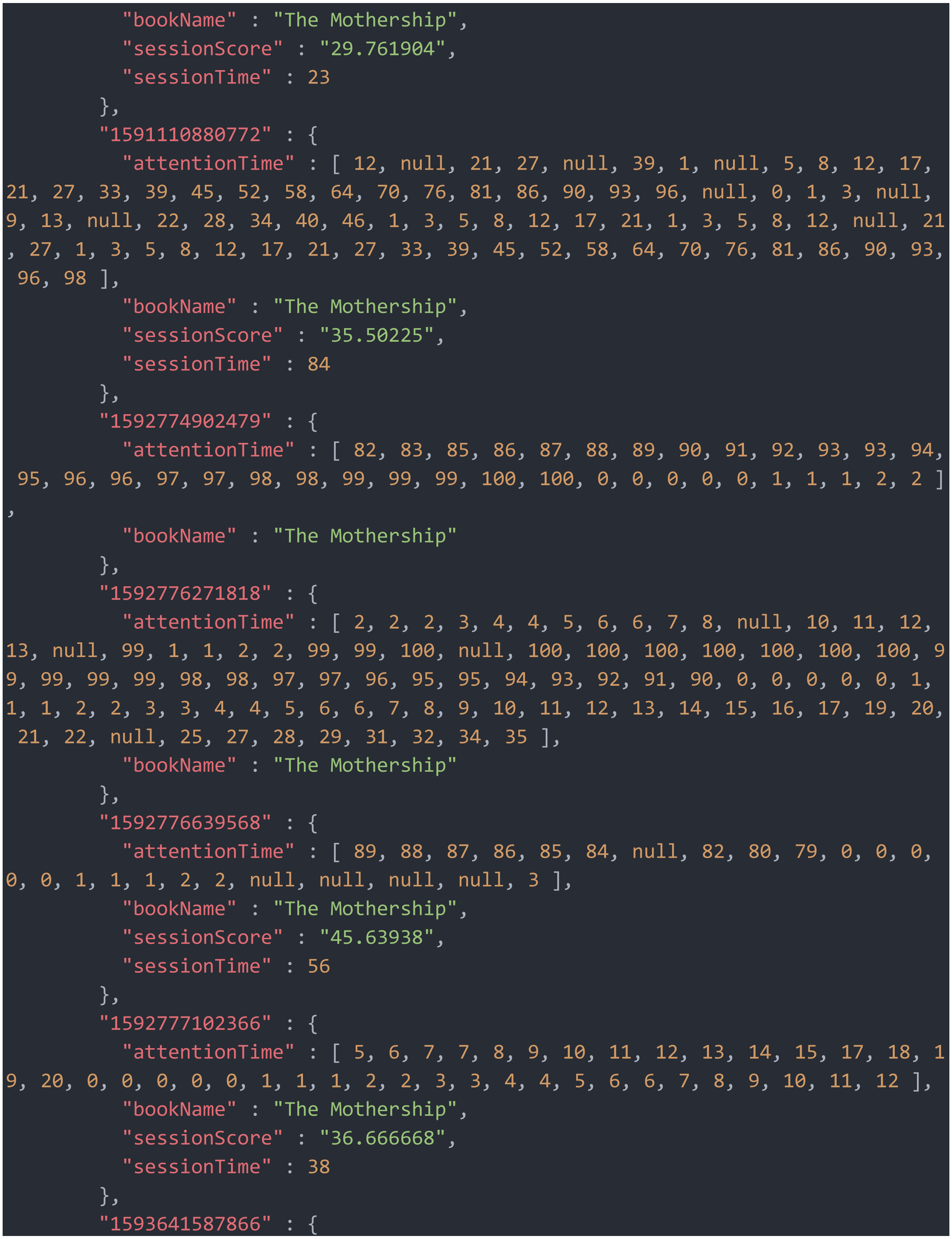
```
        "sessionTime" : 38
      },
      "medianActivity" : 20,
      "mostReadBook" : "The Mothership",
      "readingHourDistribution" : [ 0, 13, 21, 56, 0, 0, 0, 0, 2, 9, 0, 0, 0, 0,
20, 0, 0, 0, 0, 0, 160, 0, 0, 0 ],
      "sessions" : {
        "1589081737793" : {
          "attentionTime" : [ 52, 54, 55, 57, 58, 60, 61, 63, 65, 66, 68, 69, 71
],
          "bookName" : "(Anarchist Pamphlets) Daniel Guérin, Mary Klopper, Noam C
homsky - Anarchism  From Theory To Practice -Monthly Review Press (1970)",
          "sessionScore" : "41.56555",
          "sessionTime" : 12
        },
        "1589081958280" : {
          "attentionTime" : [ 84, 86, 87, 88, 89, 90, 91, 92 ],
          "bookName" : "(Anarchist Pamphlets) Daniel Guérin, Mary Klopper, Noam C
homsky - Anarchism  From Theory To Practice -Monthly Review Press (1970)",
          "sessionScore" : "54.059784",
          "sessionTime" : 7
        },
        "1589082685576" : {
          "attentionTime" : [ 1, 2, 2, 3, 3, 4, 5, 5, 6, 7, 8, 9 ],
          "bookName" : "(Anarchist Pamphlets) Daniel Guérin, Mary Klopper, Noam C
homsky - Anarchism  From Theory To Practice -Monthly Review Press (1970)",
          "sessionScore" : "45.92593",
          "sessionTime" : 4
        },
        "1589082834770" : {
          "attentionTime" : [ 41, 40, 38 ],
          "bookName" : "(Anarchist Pamphlets) Daniel Guérin, Mary Klopper, Noam C
homsky - Anarchism  From Theory To Practice -Monthly Review Press (1970)",
          "sessionScore" : "59.74796",
          "sessionTime" : 4
        },
        "1589082935604" : {
          "attentionTime" : [ 59, 61, 62, 64 ],
          "bookName" : "(Anarchist Pamphlets) Daniel Guérin, Mary Klopper, Noam C
homsky - Anarchism  From Theory To Practice -Monthly Review Press (1970)",
          "sessionScore" : "72.09375",
          "sessionTime" : 4
        },
        "1589083016783" : {
          "attentionTime" : [ 0, 67 ],
```



