

# Software Design Description

## 1. Introduction

### 1.1. Purpose

This Software Design Document is intended for the project “Dyslexia Diagnosis Tests and Educational Games,” an inclusive digital solution aimed at diagnosing and supporting children with dyslexia at four or older years old. The primary goal of this document is to provide a detailed plan for the design and development of the software. It will serve as a guide for the development and a reference for stakeholders to understand the project's design development and implementation strategies clearly.

Dyslexia, a widespread disease affecting reading, writing, and spelling, is often misunderstood and underdiagnosed. Dyslexia is complex; it varies significantly in its presentation, making it difficult to identify with only observation. It changes differently from person to person, influenced by various factors, including language, age, and education. Due to those facts, there is a widespread lack of understanding and numerous misleading conclusions about dyslexia. Many still see it as just a reading disorder, ignoring its impacts on learning and cognition.

In addition, professional diagnosis of dyslexia can be costly. It often contains comprehensive assessments by psychologists, educators, or speech-language therapists. In many regions, especially low-income areas, access to tools capable of diagnosing dyslexia is limited. Parents must travel long distances, adding to the overall cost and difficulty of obtaining a diagnosis.

Schools may lack the resources or trained personnel to identify and support students with dyslexia effectively. Without a proper diagnosis, students with dyslexia often are unsuccessful in their academic lives, and that leads to long-term educational and social consequences.

The aim of our Dyslexia Diagnosis Tests and Educational Games is to facilitate all these problems by giving children two types of games, one for diagnosing dyslexia and the second for increasing their skill development. Diagnosis involves engaging in three innovative and child-friendly tests: Letter Matching, Navigation Skills, and Symmetry. Those tests will test children's cognitive and linguistic abilities

related to reading, writing, and directional knowledge. For skill improvement, a Picture Matching Game will be available. This game is designed to address specific challenges faced by children with dyslexia. After children play the diagnosis game, depending on their score, accuracy will be calculated and shown on the screen. The accuracy will tell parents whether their children are at risk of having dyslexia disease.

This document outlines the software's overall structure, including the high-level architecture, data flow, user interface design, and algorithm for diagnosis and educational activities. It also addresses critical factors such as security, privacy, and compliance with relevant academic and health standards.

This implementation of “Dyslexia Diagnosis Tests and Educational Games” will provide a significant advancement in educational technology, offering an efficient tool for early dyslexia detection and intervention.

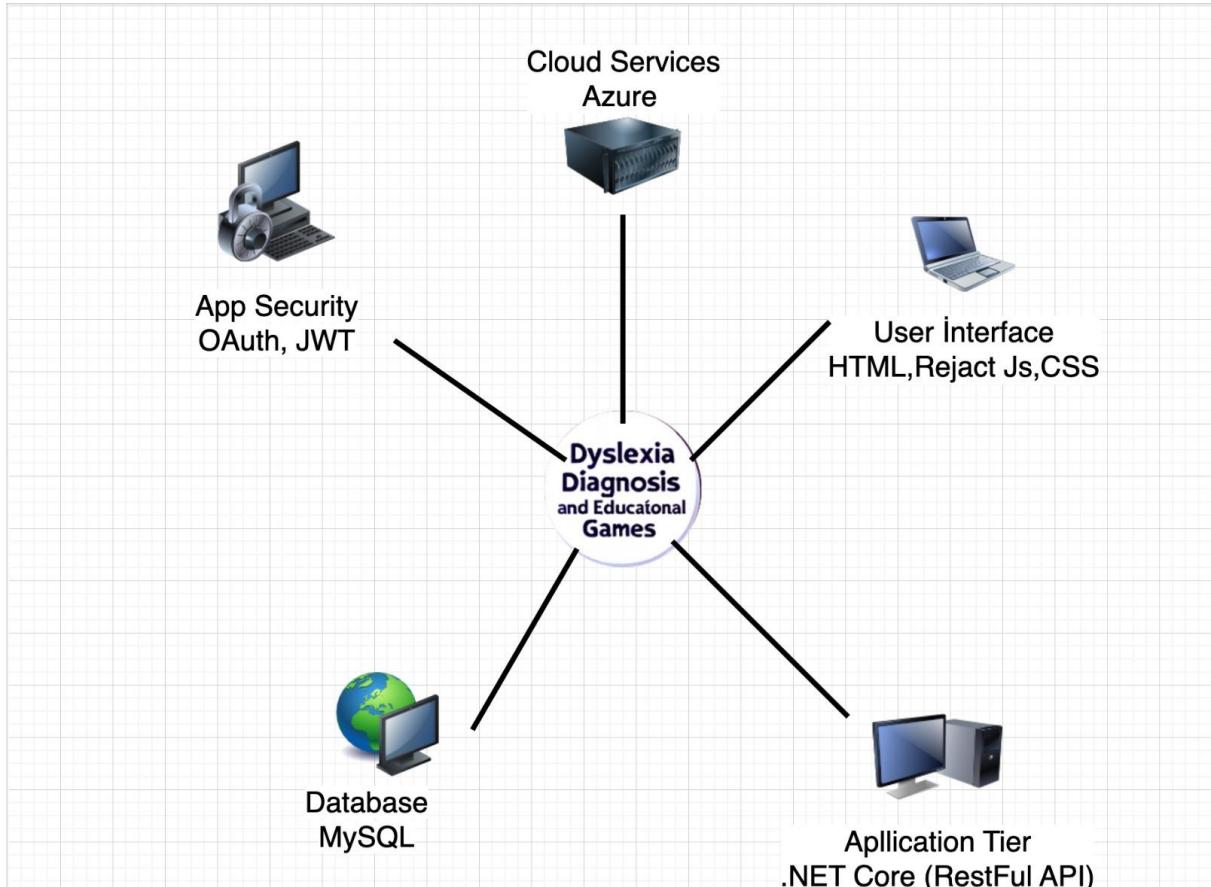
## **1.2. Scope of Project**

This document contains a complete description of the design of “Dyslexia Diagnosis Tests and Educational Games.” While using the application, the user is expected to log in. After login, a child-friendly screen with two options will be provided—one for dyslexia diagnosis tests and one for educational games. Letter Matching Test is a test that tests the children’s reading abilities. The test will show four random letters and an answer letter, and the children will choose the correct letter among the four random letters. Navigation Skills Test is a test that tests the children’s directional knowledge. The test will display alerts based on location and left/right buttons for selection. The children will try to go to that location using the left/right buttons. The Symmetry Test is a test that tests the children’s visual abilities. The test will display four random pictures and one answer, and the children will choose the correct symmetrical match.

## **1.3. Overview of the software architecture**

The software architecture for “Dyslexia Diagnosis Tests and Educational Games” is designed to support its dual functionality of dyslexia diagnosis and educational gameplay. MySQL database, a widely used open-source relational database management system known for its reliability, ensures consistency in the app data. Developed with an emphasis on performance and stability, MySQL is a system for applications requiring structured data organization and integrity. The project's architecture is based on a cross-platform running on .Net Core, which will enable

access to the application on both mobile and desktop using ReactJS, HTML, and CSS. The business logic will access the data through RESTful APIs provided by .NET Core. OAuth and JWT will be used for user login and authorization, ensuring the application's security. All these components will be hosted on the Azure cloud service, enabling the application to run and scale online.



## 2. Requirements

### 2.1. Functional Requirements

#### 2.1.1.

#### 2.1.2. User Registration and Login System

- A system that allows users to register and log in to the application.
- Secure storage of user information.
- A user interface suitable for children's ages.

#### 2.1.3. Dyslexia Diagnosis Tests Module

- Interactive tests and assessments to determine users' dyslexia status.

- Record test results and provide feedback to parents/users.

#### **2.1.4. Educational Games Module**

- A variety of educational games that address challenges specific to dyslexia.
- Each game has goals to improve children's reading and writing skills.
- We are monitoring children's progress through games and recording this progress.

#### **2.1.5. Parent/User Monitoring and Reporting**

- Parents can monitor the child's activities and progress within the application.
- We are providing periodic reports and recommendations to parents.

#### **2.1.6. User-Friendly Interface**

- An intuitive and visually appealing user interface that children can easily use.
- Easy navigation and straightforward instructions within the app.

#### **2.1.7. Data Security and Privacy**

- Secure storage and processing of user data.
- Clearly state privacy policies and protect users' data.

#### **2.1.8. In-App Help and Support**

- Help sections that guide users on how to use the app.
- Availability of technical support and feedback mechanisms.

### **2.2. Non-Functional Requirements**

#### **2.2.1. Performance**

- Fast loading of the app and low latency.
- Performance optimized for multi-user support.
- Scalability and resource management are provided by Azure infrastructure.

#### **2.2.2. Security**

- Encryption of user data and secure data transfer.
- Effective use of Azure security features (e.g., firewalls, authentication mechanisms).

- Privacy policies and compliance are designed specifically for children.

#### **2.2.3. Availability**

- It is an easy-to-understand and intuitive user interface.
- The design is suitable for children's age and skill levels.
- Multi-platform support (e.g., iOS, Android, web).

#### **2.2.4. Reliability**

- High availability and reduced system interruptions.
- Managing data backup and disaster recovery plans via Azure.
- Automatic updates and regular maintenance.

#### **2.2.5. Scalability**

- It automatically increases resources as users increase.
- We are leveraging Azure's scaling capabilities.
- Flexible management of data storage and processing capacities.

#### **2.2.6. Compatibility**

- Compliance with local and international laws regarding protecting children's data.
- Compliance with educational standards and best practices for dyslexia.

#### **2.2.7. Sustainability**

- Use of Azure's environmentally friendly features for energy-efficient operations.
- Long-term sustainable development plans.

