**COMSATS University Islamabad,   
Abbottabad Campus**

**SOFTWARE DESIGN DESCRIPTION   
(SDD DOCUMENT)**

**for**

**<babyshARk>**  
Version 1.0

***By***

**Mubeen Khalid CIIT/SP21-BSE-015/ATD**

**Syed Faizan Haider CIIT/SP21-BSE-001/ATD**

**Omiya Jadoon CIIT/SP21-BSE-021/ATD**

***Supervisor*Mam Bushra Mushtaq**

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

**Table of Contents**

**Revision History 3**

**1.** **Introduction 5**

**2.** **Design Methodology and software process model 5**

**3.** **System Overview 5**

3.1 Architectural Design 5

3.2 Process Flow/Representation 5

**4.** **Design Models [along with descriptions] 5**

**5.** **Data Design 6**

5.1 Data Dictionary 6

**6.** **Algorithm & Implementation 6**

**7.** **Software Requirements Traceability Matrix 6**

**8.** **Human Interface Design 7**

8.1 Screen Images 7

8.2 Screen Objects and Actions 7

**9.** **Appendix I 7**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Application Evaluation History**

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |

**Supervised by**

**<Supervisor’s Name>**

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Introduction**

Early childhood is a time when kids learn and develop their skills. Technology can help with this, and that's where "babyshARk" comes in. This app has fun features like "guided writing," where kids can show off their art skills. It also has a "learning" module that helps kids learn to pronounce letters of the alphabet. With "Alphabet detection and 3D models," learning becomes interactive and exciting using AR technology. After learning, kids can test their skills in the "evaluation" module, which uses 3D models and quizzes to check what they know. Unlike other apps that leave kids unsupervised, "babyshARk" includes "screen-time management." This feature lets parents set time limits on how long their kids can use the app. The scope includes:

* Development of an augmented reality (AR) based educational application for children aged 3 to 6.
* Availability for Android and iOS platforms, ensuring compatibility across different mobile devices.
* Alphabet Detection and 3D Model Presentation Module detects alphabets in the AR environment and triggers corresponding 3D models. This requires a downloadable PDF that contains target alphabet images for detection.
* Learning Module displays all alphabets to the kids and allows them to learn pronunciation by clicking on the alphabets, with voice lines played for each.
* The Evaluation Module presents interactive and engaging quizzes to kids to test their understanding of the alphabet and allows parents to view their child's results.
* Guide Writing Module guides kids in writing alphabets step by step in AR, providing real-time interaction.
* Screen Time Management Module allows parents to control and set time limits on their child’s usage of the app.
* UI development involves creating user interfaces for both Android and iOS platforms, ensuring that users can interact seamlessly on both types of devices.

The scope of the project does not extend beyond these functionalities and platforms.

**Design methodology and software process model**

**Design Methodology Choice:** Object-Oriented Programming (OOP)

In creating the "babyshARk" app, Object-Oriented Programming (OOP) is used to divide the app into smaller sections, such as alphabet recognition, 3D models, and quizzes. OOP keeps each section organized by encapsulating their internal workings, making updates easier. It also allows us to reuse code for different activities, ensuring that everything fits together seamlessly. This approach makes "babyshARk" adaptable for adding new educational features, while keeping the development process manageable.

**Software Process Model Choice:** Agile

Agile methods good for building "babyshARk" because they break work into small tasks that can be tested and improved often. This helps quickly adjust to user feedback and new technology. Agile encourages learning and teamwork by regularly reviewing progress and making changes as needed. By delivering parts of the app in short cycles (sprints), Agile shows progress early and makes it easier to plan releases. This approach helps "babyshARk" improve constantly and meet both technical needs and what users want in an educational AR app.

**System overview**

**Functionality:**

* **Alphabet Detection and 3D Model Presentation:** The app will detect and display 3D models corresponding to recognized alphabets in an AR environment, making learning engaging and interactive.
* **Learning and Pronunciation:** Children can learn to pronounce letters by interacting with them within the app. Each letter is accompanied by an audio pronunciation, allowing children to hear its sound when they touch it.
* **Guided Writing:** The app will guide children step-by-step in writing alphabets within the AR environment.
* **Interactive Quizzes:** Educational quizzes will test children's understanding of alphabets, with results accessible to parents.
* **Screen Time Management:** Parents can set limits on how long their children can use the app, promoting healthy screen time habits.