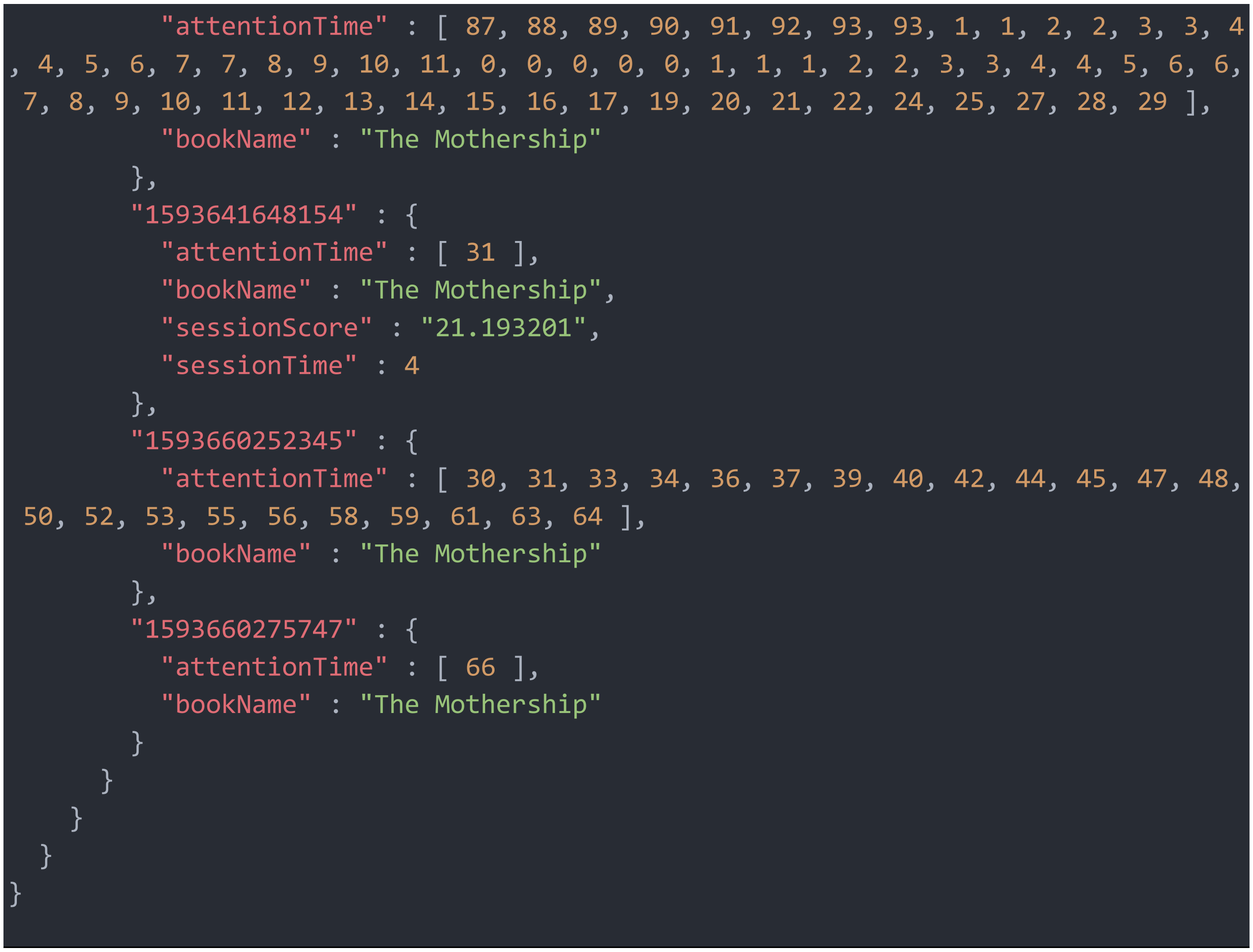
```
          "bookName" : "(Anarchist Pamphlets) Daniel Guérin, Mary Klopper, Noam C
homsky - Anarchism  From Theory To Practice -Monthly Review Press (1970)",
          "sessionScore" : "79.0",
          "sessionTime" : 2
        },
        "1589083084176" : {
          "attentionTime" : [ 1, 1 ],
          "bookName" : "(Anarchist Pamphlets) Daniel Guérin, Mary Klopper, Noam C
homsky - Anarchism  From Theory To Practice -Monthly Review Press (1970)",
          "sessionScore" : "82.0",
          "sessionTime" : 2
        },
        "1589083208195" : {
          "attentionTime" : [ 77, 76, 75 ],
          "bookName" : "(Anarchist Pamphlets) Daniel Guérin, Mary Klopper, Noam C
homsky - Anarchism  From Theory To Practice -Monthly Review Press (1970)",
          "sessionScore" : "88.7013",
          "sessionTime" : 3
        },
        "1589083215505" : {
          "attentionTime" : [ 67, 66 ],
          "bookName" : "(Anarchist Pamphlets) Daniel Guérin, Mary Klopper, Noam C
homsky - Anarchism  From Theory To Practice -Monthly Review Press (1970)",
          "sessionScore" : "85.25373",
          "sessionTime" : 14
        },
        "1589083371447" : {
          "attentionTime" : [ 100, 100, 100, 100, 100, 100 ],
          "bookName" : "(Anarchist Pamphlets) Daniel Guérin, Mary Klopper, Noam C
homsky - Anarchism  From Theory To Practice -Monthly Review Press (1970)",
          "sessionScore" : "90.0",
          "sessionTime" : 5
        },
        "1591110386278" : {
          "attentionTime" : [ 19, 24, 29, 35, 41, 48, 54, 60, 67, null, 78, 83 ],
          "bookName" : "The Mothership",
          "sessionScore" : "61.244984",
          "sessionTime" : 12
        },
        "1591110448328" : {
          "attentionTime" : [ 90, 94, 96, null, 98, null, null, 98, 96, null, 94,
 90, null, null, 71, 65, 59, 52, null, 40, null, null, null, 17, 13 ],
          "bookName" : "The Mothership",
          "sessionScore" : "71.034485",
          "sessionTime" : 25
```



```
        },
        "1591110538116" : {
          "attentionTime" : [ 99, 97, 95, 91, 88, 83, 78 ],
          "bookName" : "(Anarchist Pamphlets) Daniel Guérin, Mary Klopper, Noam C
homsky - Anarchism  From Theory To Practice -Monthly Review Press (1970)",
          "sessionScore" : "91.05339",
          "sessionTime" : 6
        },
        "1591110546854" : {
          "attentionTime" : [ 61, 54, 48, 42, 35, 30, 24, 19 ],
          "bookName" : "(Collector'SSSSs Library) Jonathan Swift - Gulliver's Tra
vels (Collector's Library)-Collector's Library (2011)",
          "sessionScore" : "64.13934",
          "sessionTime" : 7
        },
        "1591110580833" : {
          "attentionTime" : [ 91, null, 97, 99 ],
          "bookName" : "(Collector'SSSSs Library) Jonathan Swift - Gulliver's Tra
vels (Collector's Library)-Collector's Library (2011)",
          "sessionScore" : "96.464645",
          "sessionTime" : 4
        },
        "1591110587148" : {
          "attentionTime" : [ 99, 98, 96, 93 ],
          "bookName" : "(Anarchist Pamphlets) Daniel Guérin, Mary Klopper, Noam C
homsky - Anarchism  From Theory To Practice -Monthly Review Press (1970)",
          "sessionScore" : "97.47475",
          "sessionTime" : 3
        },
        "1591110593630" : {
          "attentionTime" : [ 75, 69, 63 ],
          "bookName" : "(Landmarks Of World Literature) Graver, Lawrence  Beckett
, Samuel - Samuel Beckett, Waiting For Godot-Cambridge University Press (2004)",
          "sessionScore" : "92.0",
          "sessionTime" : 3
        },
        "1591110599149" : {
          "attentionTime" : [ 44, 38 ],
          "bookName" : "(WebSster's French Thesaurus Edition) Thomas Hardy - The
Mayor Of Casterbridge -ICON Group International, Inc. (2006)",
          "sessionScore" : "93.181816"
        },
        "1591110723109" : {
          "attentionTime" : [ 17, 13, 9, 6, 3, 1, 0, 0, 1, 2, 4, 7, 11, 15, 20, 2
5, 31, 37, 43, 49, 56 ],
```



```
          "bookName" : "The Mothership",
          "sessionScore" : "29.761904",
          "sessionTime" : 23
        },
        "1591110880772" : {
          "attentionTime" : [ 12, null, 21, 27, null, 39, 1, null, 5, 8, 12, 17,
21, 27, 33, 39, 45, 52, 58, 64, 70, 76, 81, 86, 90, 93, 96, null, 0, 1, 3, null,
9, 13, null, 22, 28, 34, 40, 46, 1, 3, 5, 8, 12, 17, 21, 1, 3, 5, 8, 12, null, 21
, 27, 1, 3, 5, 8, 12, 17, 21, 27, 33, 39, 45, 52, 58, 64, 70, 76, 81, 86, 90, 93,
 96, 98 ],
          "bookName" : "The Mothership",
          "sessionScore" : "35.50225",
          "sessionTime" : 84
        },
        "1592774902479" : {
          "attentionTime" : [ 82, 83, 85, 86, 87, 88, 89, 90, 91, 92, 93, 93, 94,
 95, 96, 96, 97, 97, 98, 98, 99, 99, 99, 100, 100, 0, 0, 0, 0, 0, 1, 1, 1, 2, 2 ]
,
          "bookName" : "The Mothership"
        },
        "1592776271818" : {
          "attentionTime" : [ 2, 2, 2, 3, 4, 4, 5, 6, 6, 7, 8, null, 10, 11, 12,
13, null, 99, 1, 1, 2, 2, 99, 99, 100, null, 100, 100, 100, 100, 100, 100, 100, 9
9, 99, 99, 99, 98, 98, 97, 97, 96, 95, 95, 94, 93, 92, 91, 90, 0, 0, 0, 0, 0, 1,
1, 1, 2, 2, 3, 3, 4, 4, 5, 6, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20,
 21, 22, null, 25, 27, 28, 29, 31, 32, 34, 35 ],
          "bookName" : "The Mothership"
        },
        "1592776639568" : {
          "attentionTime" : [ 89, 88, 87, 86, 85, 84, null, 82, 80, 79, 0, 0, 0,
0, 0, 1, 1, 1, 2, 2, null, null, null, null, 3 ],
          "bookName" : "The Mothership",
          "sessionScore" : "45.63938",
          "sessionTime" : 56
        },
        "1592777102366" : {
          "attentionTime" : [ 5, 6, 7, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 1
9, 20, 0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 3, 3, 4, 4, 5, 6, 6, 7, 8, 9, 10, 11, 12 ],
          "bookName" : "The Mothership",
          "sessionScore" : "36.666668",
          "sessionTime" : 38
        },
        "1593641587866" : {
```



```
          "attentionTime" : [ 87, 88, 89, 90, 91, 92, 93, 93, 1, 1, 2, 2, 3, 3, 4
, 4, 5, 6, 7, 7, 8, 9, 10, 11, 0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 3, 3, 4, 4, 5, 6, 6,
 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 24, 25, 27, 28, 29 ],
          "bookName" : "The Mothership"
        },
        "1593641648154" : {
          "attentionTime" : [ 31 ],
          "bookName" : "The Mothership",
          "sessionScore" : "21.193201",
          "sessionTime" : 4
        },
        "1593660252345" : {
          "attentionTime" : [ 30, 31, 33, 34, 36, 37, 39, 40, 42, 44, 45, 47, 48,
 50, 52, 53, 55, 56, 58, 59, 61, 63, 64 ],
          "bookName" : "The Mothership"
        },
        "1593660275747" : {
          "attentionTime" : [ 66 ],
          "bookName" : "The Mothership"
        }
      }
    }
  }
}
```

4.4 Data dictionary

The data dictionary for the schemas above:

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	“ “

4.5 Process Flow/Representation

Refer to Data Representation

4.6 Design Models

The use case models of Atom:

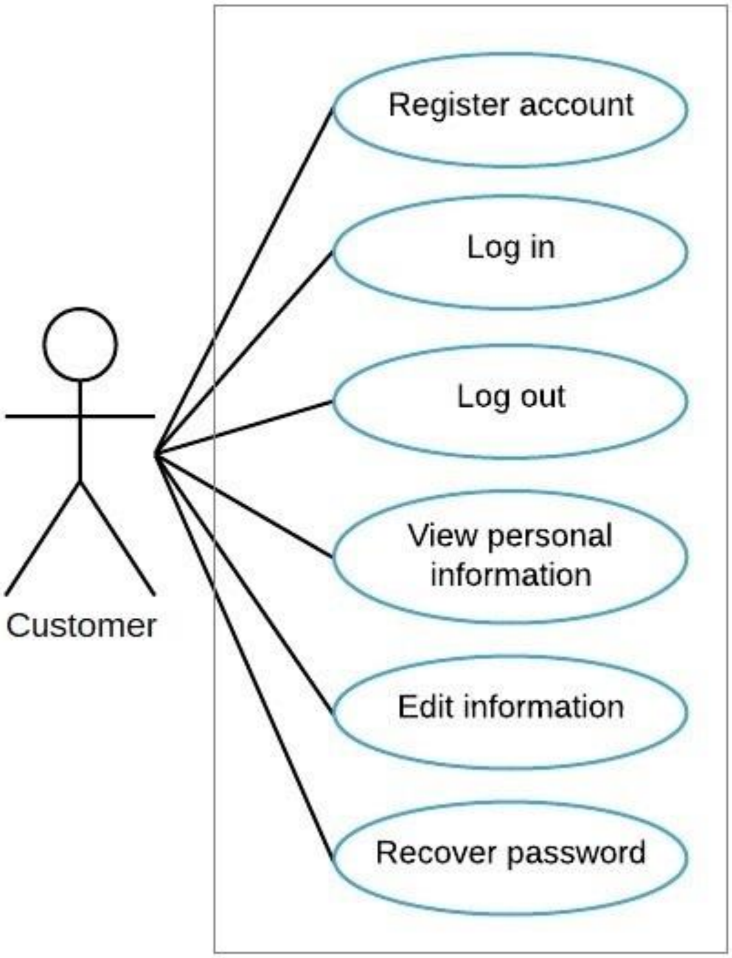


Figure 9 Use case module for Account Handling module

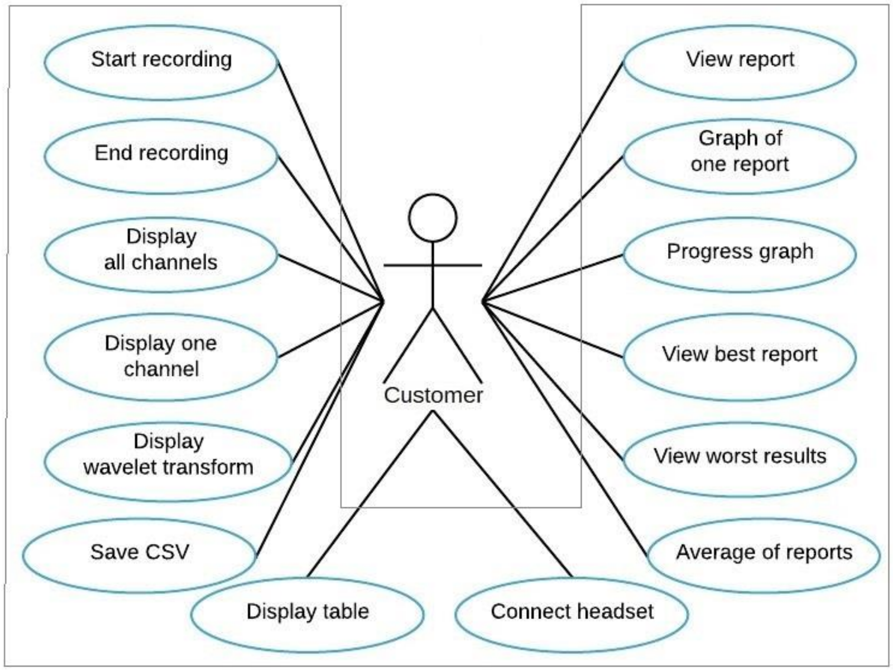


Figure 10 Use case model for User Analytics and Statistics module

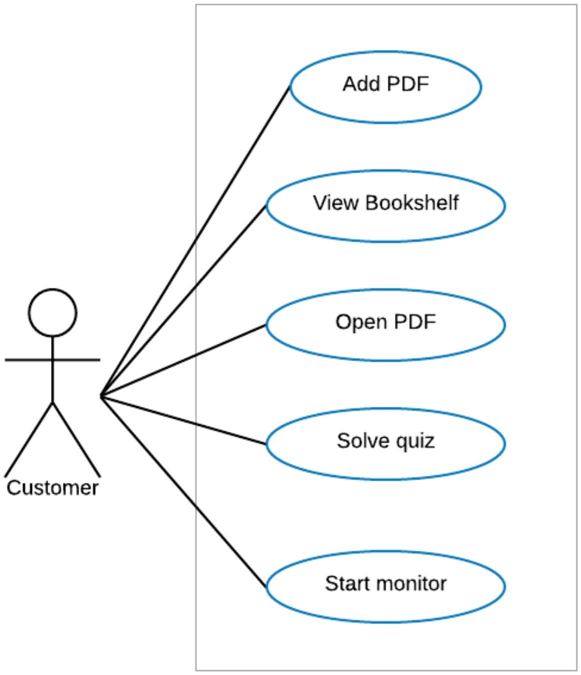


Figure 11 Use case model for Specialized Control Training module

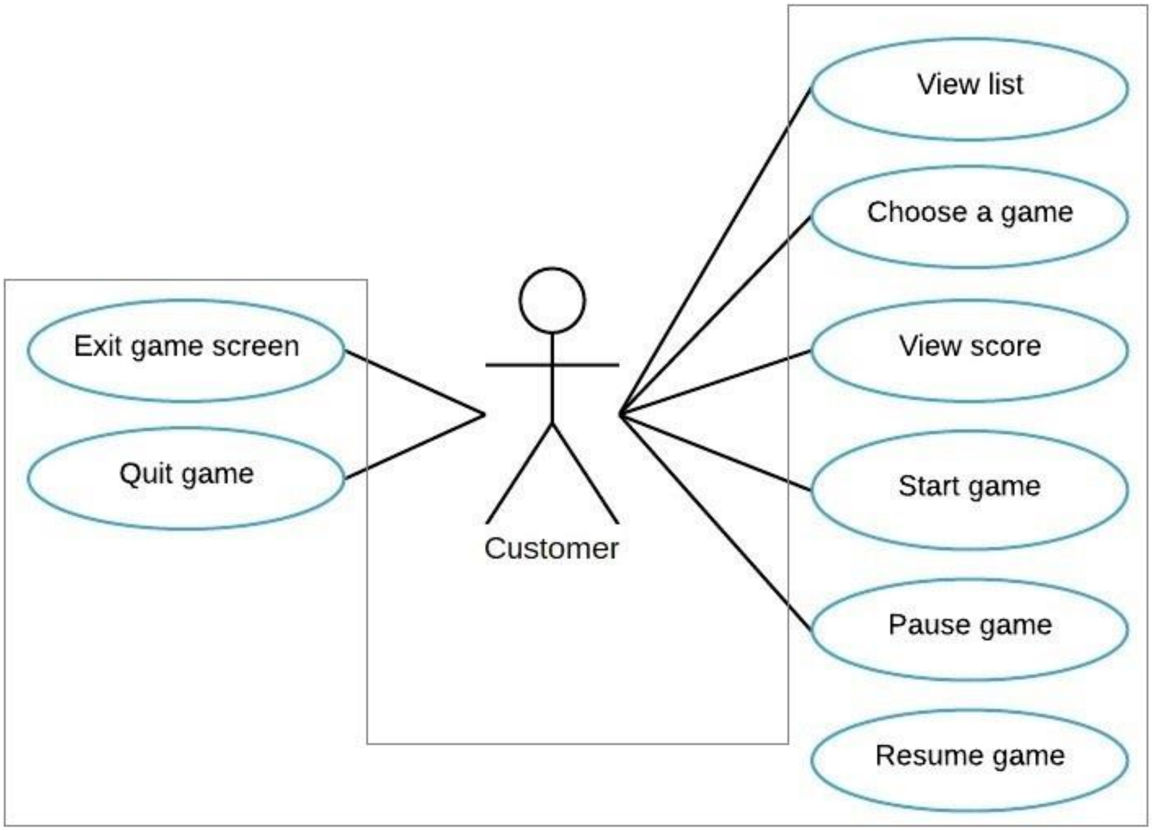


Figure 12 Use case model for Entertainment Incentivized Training module

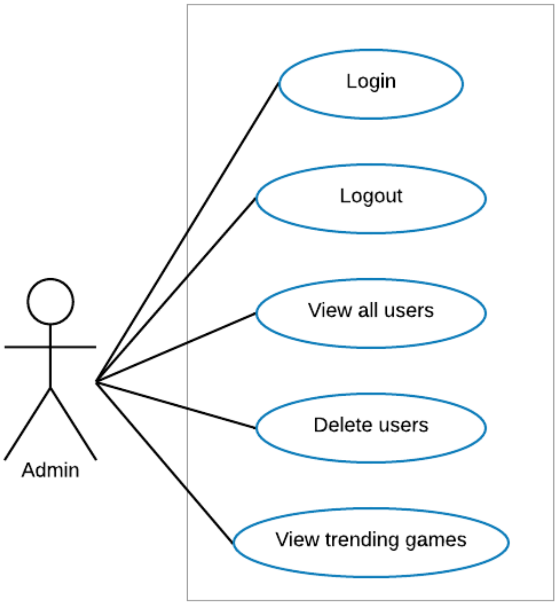


Figure 13 Use case model for Admin functionality

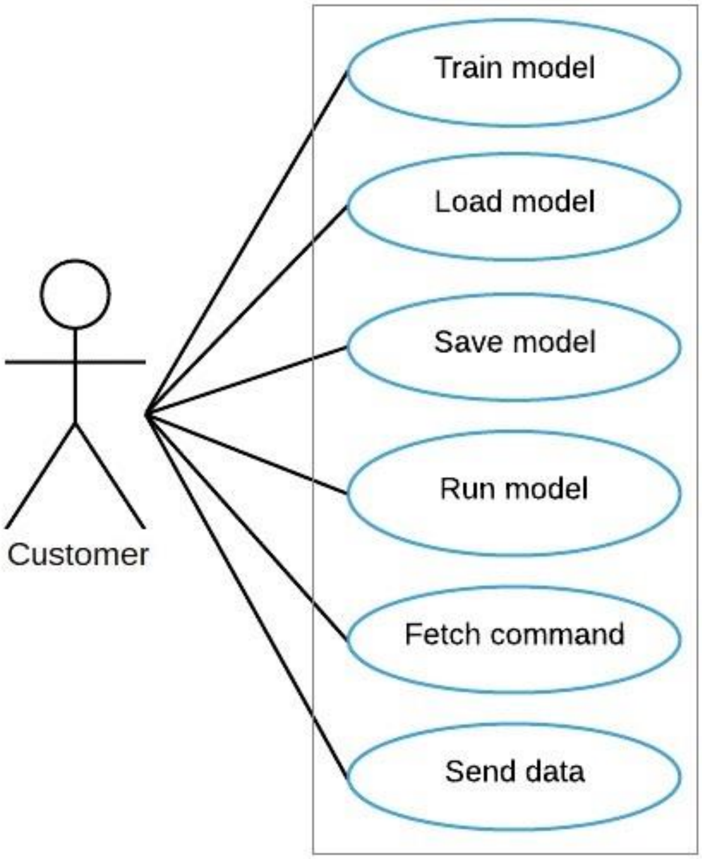


Figure 14 Use case model for EEG Feature Extraction module

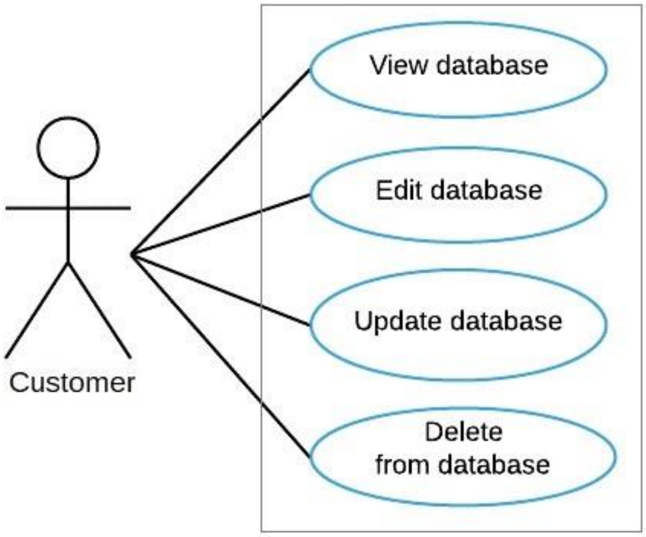
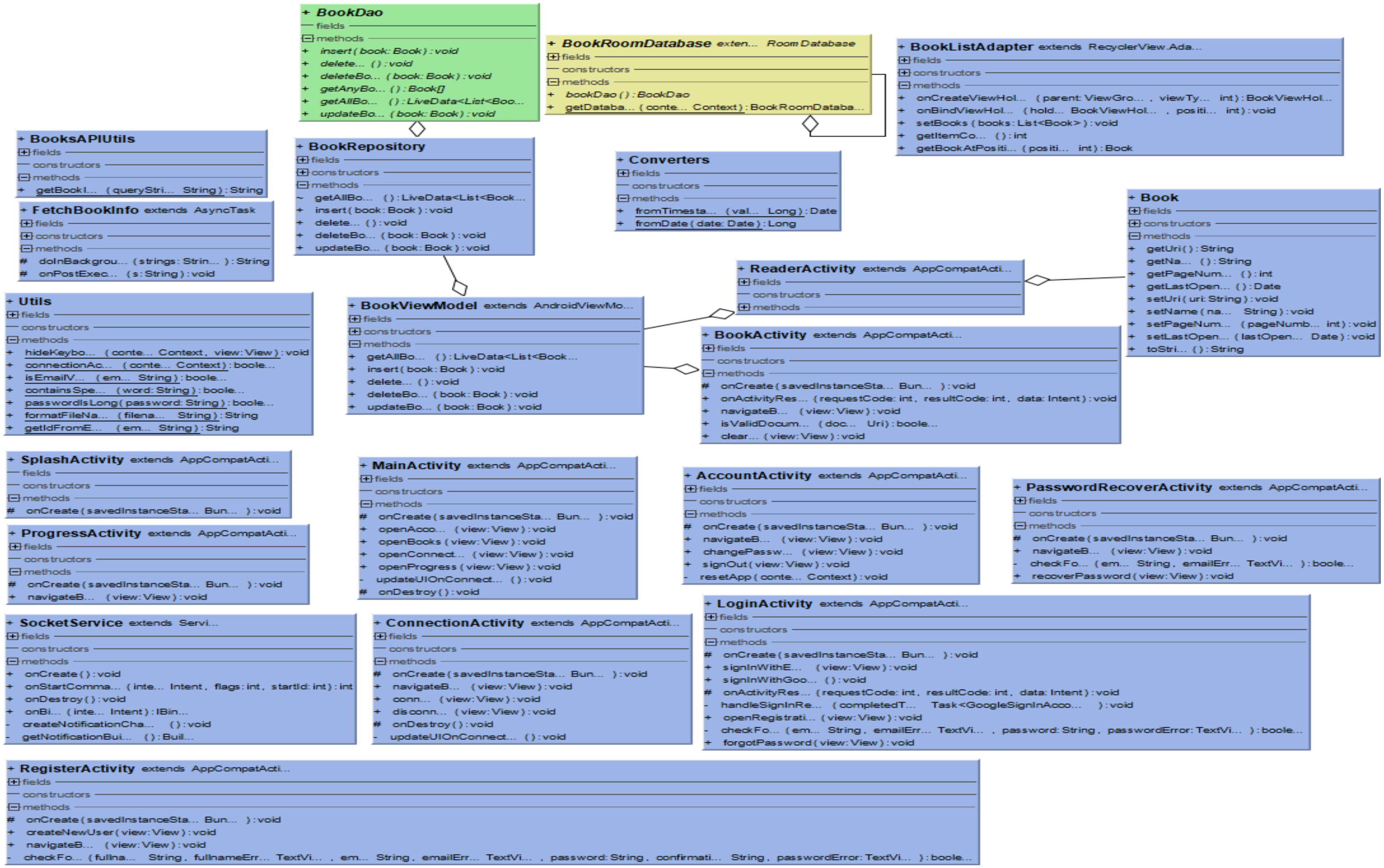