### **2.3. Constraints and Assumptions**

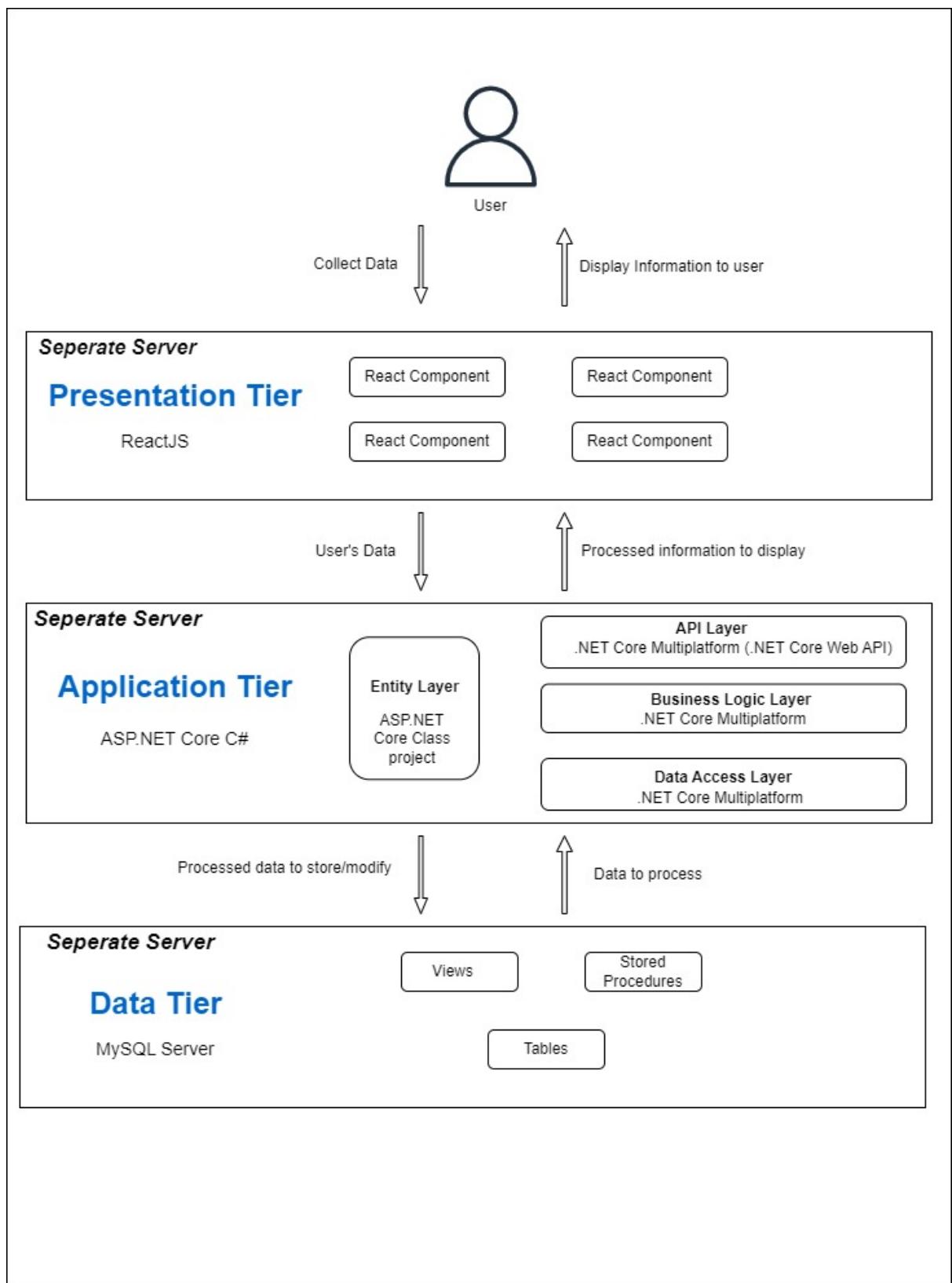
The system will be optimized for desktop and mobile devices. Performance will decrease due to internet connection requirements and low internet speeds. There will be compliance with legal regulations regarding children's data privacy and security. There will be compliance with regulations regarding education and health.

- The system assumes a reliable internet connection.
- It operates within the constraints of web browser capabilities.

## **3. Architecture**

### **3.1. A High-Level Overview of the Software Architecture**

Consists of three tiers: client, server, and database. The arch project is architecture-based on a cross-platform application running on .NET Core and has a user interface developed using ReactJS, HTML, and CSS. This application will store critical data using the MySQL database, and the business logic will access this data through RESTful APIs provided by .NET Core. OAuth and JWT will be used for user login and authorization, ensuring the security of the SECU application. These components will be hosted on the Azure cloud service, enabling the application to run and scale online.

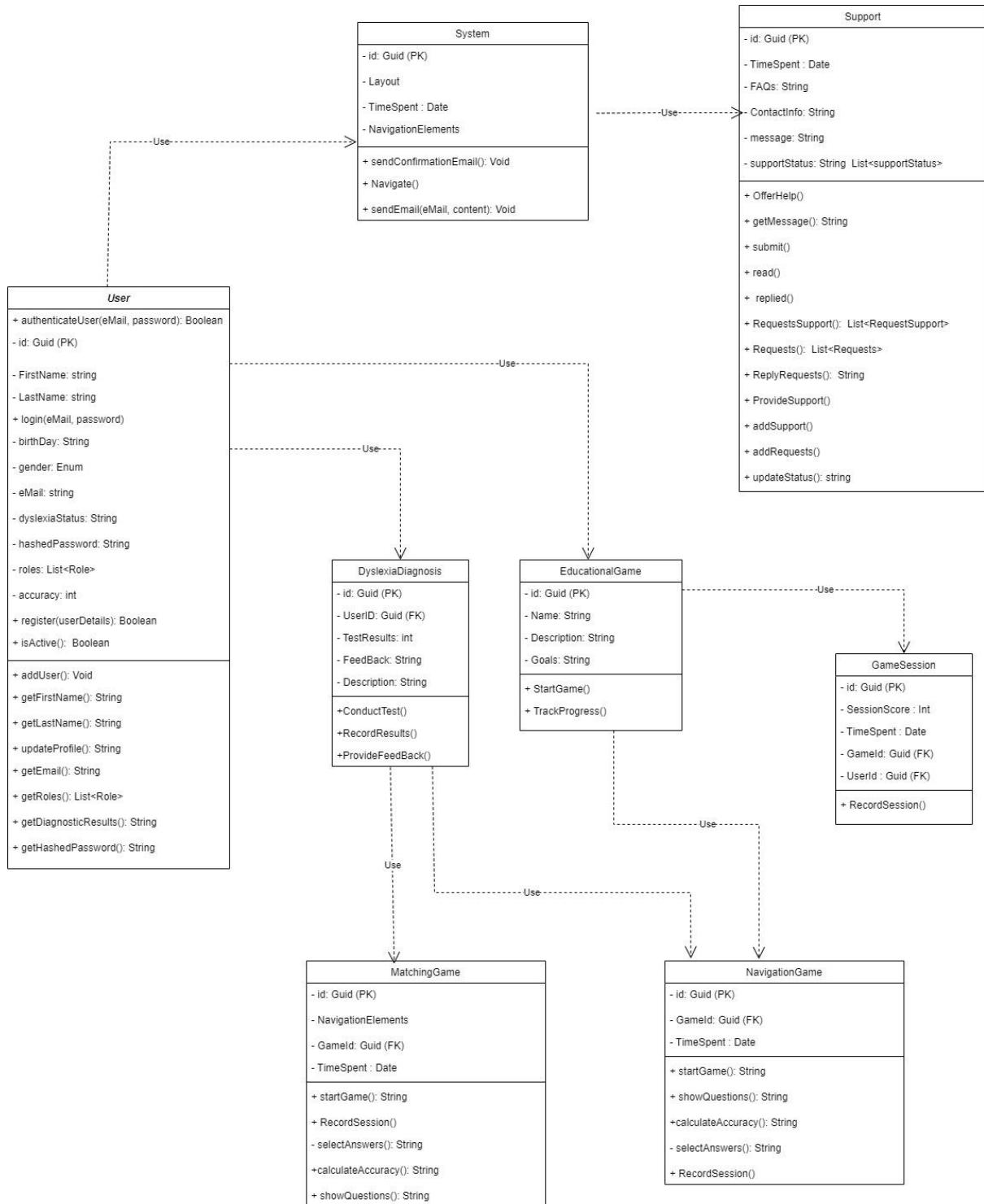


### **3.2. Used Technologies**

- Back-End Programming: .NET Core for application server, MySQL for Database.
- Front-End Programming: ReactJS, HTML, and CSS.

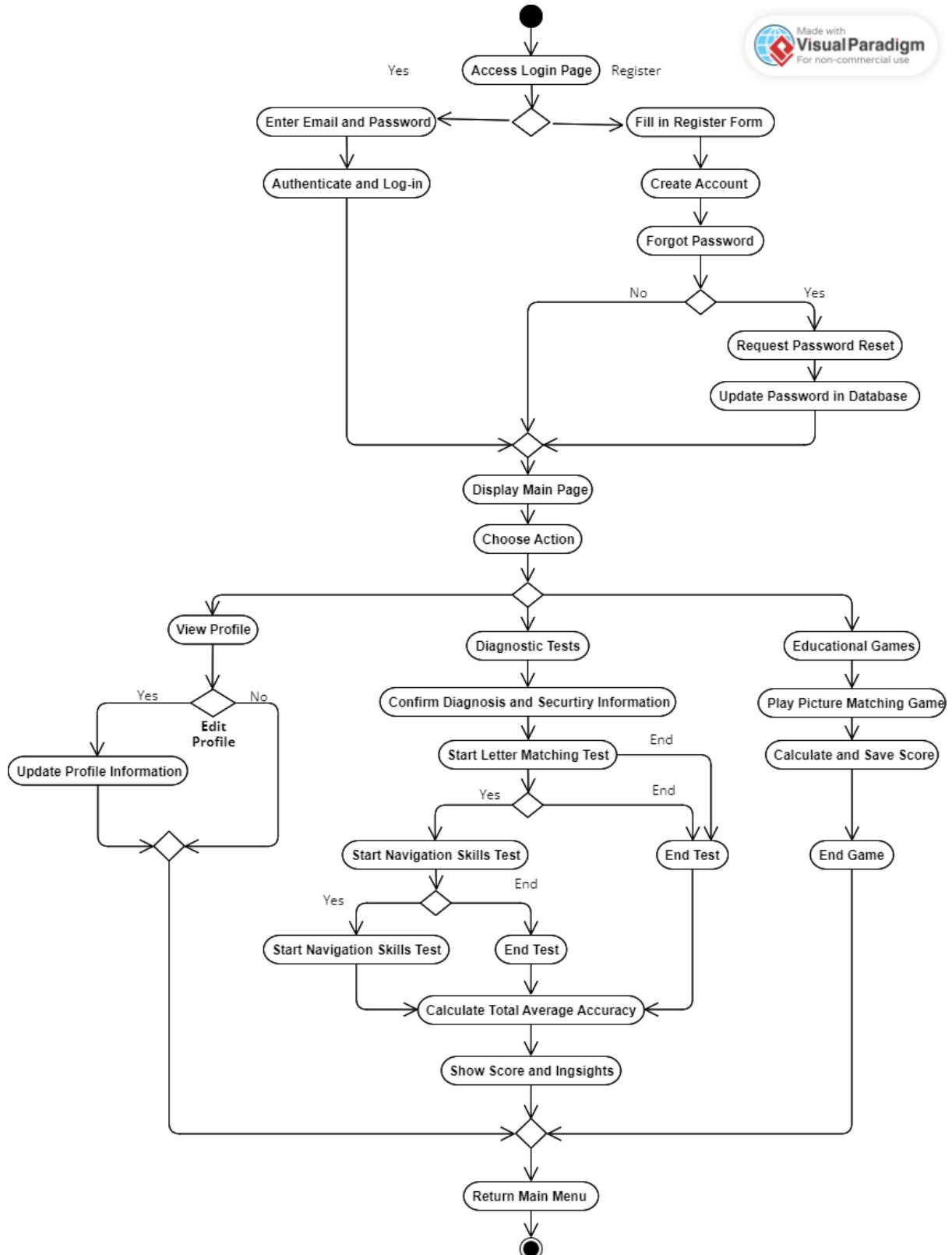
## 4. Diagrams or Sketches of the Architecture