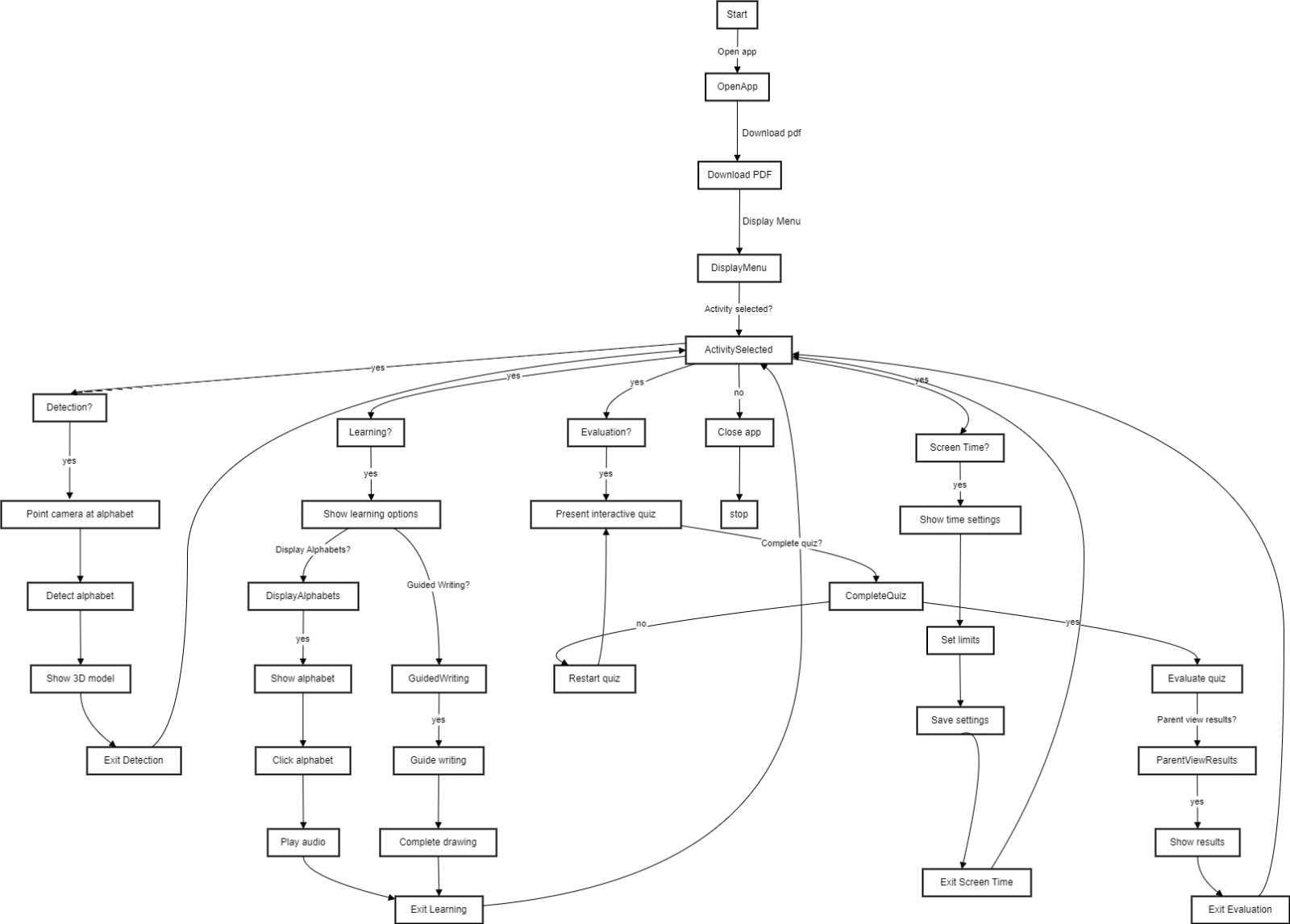
**Context:**

The "babyshARk" app helps kids learn with AR technology, making education more fun and interactive. It's designed to support parents by creating a safe digital space for their children to learn and grow. The app aims to engage kids early on and make learning enjoyable while using new technology.