Figure 15 Use case model for Database Handling module

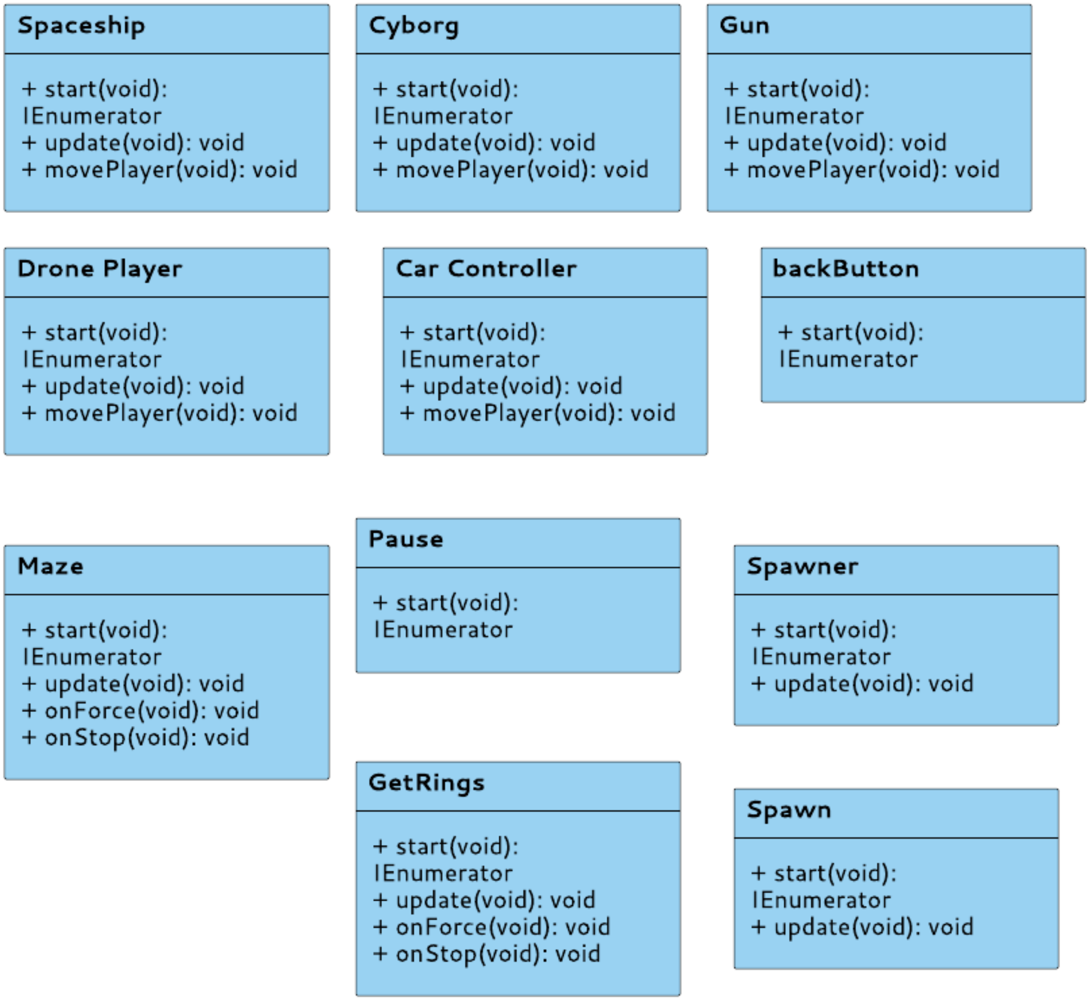
4.6.1 Structural Diagrams

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.

4.6.1.1 Class diagram


+ BookDao
fields
methods
+ insert (book: Book) : void
+ delete... ( ) : void
+ deleteBo... (book: Book) : void
+ getAnyBo... ( ) : Book[]
+ getAllBo... ( ) : LiveData<List<Boo...
+ updateBo... (book: Book) : void
+ BookRoomDatabase exten... Room Database
fields
constructors
methods
+ bookDao ( ) : BookDao
+ getDataba... (conte... Context) : BookRoomDataba...
+ BookListAdapter extends RecyclerView.Ada...
fields
constructors
methods
+ onCreateViewHol... (parent: ViewGro... , viewTy... int) : BookViewHol...
+ onBindViewHol... (hold... BookViewHol... , positi... int) : void
+ setBooks (books : List<Book>) : void
+ getItemCo... ( ) : int
+ getBookAtPositi... (positi... int) : Book
+ BooksAPIUtils
fields
constructors
methods
+ getBookI... (queryStri... String) : String
+ BookRepository
fields
constructors
methods
~ getAllBo... ( ) : LiveData<List<Book...
+ insert (book : Book) : void
+ delete... ( ) : void
+ deleteBo... (book : Book) : void
+ updateBo... (book : Book) : void
+ Converters
fields
constructors
methods
+ fromTimesta... (val... Long) : Date
+ fromDate (date: Date) : Long
+ Book
fields
constructors
methods
+ getUri ( ) : String
+ getNa... ( ) : String
+ getPageNum... ( ) : int
+ getLastOpen... ( ) : Date
+ setUri (uri: String) : void
+ setName (na... String) : void
+ setPageNum... (pageNumb... int) : void
+ setLastOpen... (lastOpen... Date) : void
+ toStri... ( ) : String
+ FetchBookInfo extends AsyncTask
fields
constructors
methods
# doInBackgrou... (strings : Strin...) : String
# onPostExec... (s : String) : void
+ ReaderActivity extends AppCompatActi...
fields
constructors
methods
+ Utils
fields
constructors
methods
+ hideKeybo... (conte... Context , view : View) : void
+ connectionAc... (conte... Context) : boole...
+ isEmailV... (em... String) : boole...
+ containsSpe... (word: String) : boole...
+ passwordIsLong (password: String) : boole...
+ formatFileNa... (filena... String) : String
+ getIdFromE... (em... String) : String
+ BookViewModel extends AndroidViewMo...
fields
constructors
methods
+ getAllBo... ( ) : LiveData<List<Book...
+ insert (book : Book) : void
+ delete... ( ) : void
+ deleteBo... (book : Book) : void
+ updateBo... (book : Book) : void
+ BookActivity extends AppCompatActi...
fields
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
+ onActivityRes... (requestCode: int , resultCode: int , data: Intent) : void
+ navigateB... (view : View) : void
+ isValidDocum... (doc... Uri) : boole...
+ clear... (view : View) : void
+ SplashActivity extends AppCompatActi...
fields
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
+ MainActivity extends AppCompatActi...
fields
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
+ openAcco... (view : View) : void
+ openBooks (view : View) : void
+ openConnect... (view : View) : void
+ openProgress (view : View) : void
- updateUIOnConnect... ( ) : void
# onDestroy ( ) : void
+ AccountActivity extends AppCompatActi...
fields
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
+ navigateB... (view : View) : void
+ changePassw... (view : View) : void
+ signOut (view : View) : void
- resetApp (conte... Context) : void
+ PasswordRecoverActivity extends AppCompatActi...
fields
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
+ navigateB... (view : View) : void
- checkFo... (em... String , emailErr... TextVi...) : boole...
+ recoverPassword (view : View) : void
+ ProgressActivity extends AppCompatActi...
fields
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
+ navigateB... (view : View) : void
+ SocketService extends Servi...
fields
constructors
methods
+ onCreate ( ) : void
+ onStartComma... (inte... Intent , flags : int , startId: int) : int
+ onDestroy ( ) : void
+ onBi... (inte... Intent) : IBin...
- createNotificationCha... ( ) : void
- getNotificationBui... ( ) : Buil...
+ ConnectionActivity extends AppCompatActi...
fields
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
+ navigateB... (view : View) : void
+ conn... (view : View) : void
+ disconn... (view : View) : void
# onDestroy ( ) : void
- updateUIOnConnect... ( ) : void
+ LoginActivity extends AppCompatActi...
fields
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
+ signInWithE... (view : View) : void
+ signInWithGoo... ( ) : void
# onActivityRes... (requestCode: int , resultCode: int , data: Intent) : void
- handleSignInRe... (completedT... Task<GoogleSignInAcco...) : void
+ openRegistrati... (view : View) : void
- checkFo... (em... String , emailErr... TextVi... , password: String , passwordError: TextVi...) : boole...
+ forgotPassword (view : View) : void
+ RegisterActivity extends AppCompatActi...
fields
constructors
methods
# onCreate (savedInstanceSta... Bun...) : void
+ createNewUser (view : View) : void
+ navigateB... (view : View) : void
- checkFo... (fullna... String , fullnameErr... TextVi... , em... String , emailErr... TextVi... , password: String , confirmati... String , passwordError: TextVi...) : boole...

Class Diagram 1 Atom Class Diagram – Android End



Class Diagram 2 Class Diagram for Atom - Unity End

4.6.2 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.