### 4.1. Class Diagram

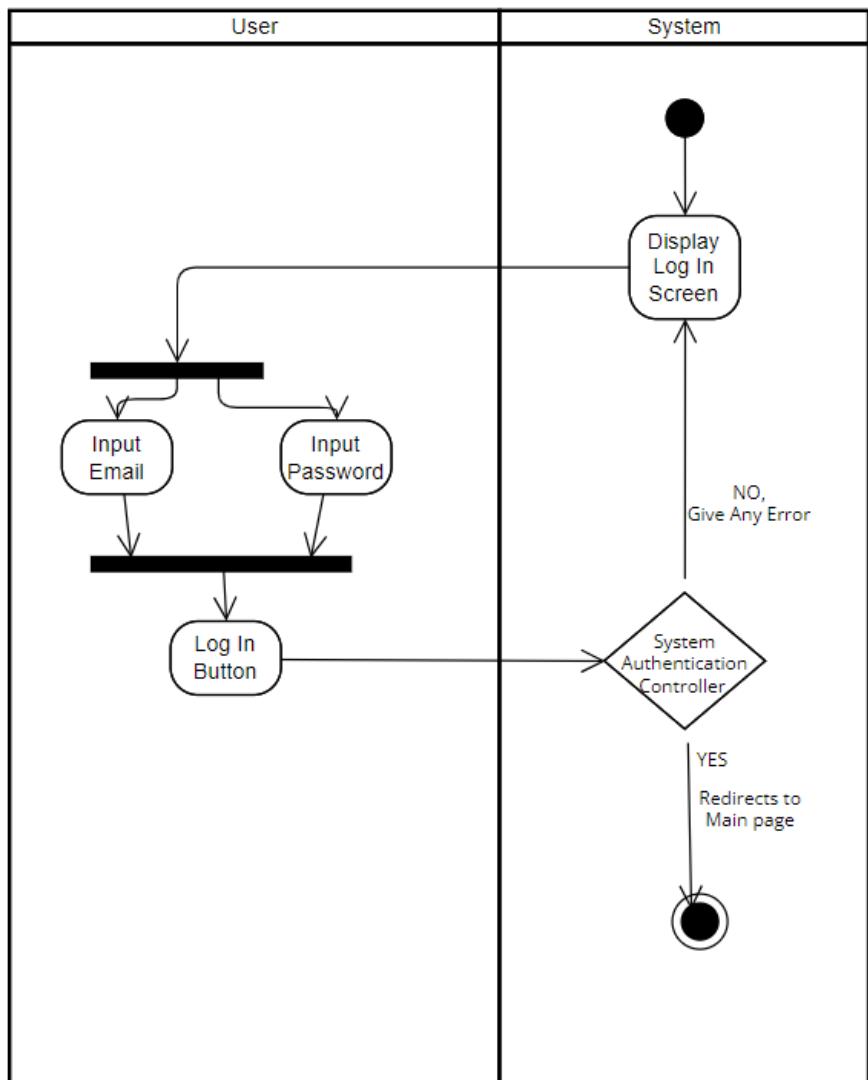


## 4.2. Activity Diagram

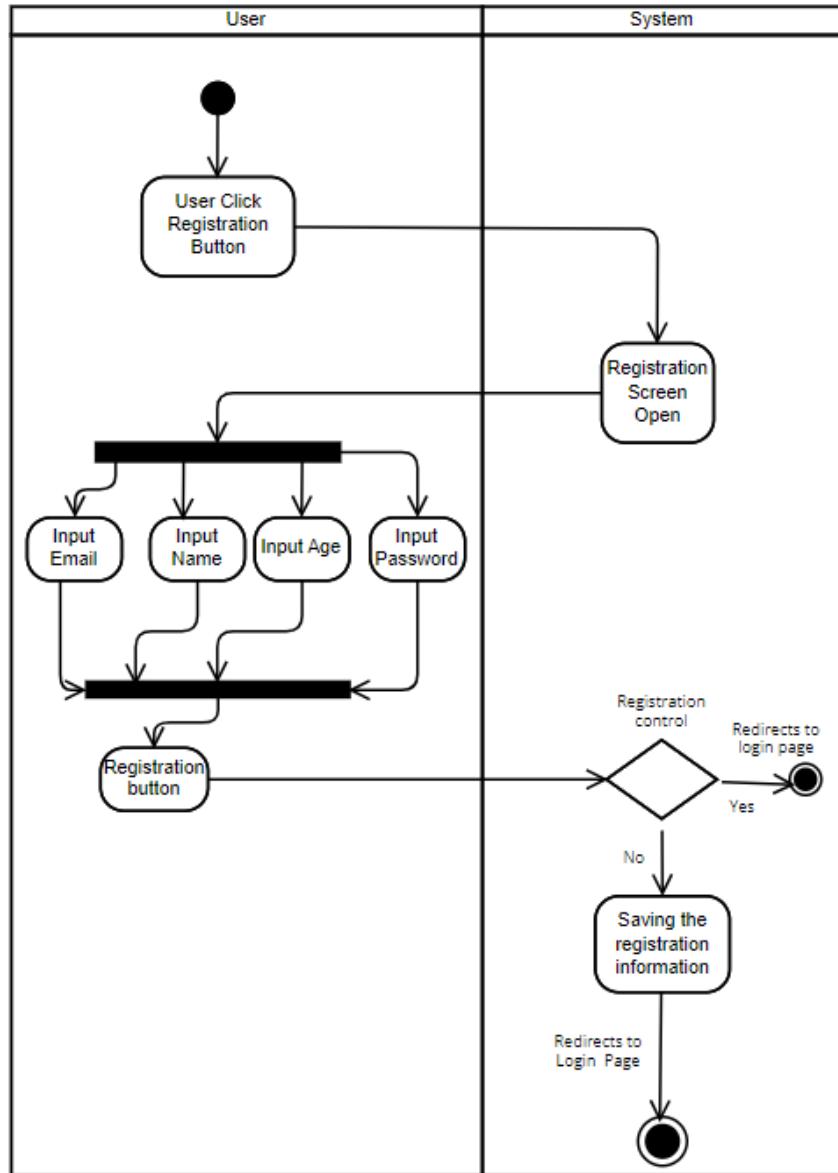
### 4.2.1. System Activity Diagram



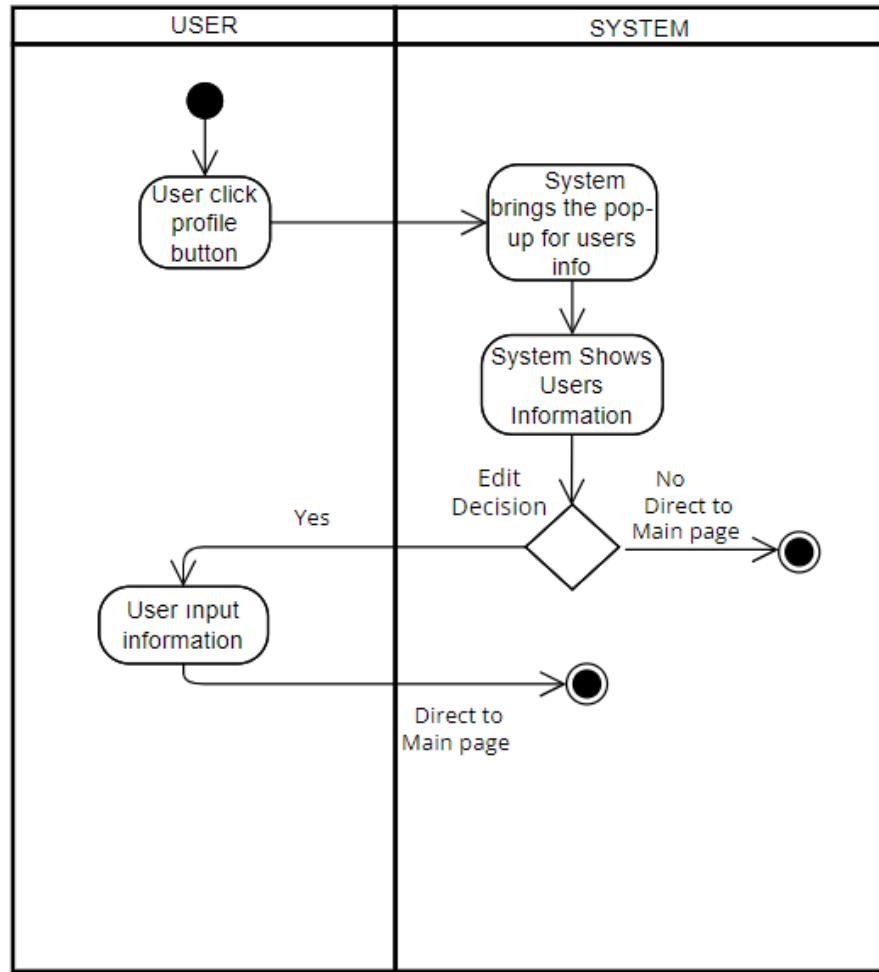
#### 4.2.2. Login Activity Diagram



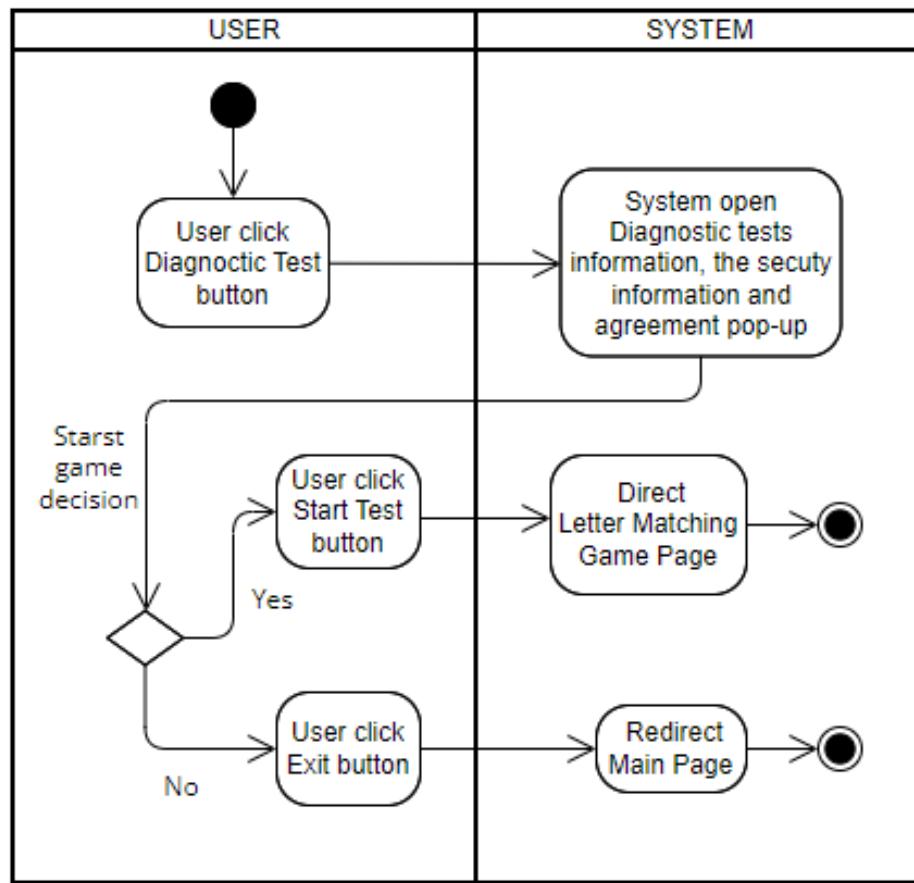
#### 4.2.3. Register Activity Diagram



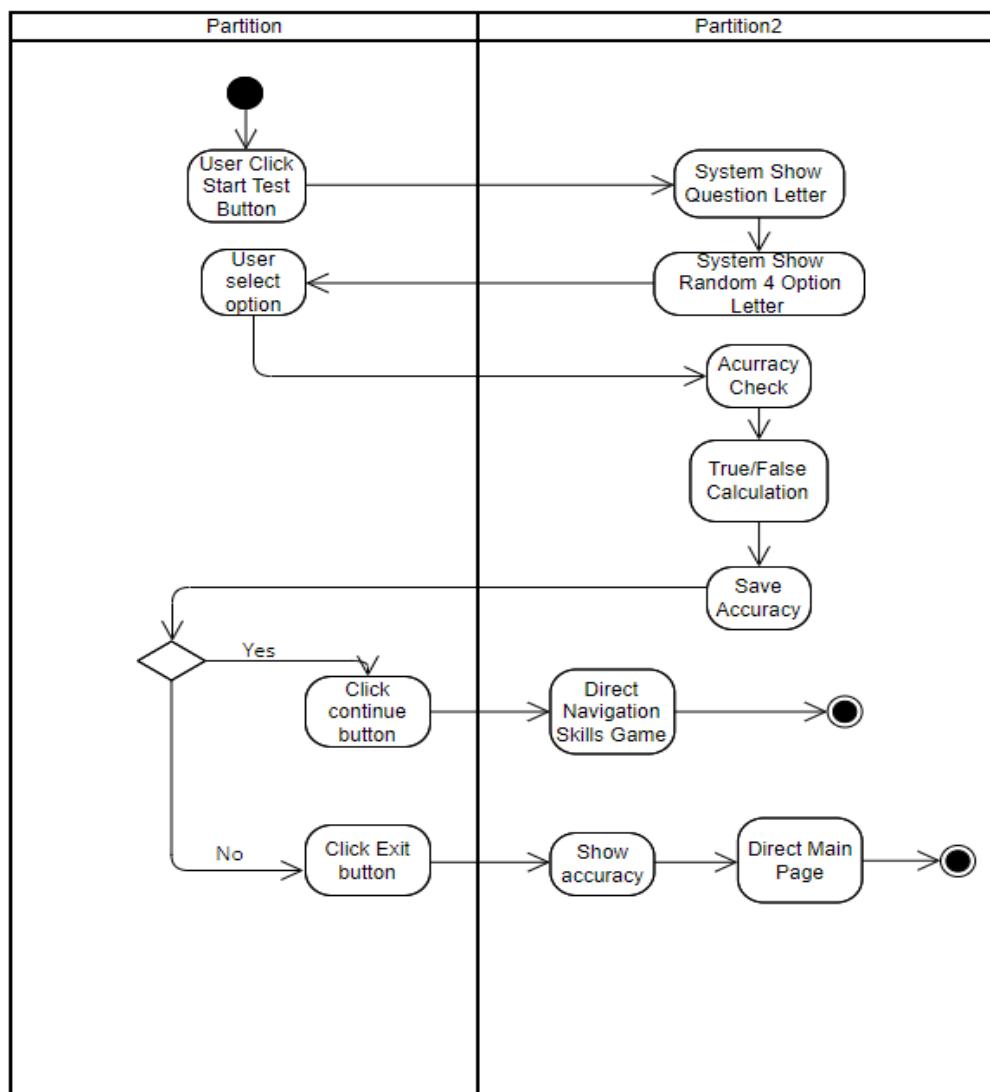