4.6.2.1 Activity diagram

The activity diagrams of modules that can be represented and correlated with the use cases of the system:

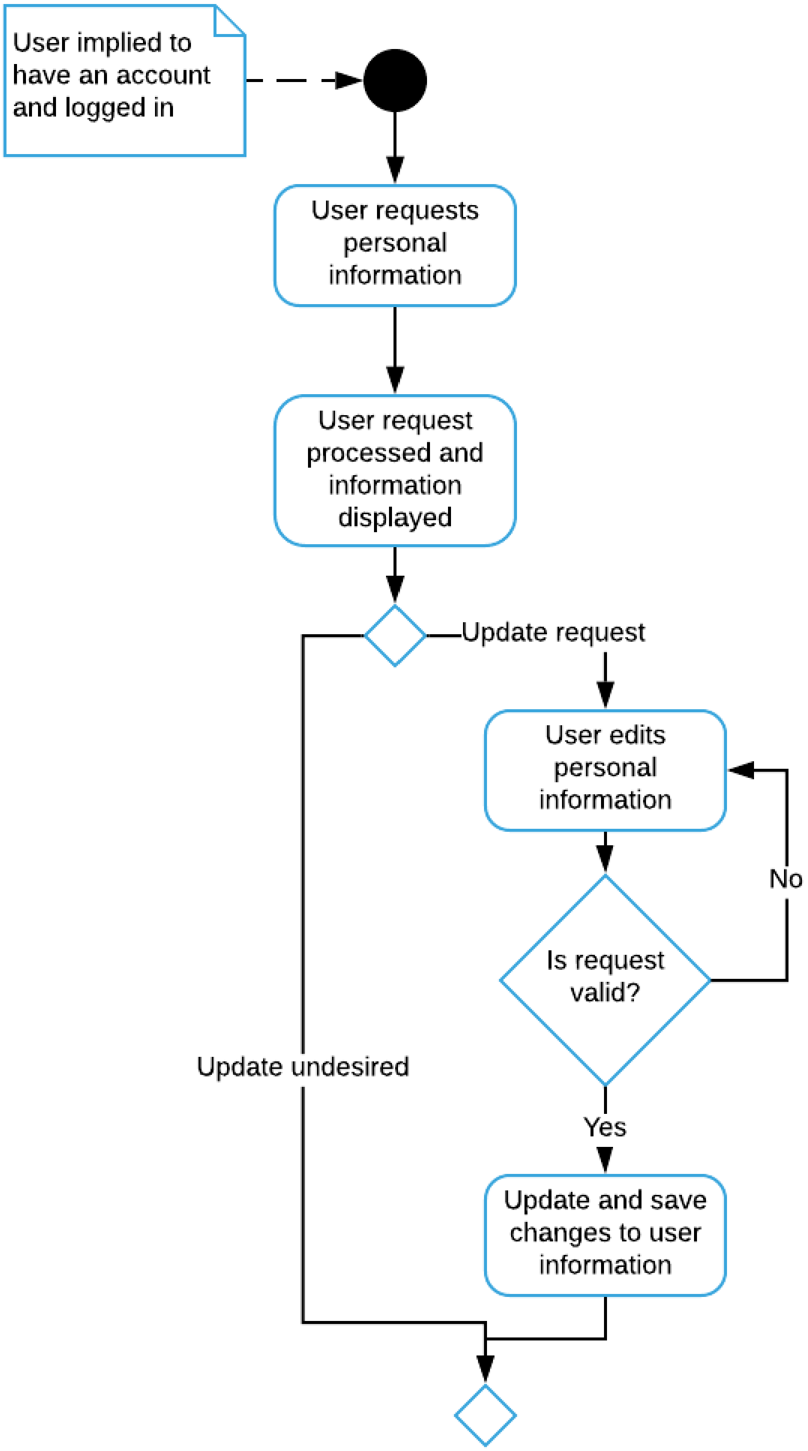

User implied to have an account and logged in
User requests personal information
User request processed and information displayed
Update request
User edits personal information
Is request valid?
No
Yes
Update and save changes to user information
Update undesired

Figure 16 Activity diagram for Account Handling module

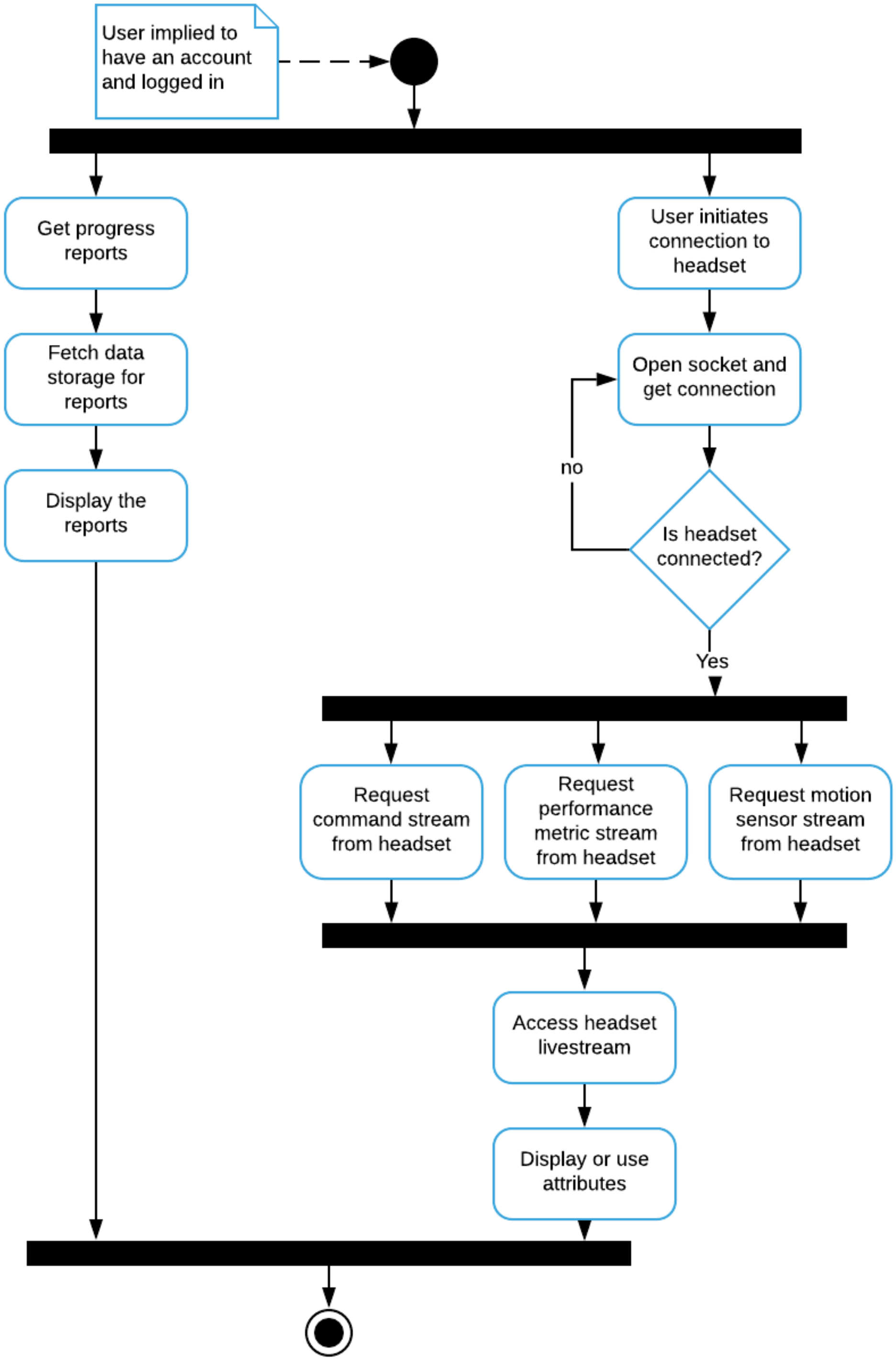

User implied to have an account and logged in
Get progress reports
Fetch data storage for reports
Display the reports
User initiates connection to headset
Open socket and get connection
no
Is headset connected?
Yes
Request command stream from headset
Request performance metric stream from headset
Request motion sensor stream from headset
Access headset livestream
Display or use attributes

Figure 17 Activity diagram for User Analytics and Statistics module

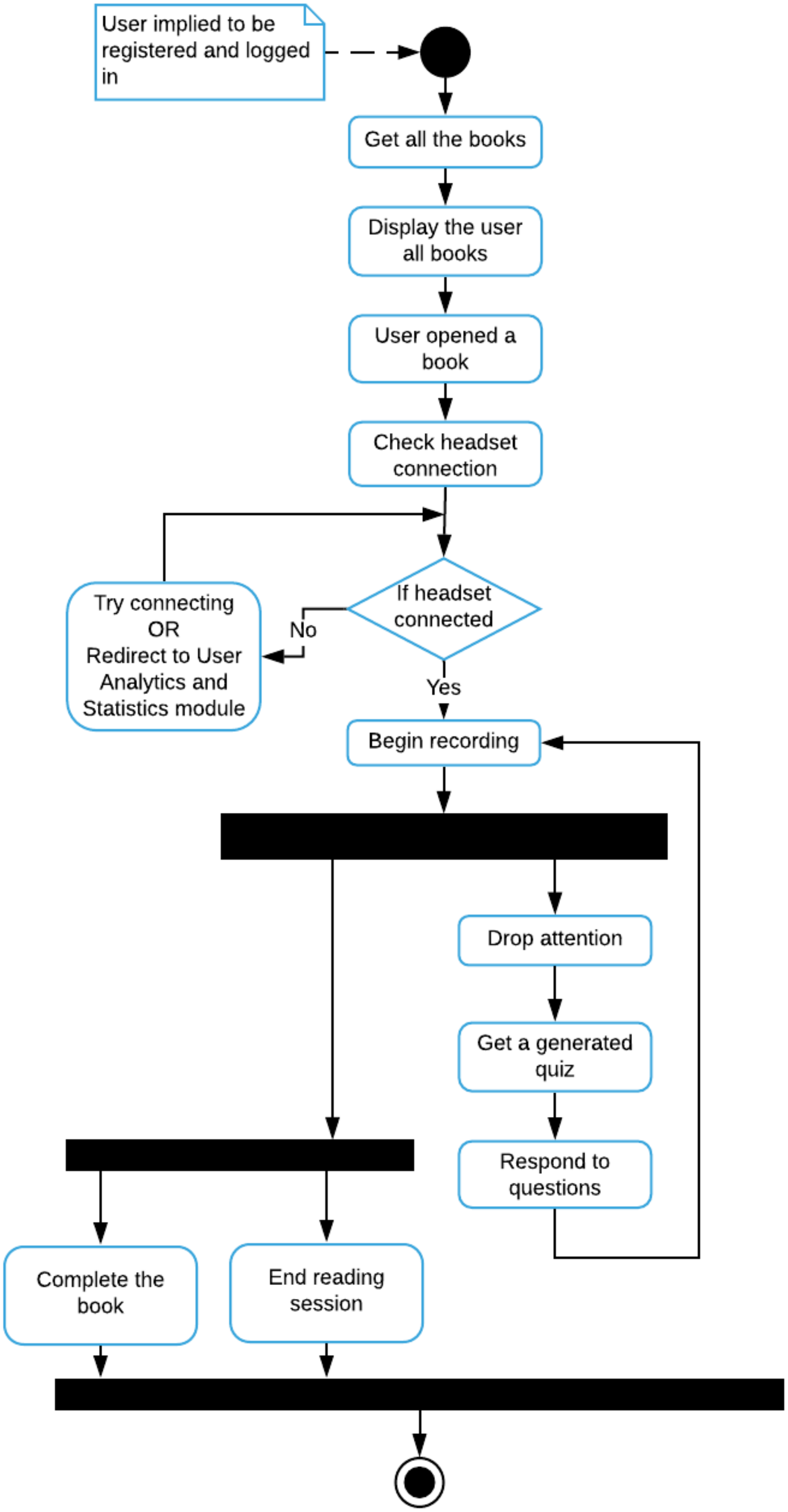


Figure 18 Activity diagram for Specialized Control Training module

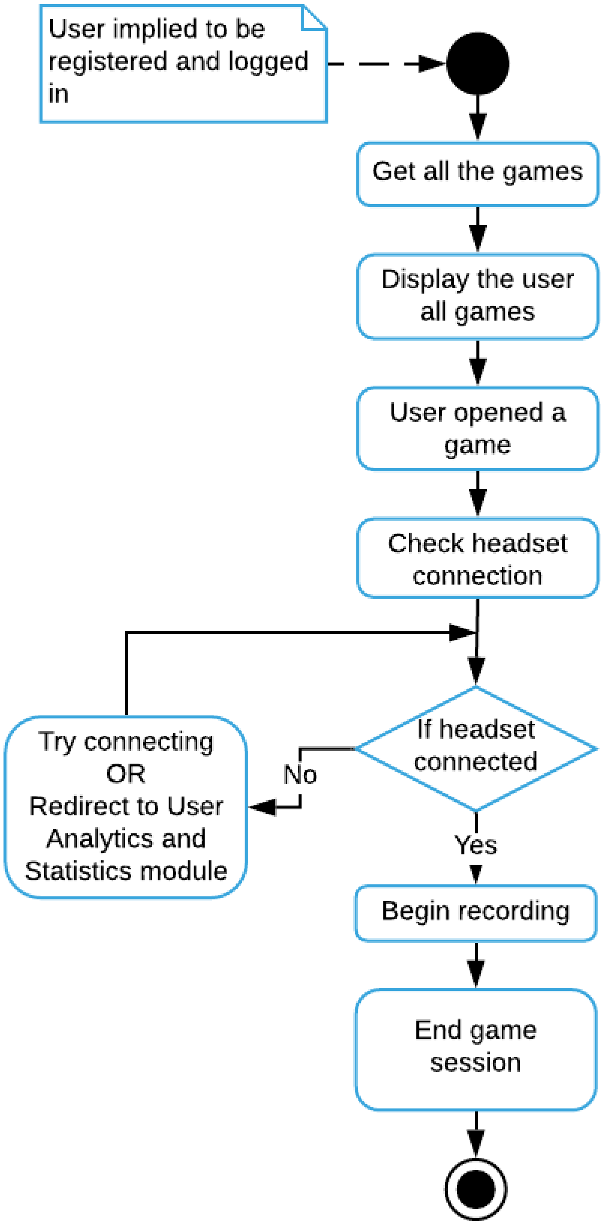

User implied to be registered and logged in
Get all the games
Display the user all games
User opened a game
Check headset connection
If headset connected
No
Try connecting OR Redirect to User Analytics and Statistics module
Yes
Begin recording
End game session

Figure 19 Activity diagram for Entertainment Incentivized Training module

4.6.2.2 Sequence diagram

The sequence diagrams of Atom:

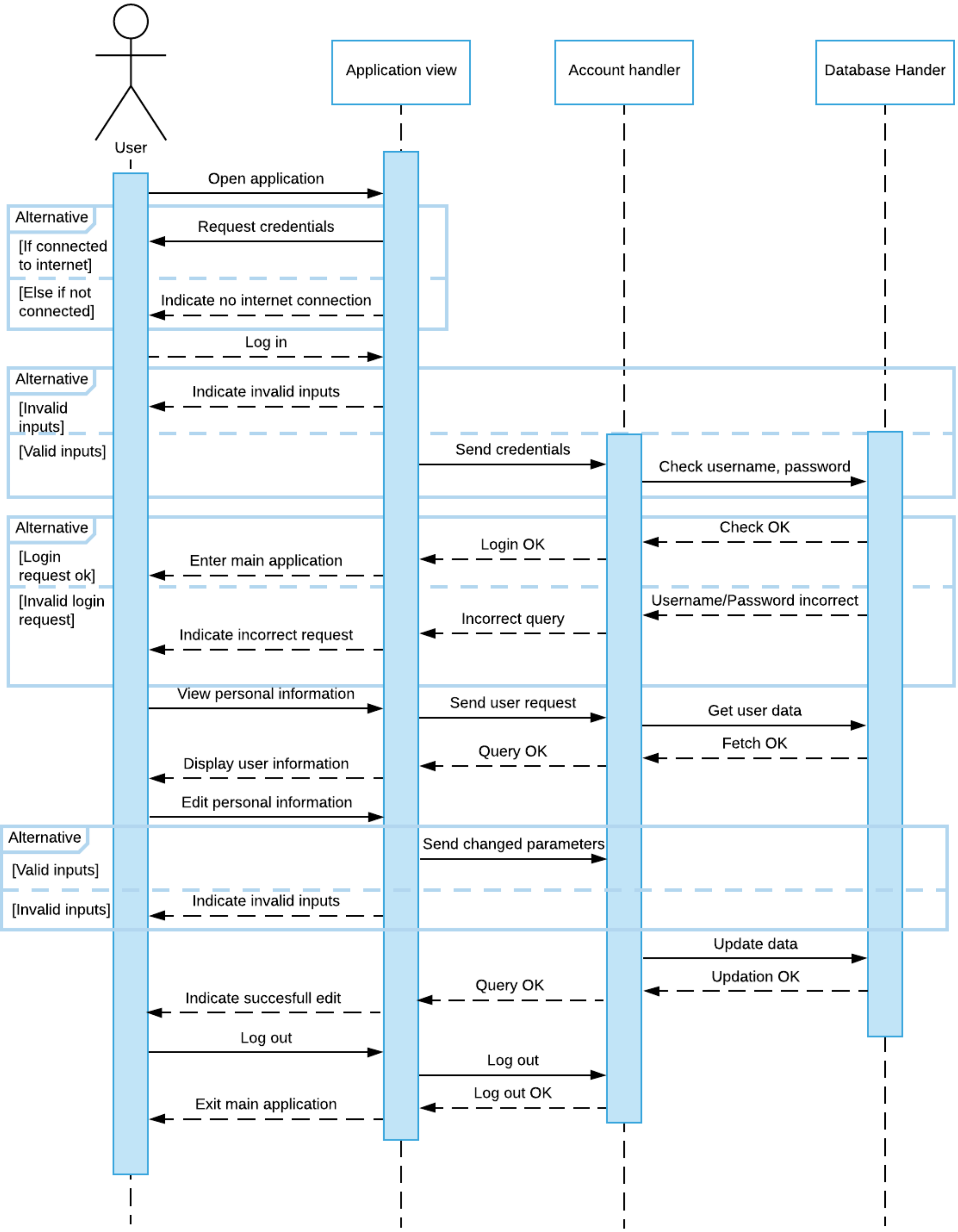


Figure 20 Sequence diagram for preliminary use cases

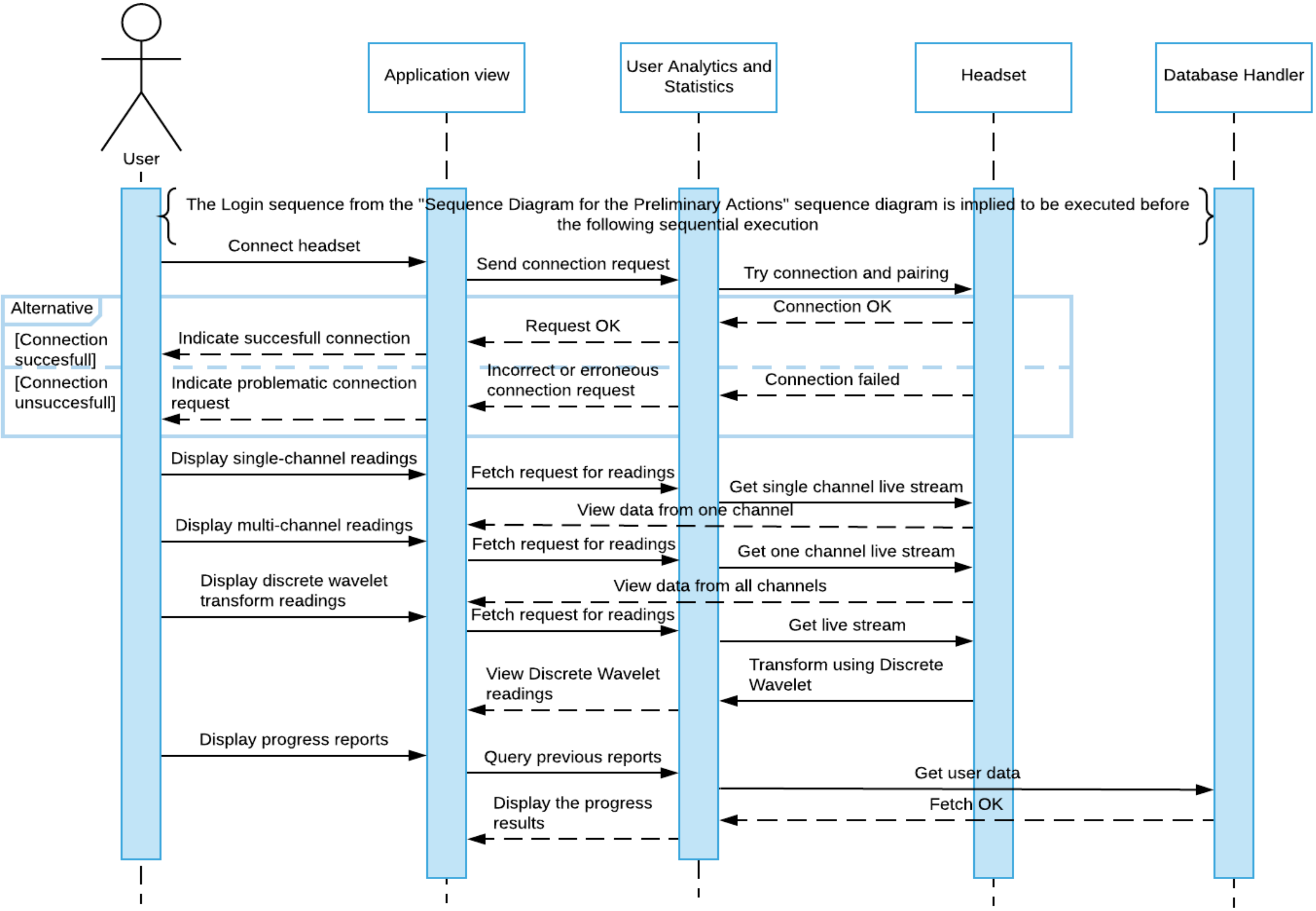

User
Application view
User Analytics and Statistics
Headset
Database Handler
The Login sequence from the "Sequence Diagram for the Preliminary Actions" sequence diagram is implied to be executed before the following sequential execution
Connect headset
Send connection request
Try connection and pairing
Alternative
[Connection succesfull]
Connection OK
Request OK
Indicate succesfull connection
[Connection unsuccesfull]
Incorrect or erroneous connection request
Connection failed
Indicate problematic connection request
Display single-channel readings
Fetch request for readings
Get single channel live stream
View data from one channel
Display multi-channel readings
Fetch request for readings
Get one channel live stream
Display discrete wavelet transform readings
View data from all channels
Fetch request for readings
Get live stream
Transform using Discrete Wavelet
View Discrete Wavelet readings
Display progress reports
Query previous reports
Get user data
Fetch OK
Display the progress results