#### 4.2.4. Profile Activity Diagram



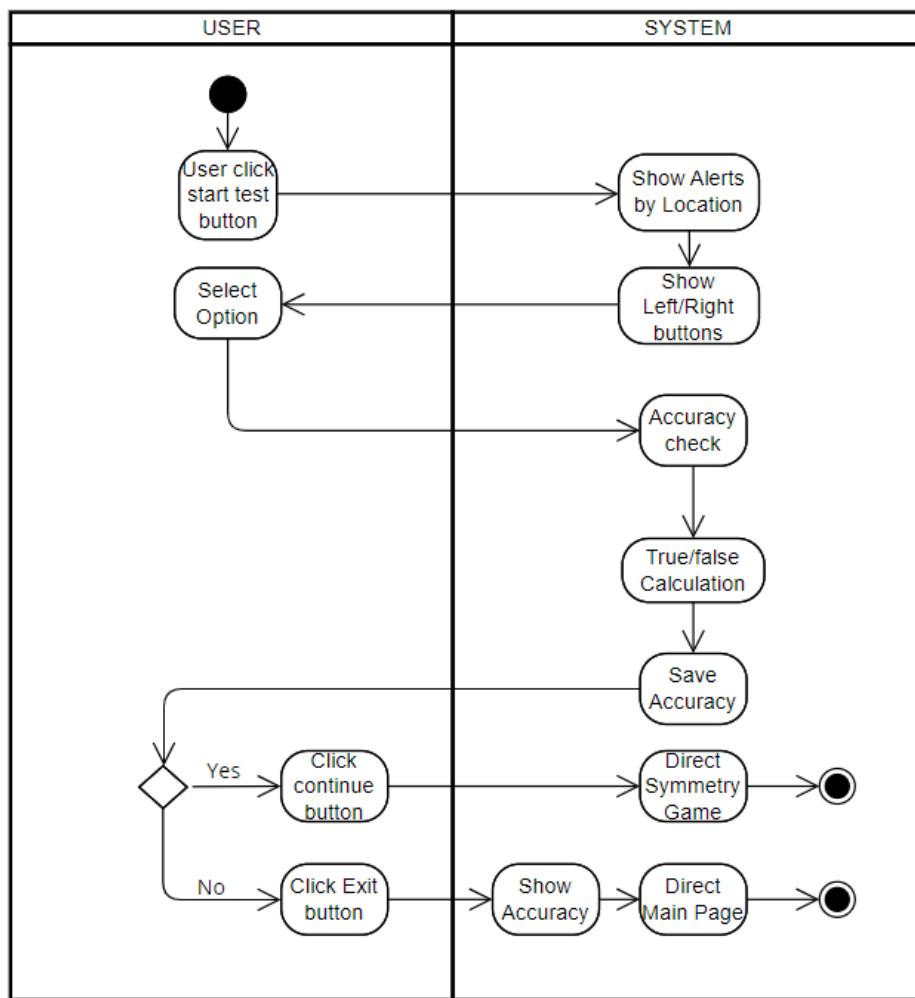
#### 4.2.5. Diagnostic Tests Information Activity Diagram



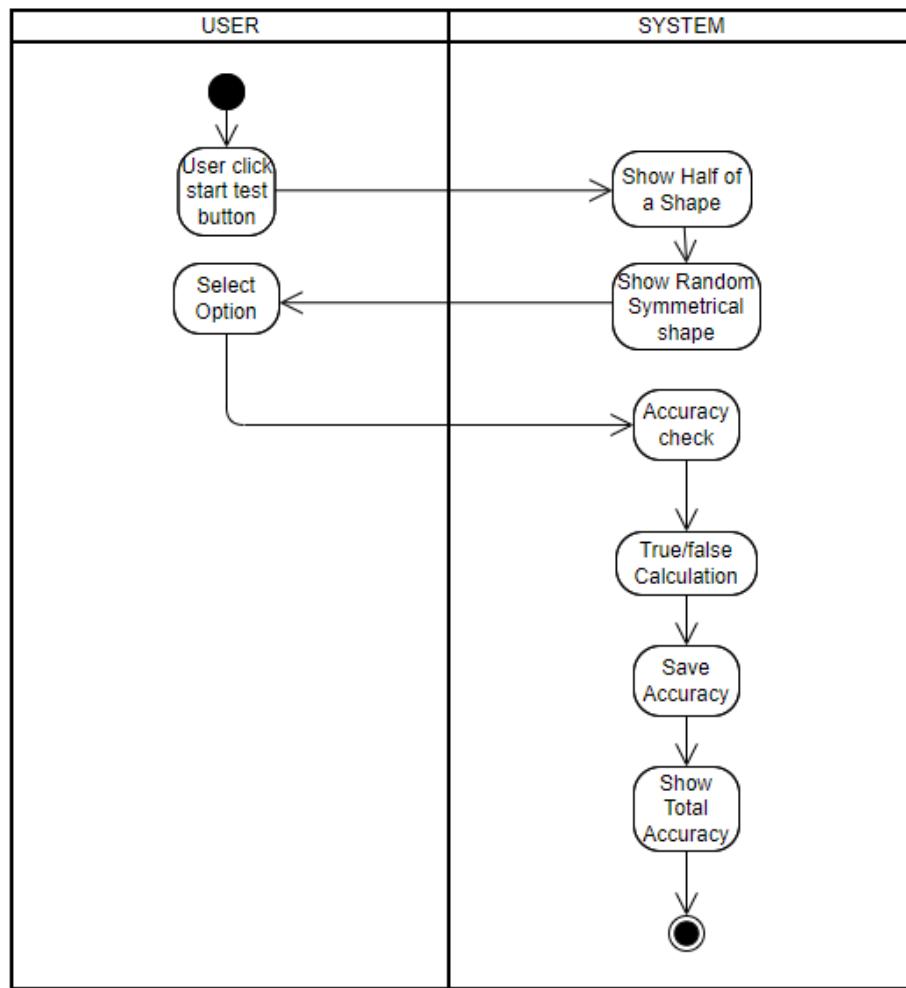
#### 4.2.6. Letter Matching Test Activity Diagram



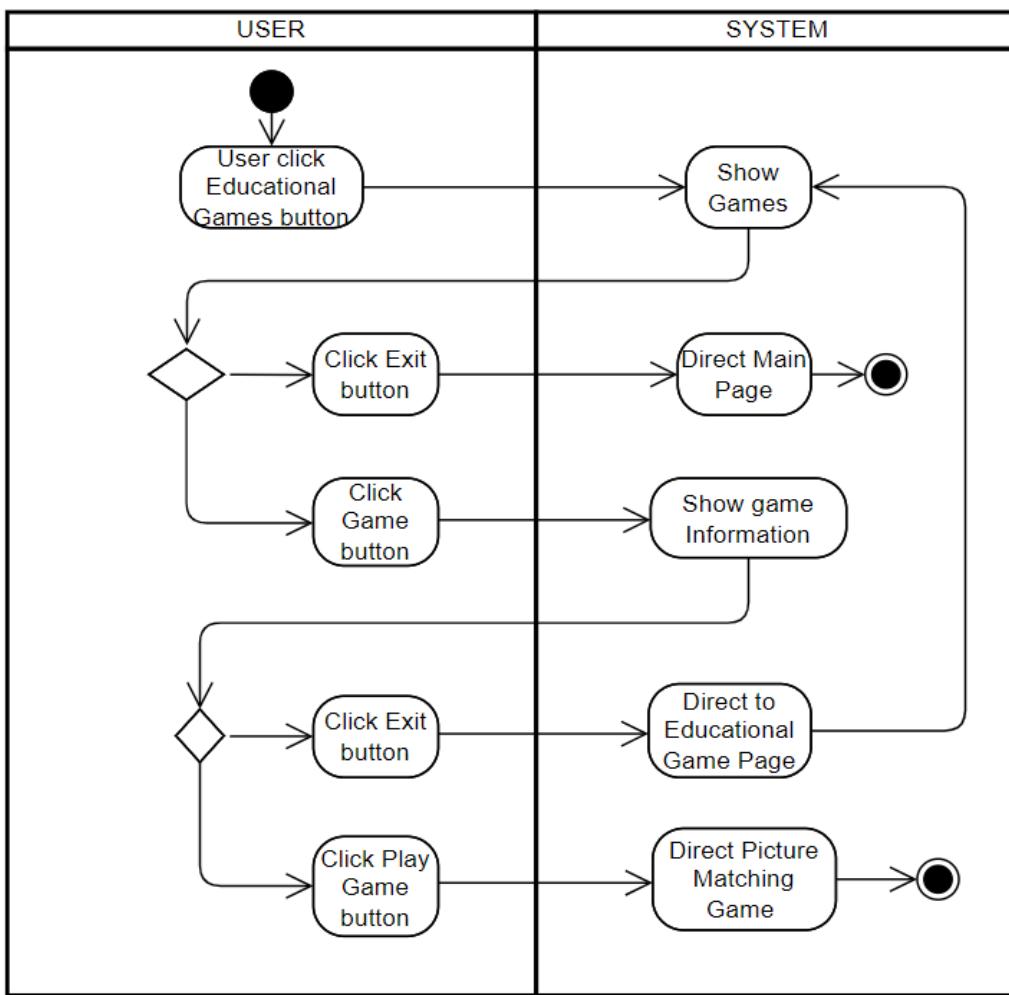
#### 4.2.7. Navigation Skills Test Activity Diagram