Figure 21 Sequence diagram for User Analytics and Statistics use case

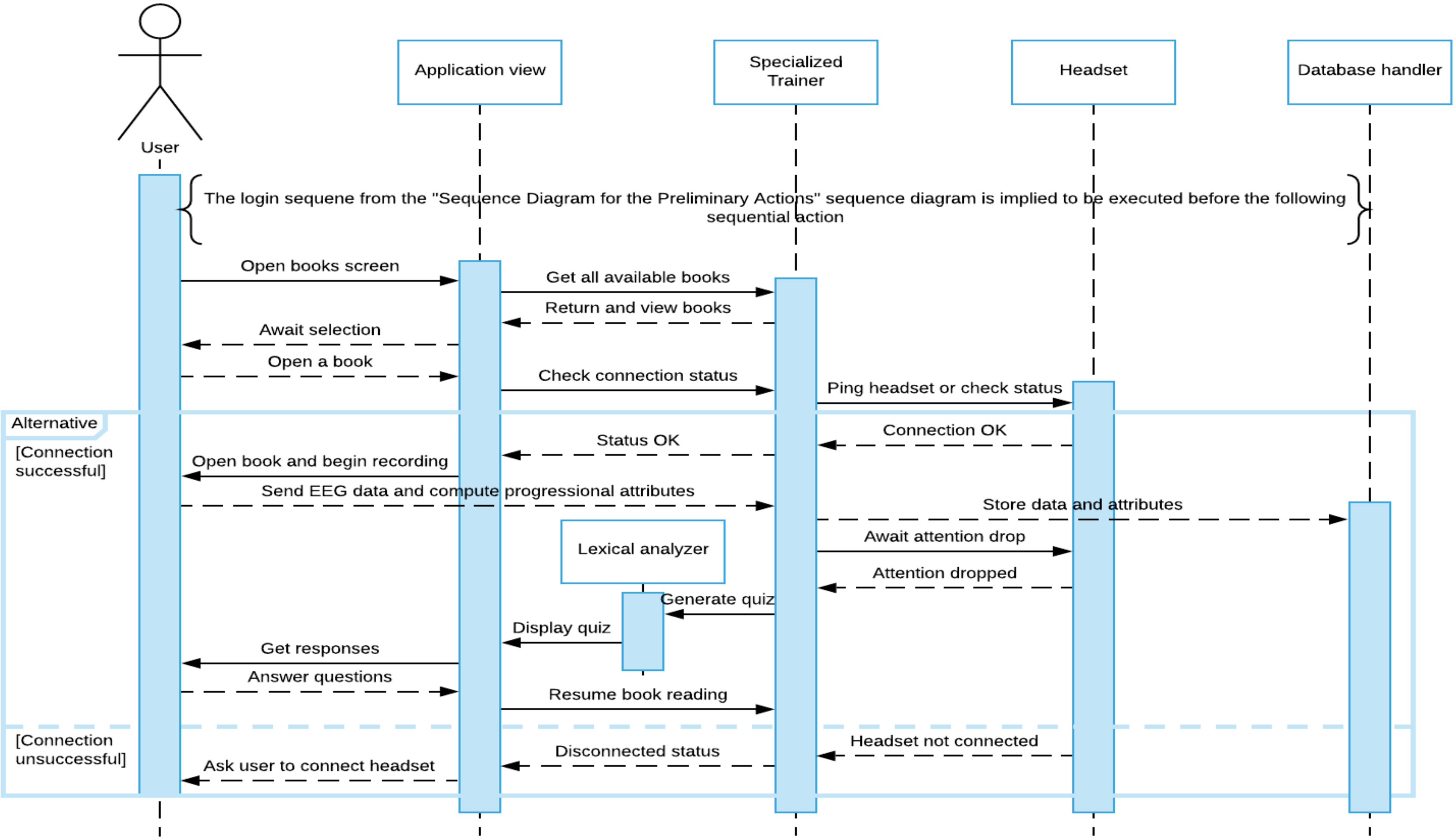


Figure 22 Sequence diagram for Specialized Control Training use cases

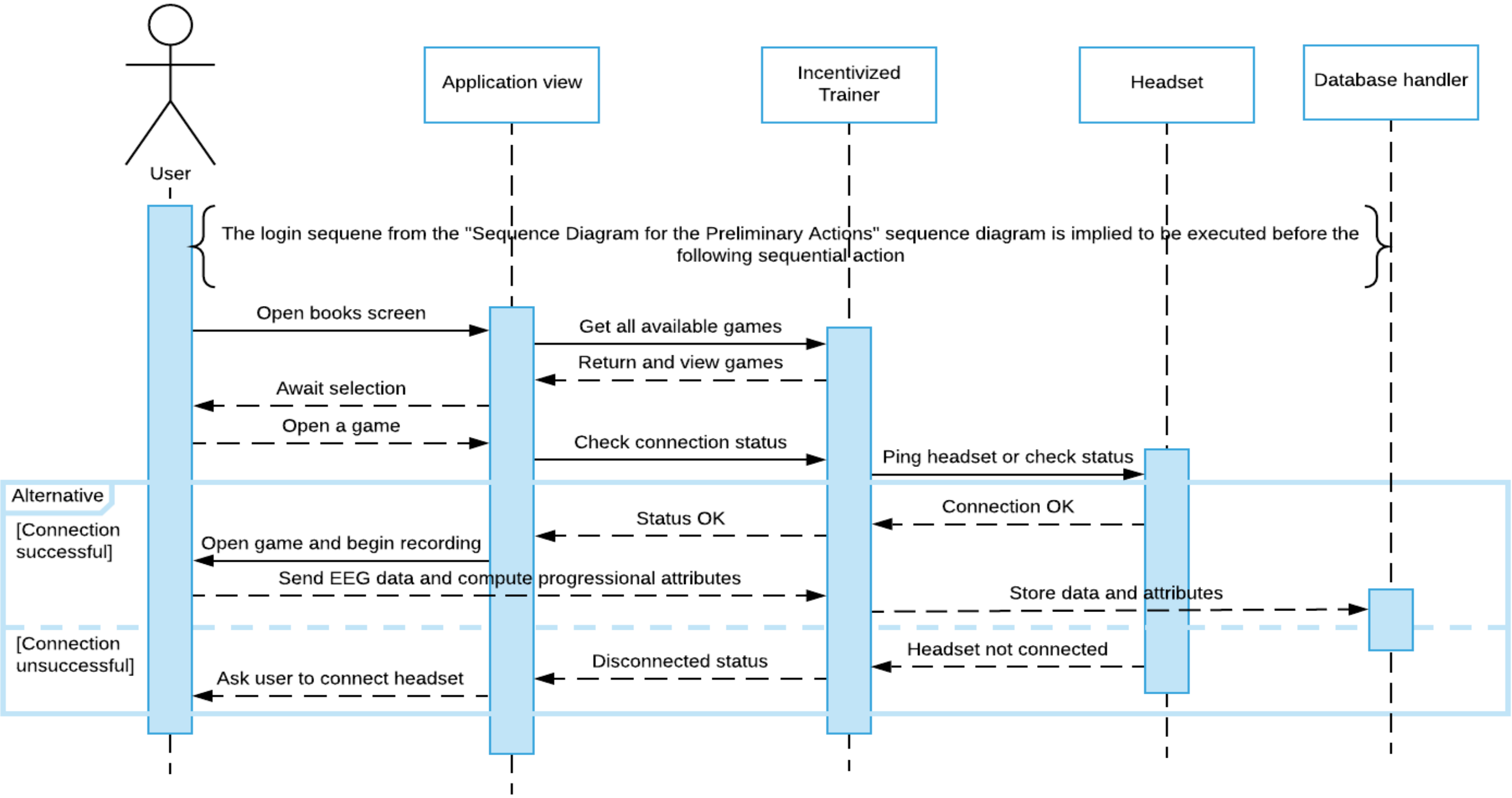

User
Application view
Incentivized Trainer
Headset
Database handler
The login sequene from the "Sequence Diagram for the Preliminary Actions" sequence diagram is implied to be executed before the following sequential action
Open books screen
Get all available games
Return and view games
Await selection
Open a game
Check connection status
Ping headset or check status
Alternative
[Connection successful]
Connection OK
Status OK
Open game and begin recording
Send EEG data and compute progressional attributes
Store data and attributes
[Connection unsuccessful]
Headset not connected
Disconnected status
Ask user to connect headset

Figure 23 Sequence diagram for Entertainment Incentivized Training use cases

5 Implementation

This section brings light to some of the implementation details at the lowest level:

5.1 Algorithm

The major algorithms in form of pseudocode:
<u>KNN:</u>

```
Classify(X,Y,x)
X= training data
Y= class labels of X
x= unknown sample

For i =1 to m
       Compute Distance d(Xi , x)
Compute set I containing indices for the k smallest distances d(Xi , x)
Return majority label for {Yi where i belongs to I)
```

<u>DWT:</u>

```
Public static int[ ] discreteWaveletTransform( int[ ] input){
       //this function assumes that input.length= 2^n , n>1
       Int[ ] output = new int[ input.length ];
       For (int length = input.length / 2 & length= length/2){
              //length is the current length of the working area of the output array
              //length starts at half of the array size and every iteration is halved 
until it is 1
              For (int i=0 ; i<length ; ++i){
                     Int sum= input[ i*2 ] + input[ i*2+1 ];
                     Int difference= input[ i*2 ] - input[ i*2+1 ];
                     Output[ i ]= sum;
                     Output[ length+i ]= difference;
}
if(length == 1){
       Return output;
}
system.arraycopy(output,0,input , 0, length);
       }
}

For i in range (X):
       coeffs= discreteWaveletTransform(X)
       cA1 , cD1= coeffs
       coeffs2= discreteWaveletTransform( cA1)
       cA2, cD2 = coeffs2
       coeffs3= discreteWaveletTransform( cA2 )
       cA3 , cD3= coeffs3
       coeffs4= discreteWaveletTransform(X)
```

```
        cA4 , cD4= coeffs4
        coeffs5= discreteWaveletTransform( cA4 )
        cA5 cD5= coeffs5

        For j in range(16):
                Processed [ i ] [ j ] [ 0 ] = cA5[ j ]
                Processed [ i ] [ j ] [ 1 ] = cD1[ j ]
                Processed [ i ] [ j ] [ 2 ] = cD2[ j ]
                Processed [ i ] [ j ] [ 3 ] = cD3[ j ]
                Processed [ i ] [ j ] [ 4 ] = cD4[ j ]
                Processed [ i ] [ j ] [ 5 ] = cD5[ j ]
```