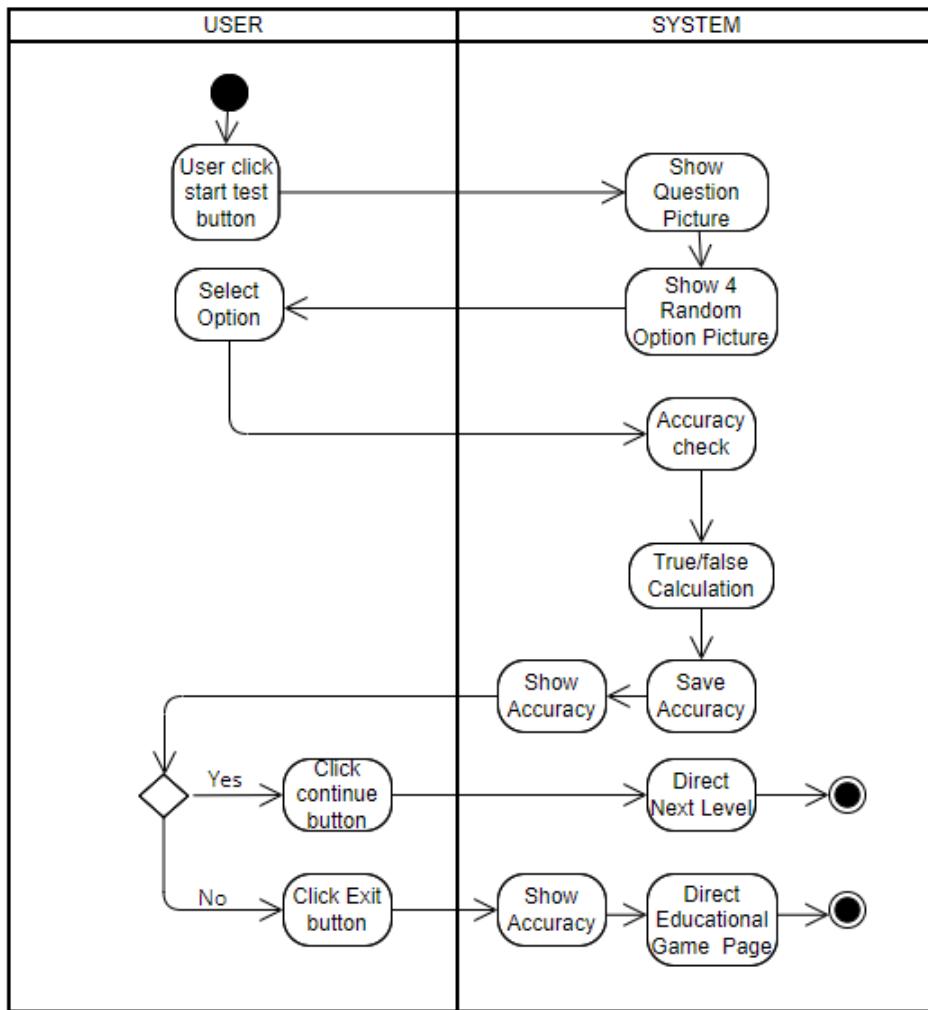
#### 4.2.8. Symmetry Test Activity Diagram



#### 4.2.9. Educational Game Activity Diagram

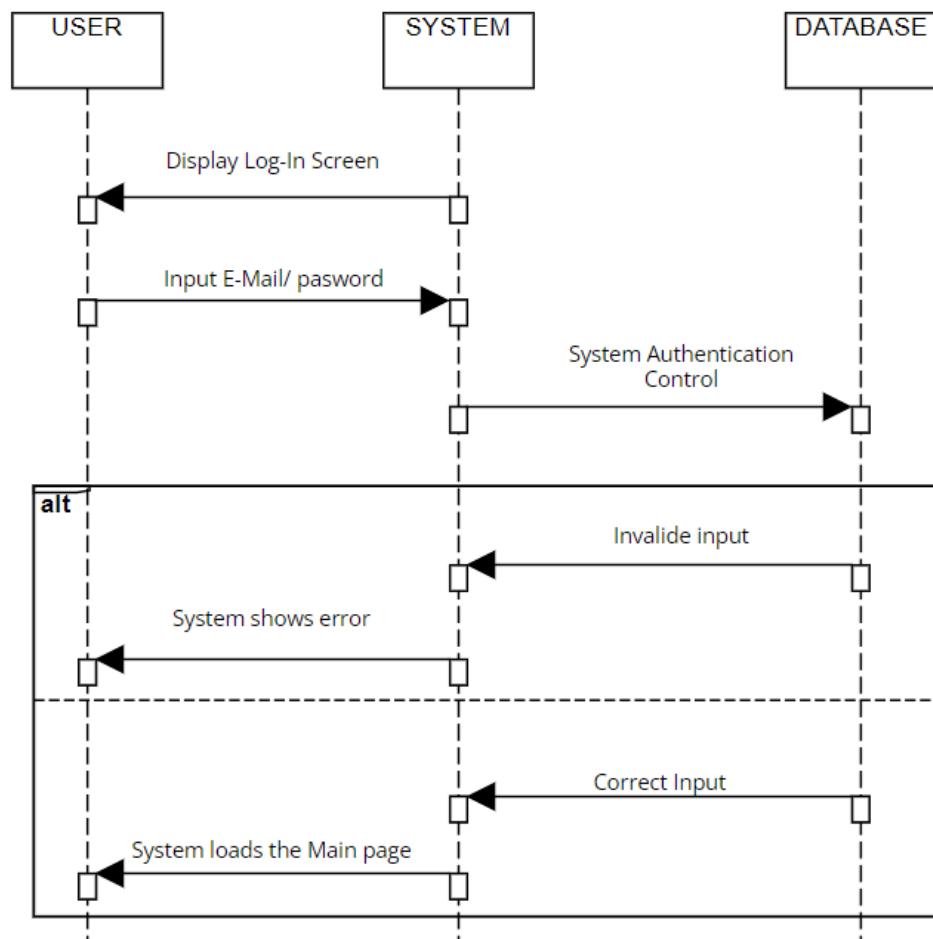


#### 4.2.10. Picture Matching Game Activity Diagram

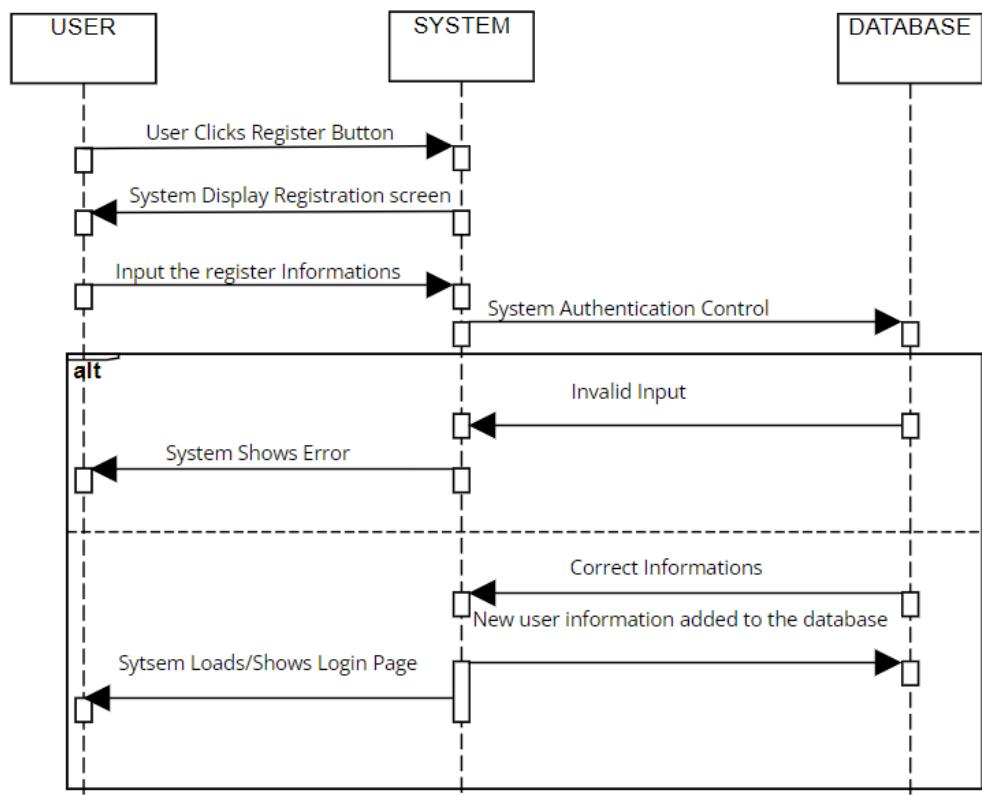


## 4.3. Sequence Diagram

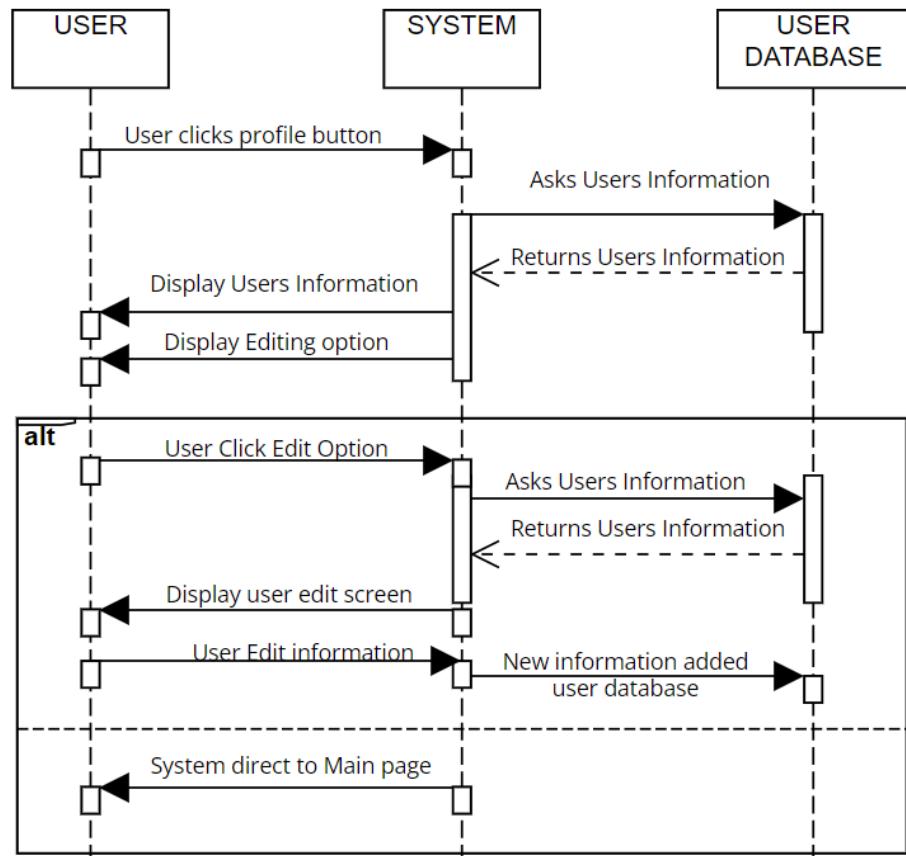
### 4.3.1. Login Sequence Diagram



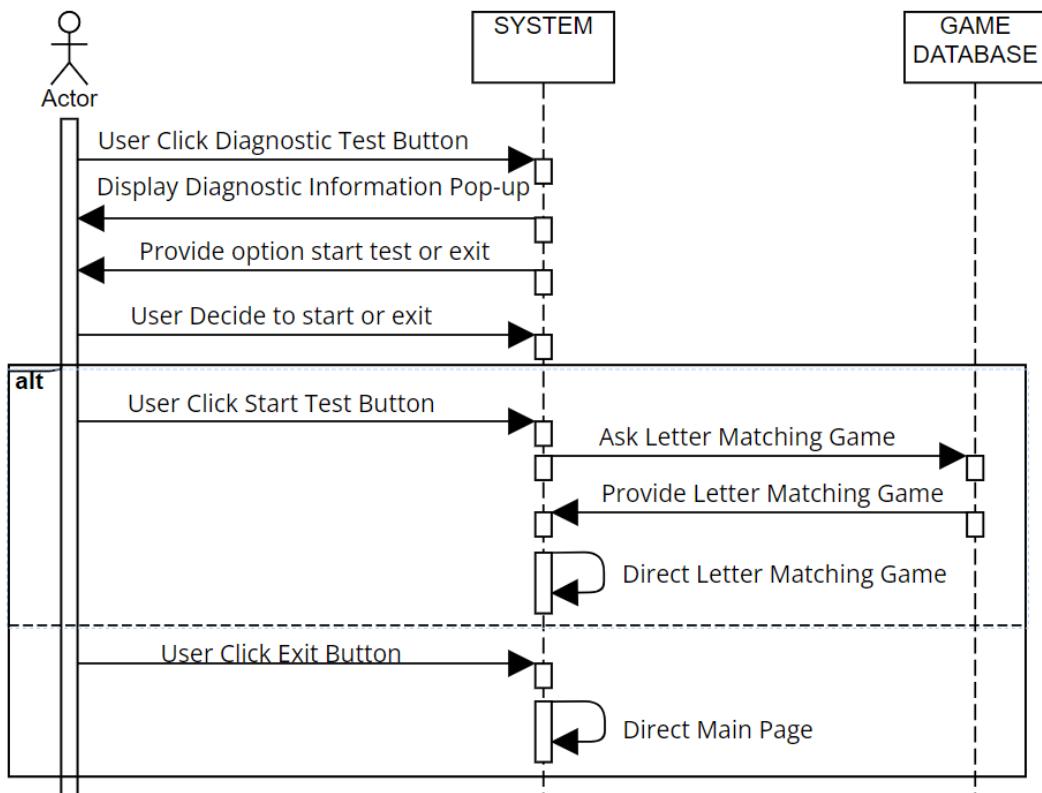
#### 4.3.2. Register Sequence Diagram



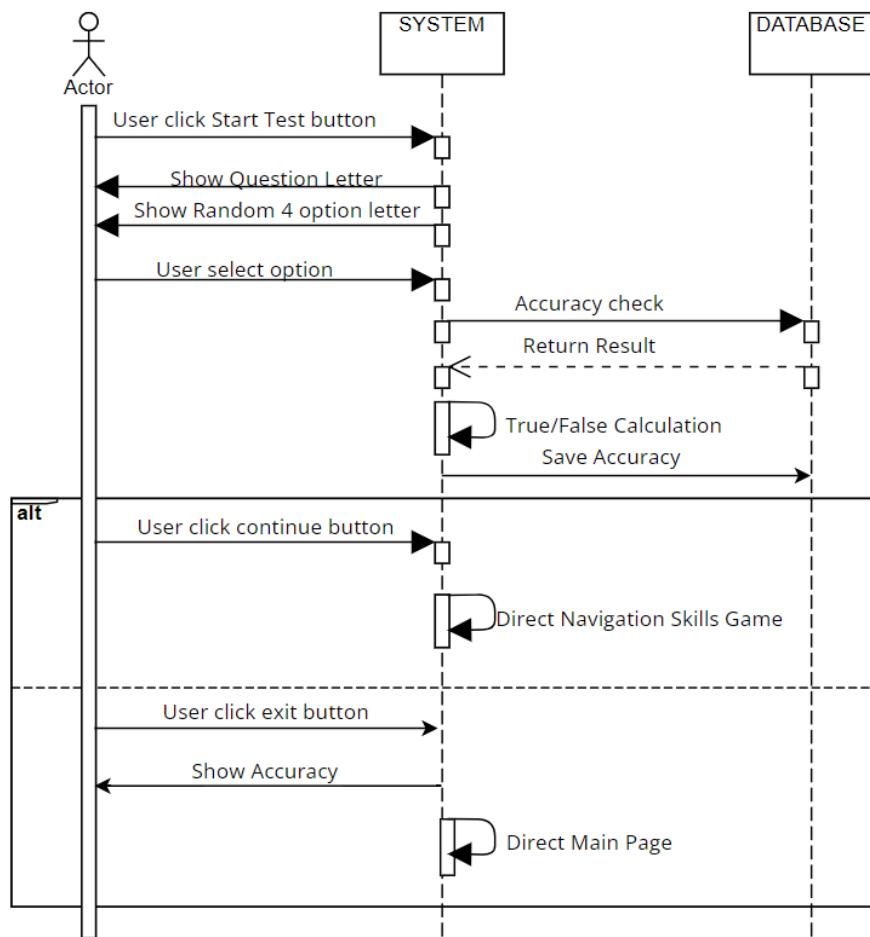
### 4.3.3. Profile Sequence Diagram



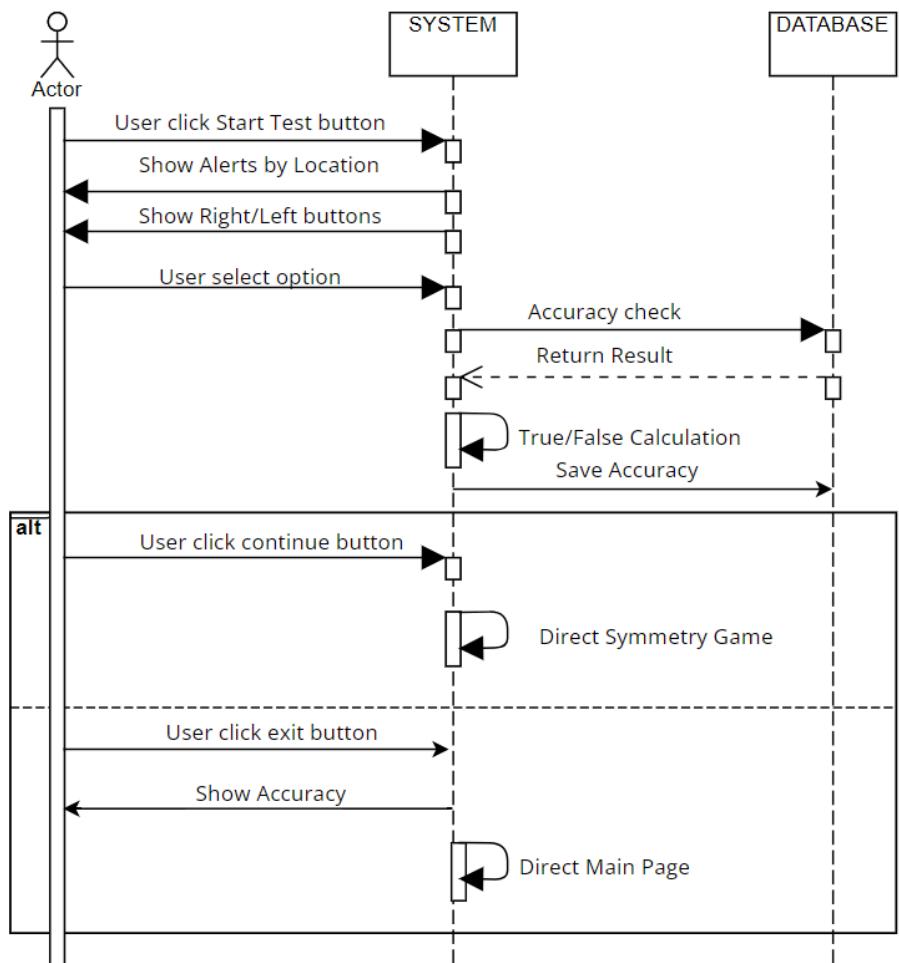
### 4.3.4. Diagnostic Tests Information Sequence Diagram



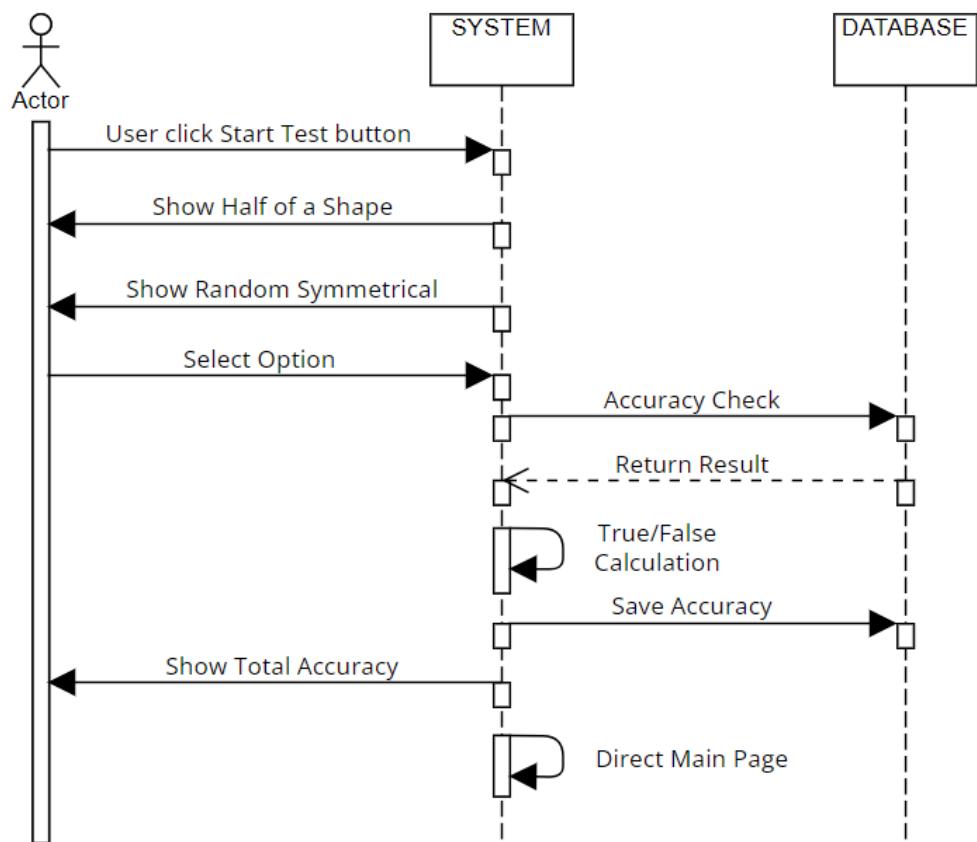
#### 4.3.5. Letter Matching Test Sequence Diagram