GAME:

```
//updata is called once per frame
Void update(){
        readData();
        //makePieces();

        for( int i=0 ; i<Input.touchCount ; i++){
                if(Input.GetTouch(i).phase == TouchPhase.Began){
                        //construct a ray from current touch coordinates
                        transform.Translate( 0, 2, 0);
                }
        }
}
Void readData(){
        //read data from the port
}
Int makePieces(){
        // make the pieces in to 500 rows to make small samples
        Int r= callModel(tempArray);
        Return r;
}
Int callModel(Array tempArray){
        //call the model and get input
        Return 1;
}
```

PDFViewer:

```
//Declare buttons
//open default ACTION_GET_CONTENT  from android to select pdf
//create chooser
//get result code and check if it is OK
//load pdf
```

Sign-in:

```
//initialize the buttons
```

```
//initialize Paper(remembers username and  password) library
//set up the onClicks on buttons
//get the text from the EditTexts
//check if the information user entered is null
//remember the username and password on Paper
//initialize firebase database
//check if the table Users exists
//check if the email exists
//check if the password is correct
```

Sign-up :

```
//initialize the buttons
//initialize Paper(remembers username and  password) library
//set up the onClicks on buttons
//get the text from the EditTexts
//check if the information user entered is null
//initialize firebase database
//check if the table Users exists
//check if the email exists
F//start the default ACTION_GET_CONTENT for GalleryPick
//if pic upload is successful then upload it to database storage
//create a HashMap of all the data
//upload the data on database
//If upload is successful then start activity login
//remember the username and password on Paper
```

5.2 External APIs

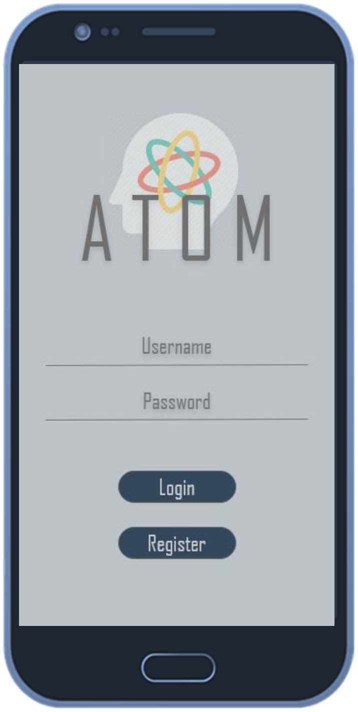
Table 5.1: Details of APIs used in the Project

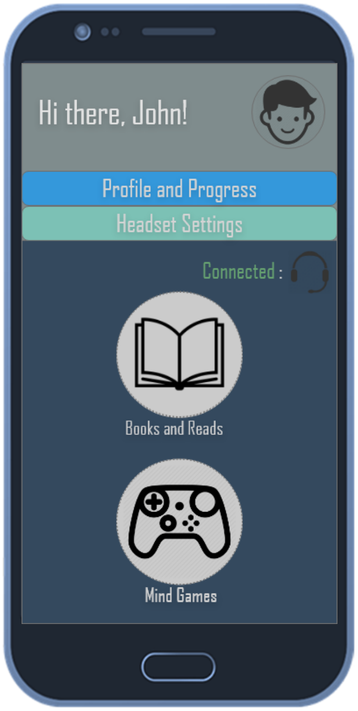
Name of API	Description of API	Purpose of Usage	List down the function/class name in which it is used
cortex	This API is given by the Emotiv headset to access the data from the headset.	This API is used to locate and connect to the headset and then send data to the user as per subscription	It is used in the server file named index.js
Firebase authentication	This is a firebase provided API to enforce user authentication and verification	It is used in the account handling module allowing users to sign in and sign up	- AccountActivity - BookActivity - ConnectionActivity - LoginActivity - MainActivity - PasswordRecoverActivity - ProgressActivity - QuestionActivity - ReaderActivity - RegisterActivity - SplashActivity

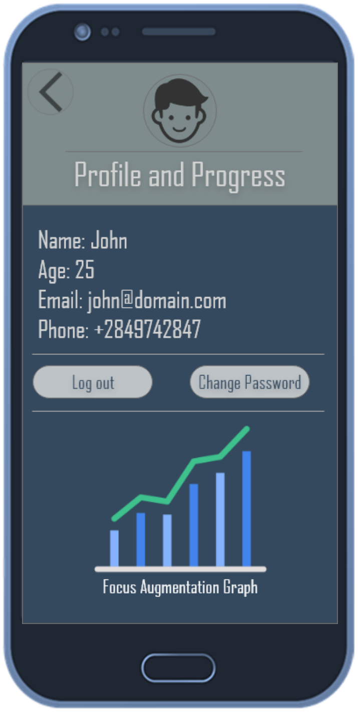
Firebase database	Another firebase provided API to add support of user data storage	Enabling preservation of user information to allow user driven content	- `ReaderActivity` - `ProgressActivity` - `BookActivity`
Facebook Android	Used alongside the Firebase auth API provided by facebook	To provide OAuth capabilities using facebook accounts	- `AccountActivity` - `LoginActivity`
Room	An API provided by the android jetpack	This was used to incorporate the model-view controller engineering strategy	- `Library(package)`
Android PDF viewer	Open-source MIT licensed API developed by barteksc	To render the .pdf file format for documents and view them	- `ReaderActivity`
Volley & OKHttp3	Open-source MIT licensed API	Used for web communication in JSON helping to achieve the Books API support	- `BookActivity`
MPAndroidChart	Open-source MIT licensed API developed by PhilJay	Used for graph viewing and chart display	- `ProgressActivity` - `ConnectionActivity`

5.3 User Interface

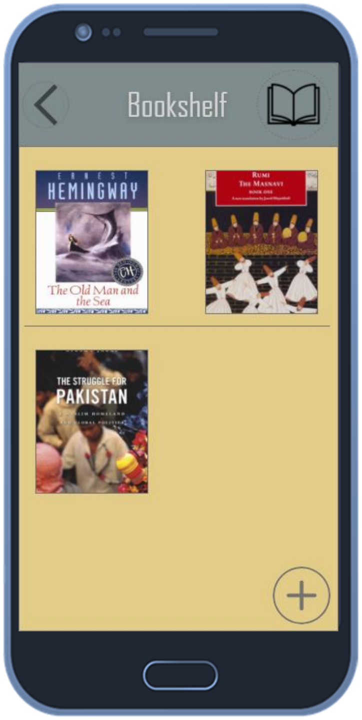
The following are the user interface mockups that were presented in the initial scope document. The final implementation interfaces follow a similar fundamental set of components and often feature exactly the same design, but majorly, there are several augmentations to this design to incorporate all the added implementations:

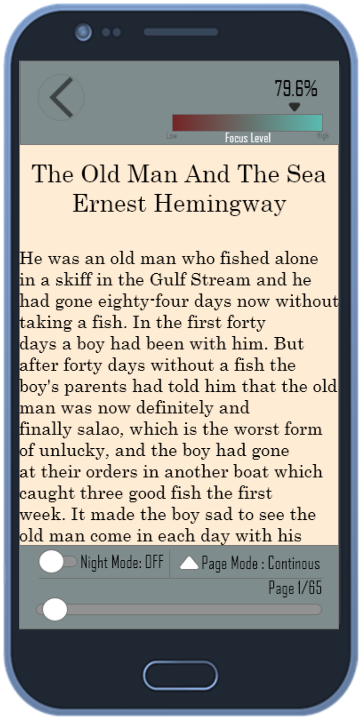






Name: John
Age: 25
Email: john@domain.com
Phone: +2849742847

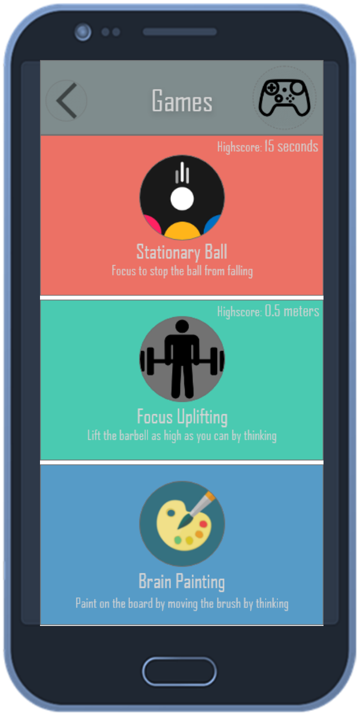




The Old Man And The Sea
Ernest Hemingway

He was an old man who fished alone in a skiff in the Gulf Stream and he had gone eighty-four days now without taking a fish. In the first forty days a boy had been with him. But after forty days without a fish the boy's parents had told him that the old man was now definitely and finally salao, which is the worst form of unlucky, and the boy had gone at their orders in another boat which caught three good fish the first week. It made the boy sad to see the old man come in each day with his

Night Mode: OFF
Page Mode : Continous
Page 1/65



Highscore: 15 seconds
Stationary Ball
Focus to stop the ball from falling

Highscore: 0.5 meters
Focus Uplifting
Lift the barbell as high as you can by thinking

Brain Painting
Paint on the board by moving the brush by thinking

6 Testing and Evaluation

This section discusses on testing strategies at all abstraction levels. Afterwards, testing is solidified by providing well posed testing test cases by estimating the type and actual test data involved.

6.1 Manual Testing

Comprises of the manual testing techniques at both system and unit levels.

6.1.1 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.

6.1.2 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’	Username: 0332157665245 Password: 123456	Account not created in ATOM	Contact must be valid.	Fail

	button on Sign up form with correct input data				

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 :	Successfully Profile is Edited .	Account is Edited successfully	Pass

		21			
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	Game quits and games-	Pass

			user back to the games-list.	list is opened.	

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	Unity updates the model.	Successfully updates the model.	model is updated to cortex profile	Pass

	model works properly.				

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

7 Conclusion

Considering the presented problem and the solution, it’s 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.

7.1 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 .

8 References

1. Younchang Li, Xiaowei Li, Martyn Ratcliffe, Li Liu, Yanbing Qi, Quanying Liu, (Sep 2011), *A Real-time EEG-based BCI System for Attention Recognition in Ubiquitous Environment*, https://www.researchgate.net/publication/254007001_A_real-time_EEG-based_BCI_system_for_attention_recognition_in_ubiquitous_environment/link/5554f1b408ae6943a871b157/download

2. Newton N. Minow (May 1961), *Television and the Public Interest* https://www.americanrhetoric.com/speeches/newtonminow.html

3. Yang Jiang, Reza Abiri, Xiaopeng Zhao (2017), *Tuning Up the Old Brain with New Tricks; Attention Training via Neurofeedback*, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5346575/