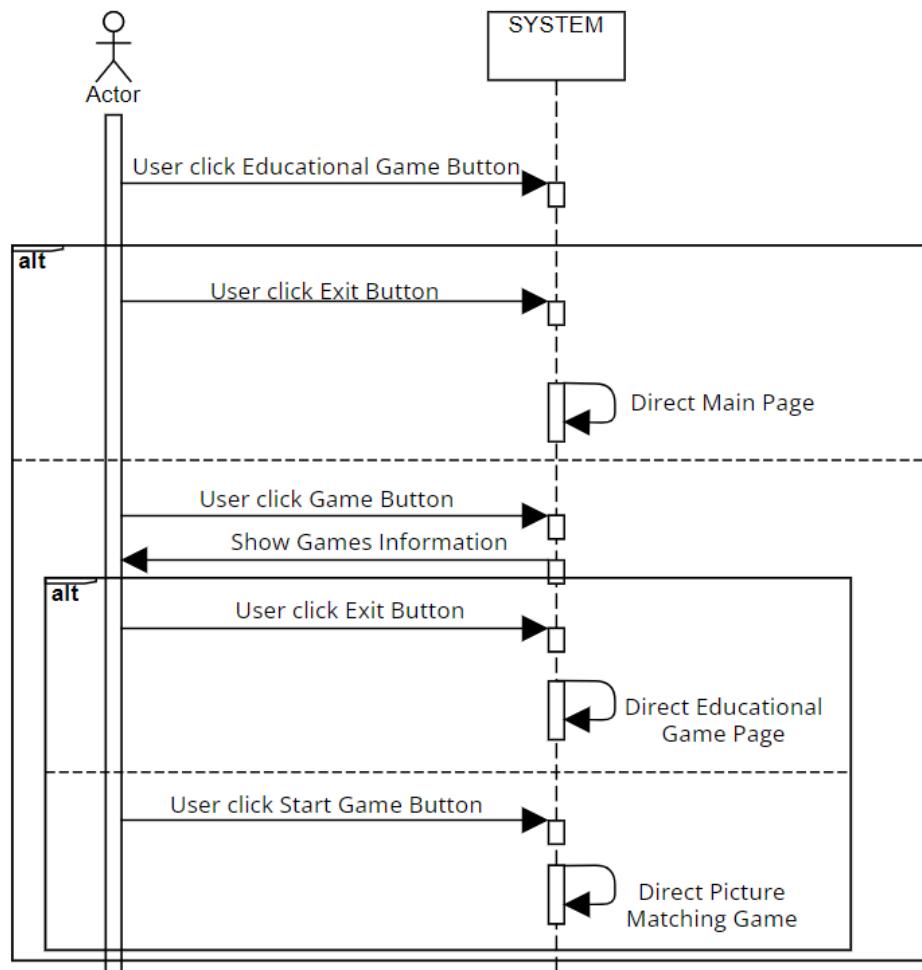
#### 4.3.6. Navigation Skills Test Sequence Diagram



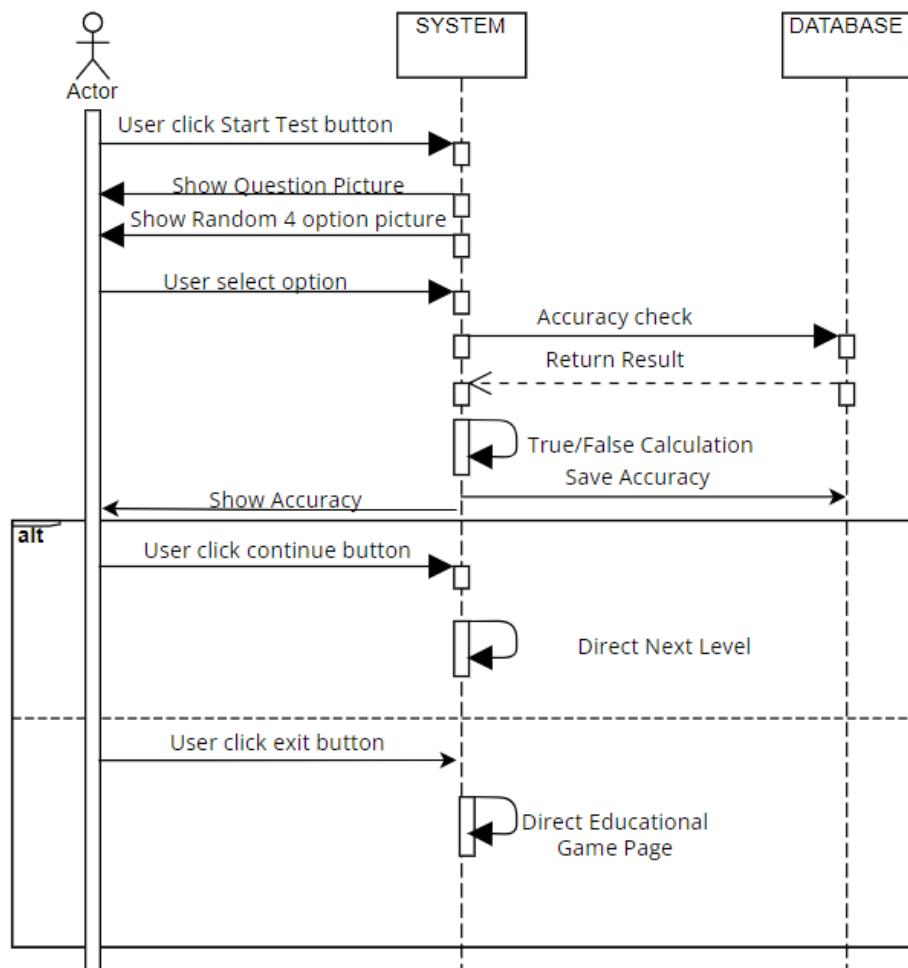
#### 4.3.7. Symmetry Test Sequence Diagram



#### 4.3.8. Educational Game Sequence Diagram



#### 4.3.9. Picture Matching Game Sequence Diagram



## 5. User Interface Detailed Design

### 5.1. Detailed Design of Each Component

#### 5.1.1. Login Page



#### 5.1.2. Registration Page



### 5.1.3. Privacy Agreement for Registration



### 5.1.4. Forgot Password Page

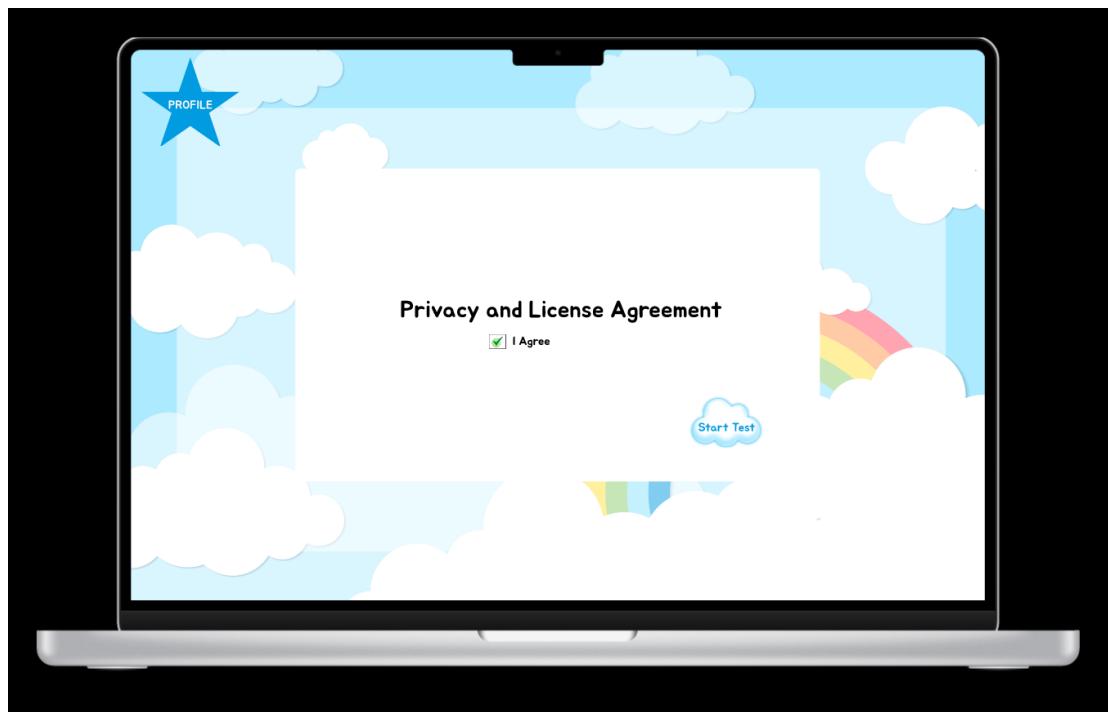




### 5.1.5. Main Page



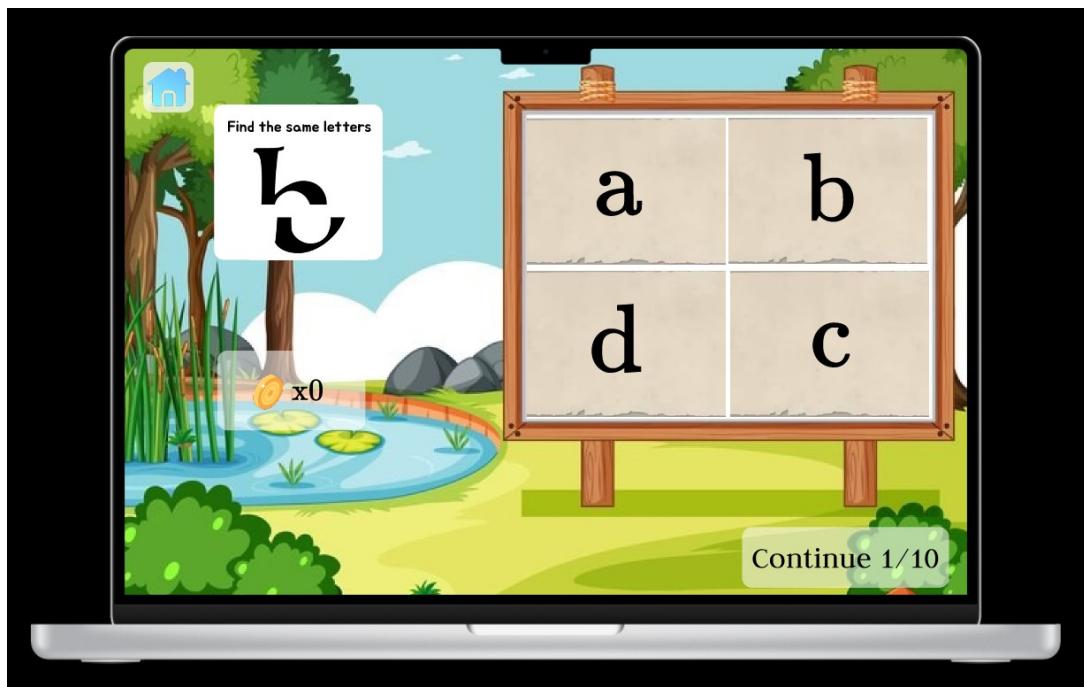
### 5.1.6. Privacy Agreement for Diagnostic Tests Page



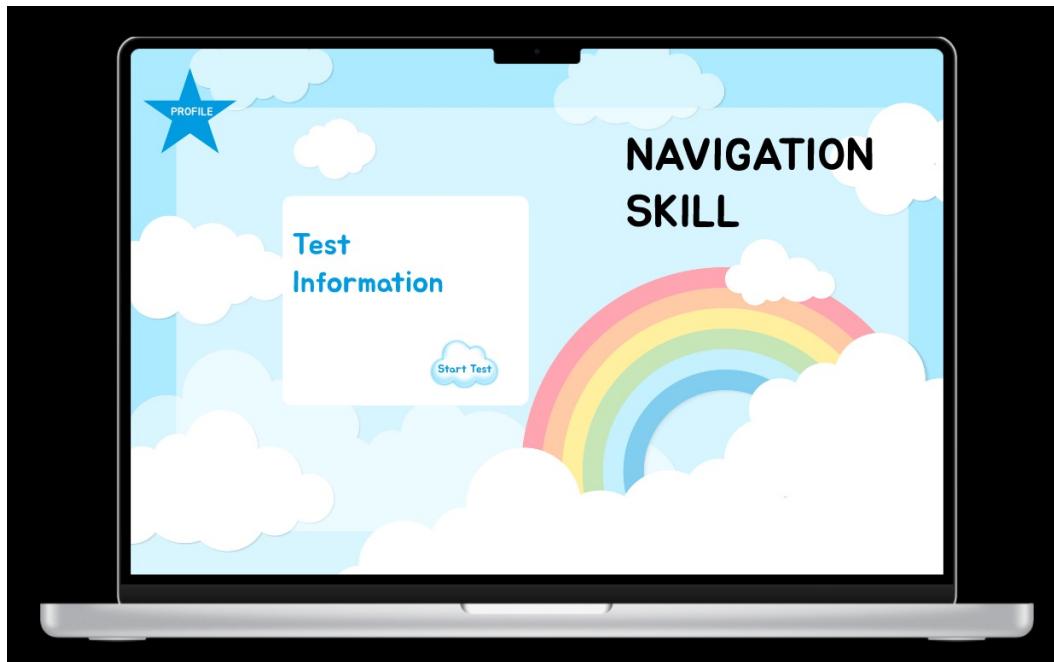
### 5.1.7. Letter Matching Test Information Page



### 5.1.8. Letter Matching Test Page



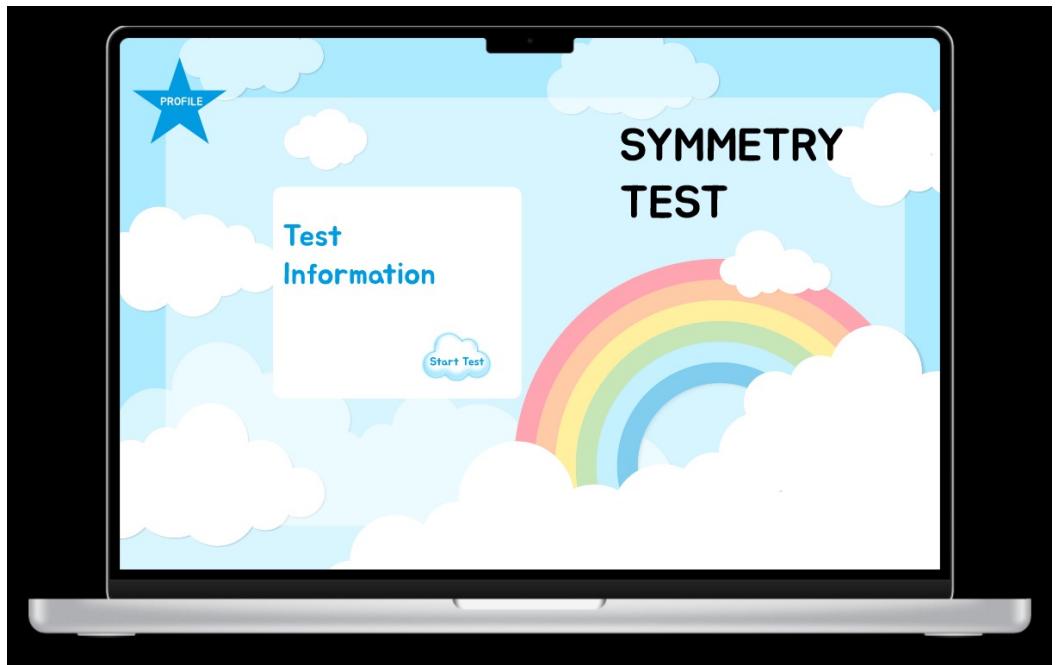
### 5.1.9. Navigation Skill Test Information Page



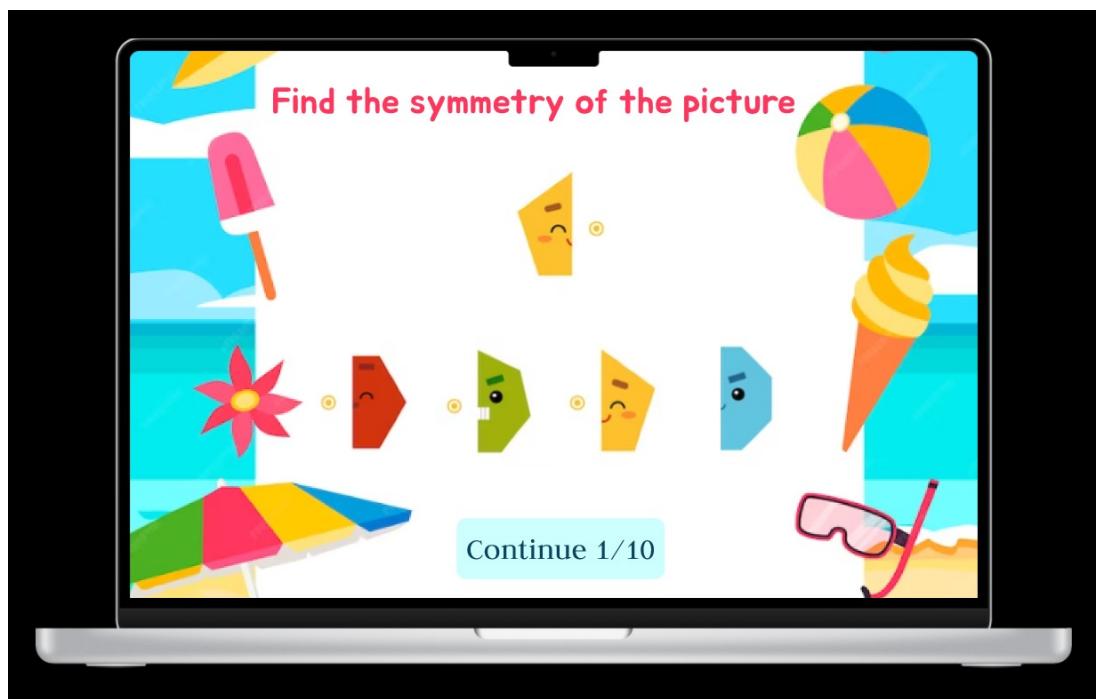
### 5.1.10. Navigation Skill Test Page



### 5.1.11. Symmetry Test Information Page



### 5.1.12. Symmetry Test Page



### 5.1.13. Educational Game Page



### 5.1.14. Picture Matching Game Page



## **6. Implementation**

### **6.1. Technologies Used**

- Frontend: Angular, HTML5, CSS3, JavaScript
- Backend: ASP.NET Core, Node.js
- Database: MySQL

## **7. Testing**

### **7.1. Testing Strategies**

- Unit testing for individual components.
- Integration testing for end-to-end system functionality.
- User acceptance tests.

## **8. Conclusion**

The Software Design Description (SDD) document is for the “Dyslexia Diagnostic Tests and Educational Games” project for dyslexic children aged four years and above. It outlines a comprehensive software development plan focused on early detection and support for children with dyslexia. The software has a user-friendly interface designed for children and includes diagnostic tests and skill-building activities. .NET Core uses a robust architecture that combines ReactJS, HTML, CSS, and MySQL with security and privacy considerations. Hosted in Azure, it provides scalability and performance. The document covers functional and non-functional requirements, user interface design, and implementation details, emphasizing accessibility, ease of use, and compliance with education and healthcare standards.

**User Class:** Represents the users of the system, with attributes for user information and methods for registration and login.

**Attributes:** UserID, Name, Email, Password, Age

**Methods:** Register(), Login()

**UserProfile Class:** Stores additional information about users, particularly children.

**Attributes:** ProfileID, UserID (Foreign Key), DyslexiaStatus, ProgressReports

**Methods:** UpdateProfile(), RetrieveProgress()

**DyslexiaDiagnosis Class:** Manages the dyslexia diagnosis process.

**Attributes:** DiagnosisID, UserID (Foreign Key), TestResults, Feedback

**Methods:** ConductTest(), RecordResults(), ProvideFeedback()

**Educational Game Class:** Represents each educational game.

**Attributes:** GameID, Name, Description, Goals

**Methods:** StartGame(), TrackProgress()

**GameSession Class:** Records individual game sessions and progress.

**Attributes:** SessionID, GameID (Foreign Key), UserID (Foreign Key), SessionScore, TimeSpent

**Methods:** RecordSession()

**Parent Class:** For parents monitoring their child's activities.

**Attributes:** ParentID, UserID (Foreign Key), Email, Reports

**Methods:** MonitorActivities(), ReceiveReports()

**Report Class:** Generates reports for parents or users.

**Attributes:** ReportID, UserID (Foreign Key), Content, Recommendations

**Methods:** GenerateReport()

**Interface Class:** Manages the user interface aspects.

**Attributes:** InterfaceID, Layout, NavigationElements

**Methods:** DisplayInterface(), Navigate()

**Security Class:** Handles data security and privacy.

**Attributes:** SecurityID, PrivacyPolicy, UserAgreements

**Methods:** EncryptData(), EnsurePrivacy()

**Support Class:** Provides in-app help and support.

**Attributes:** SupportID, FAQs, ContactInfo

**Methods:** OfferHelp(), ProvideSupport()