**Design:**

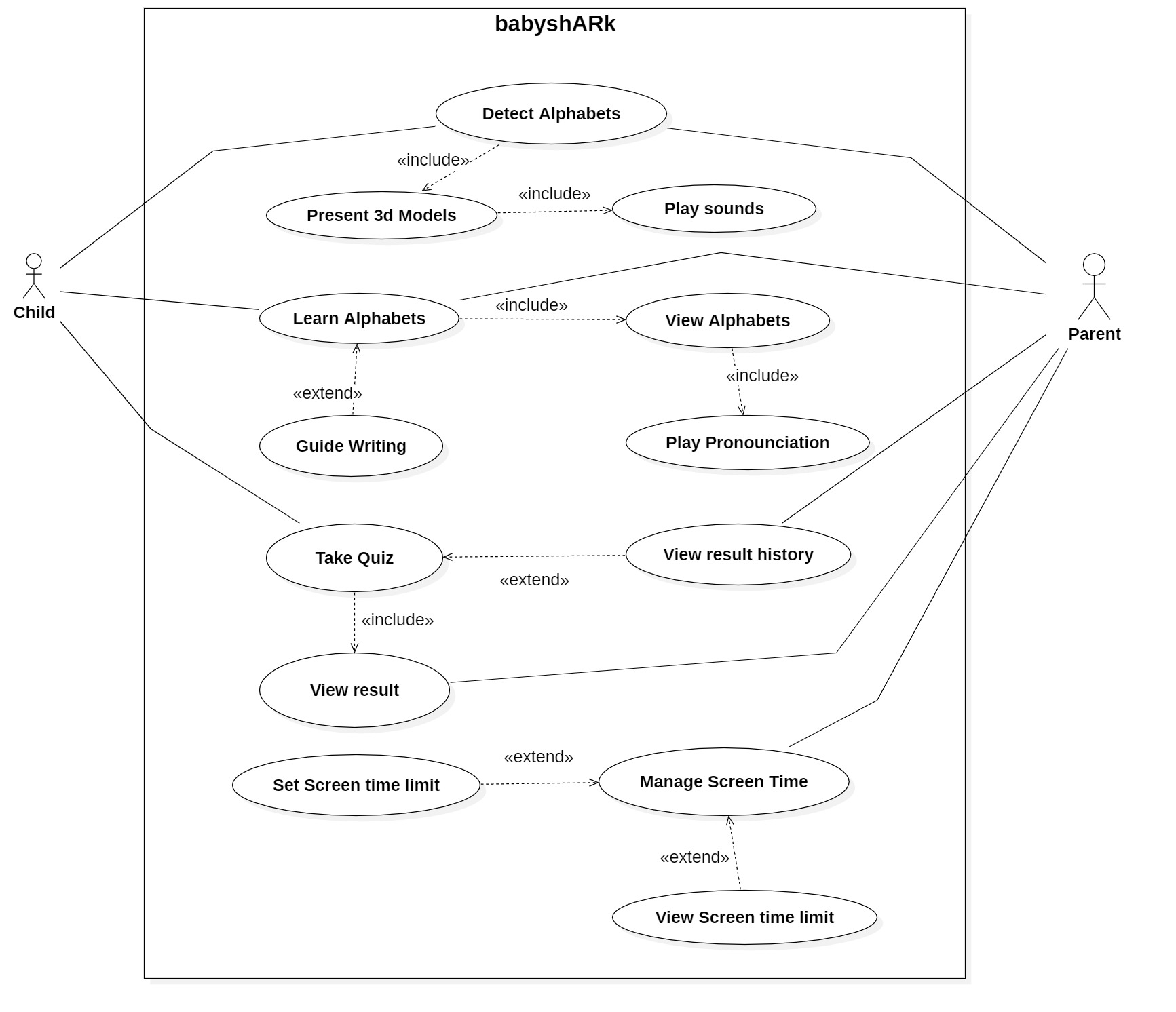
The "babyshARk" app utilizes Object-Oriented Programming (OOP) to structure its components into modular, reusable parts, facilitating ease of updates and feature enhancements. By employing Agile development methodologies, we can swiftly adapt to user feedback and emerging technologies. This iterative approach ensures continuous improvement of the app through incremental updates.

**Process flow/Representation**

**Design models [along with descriptions]**

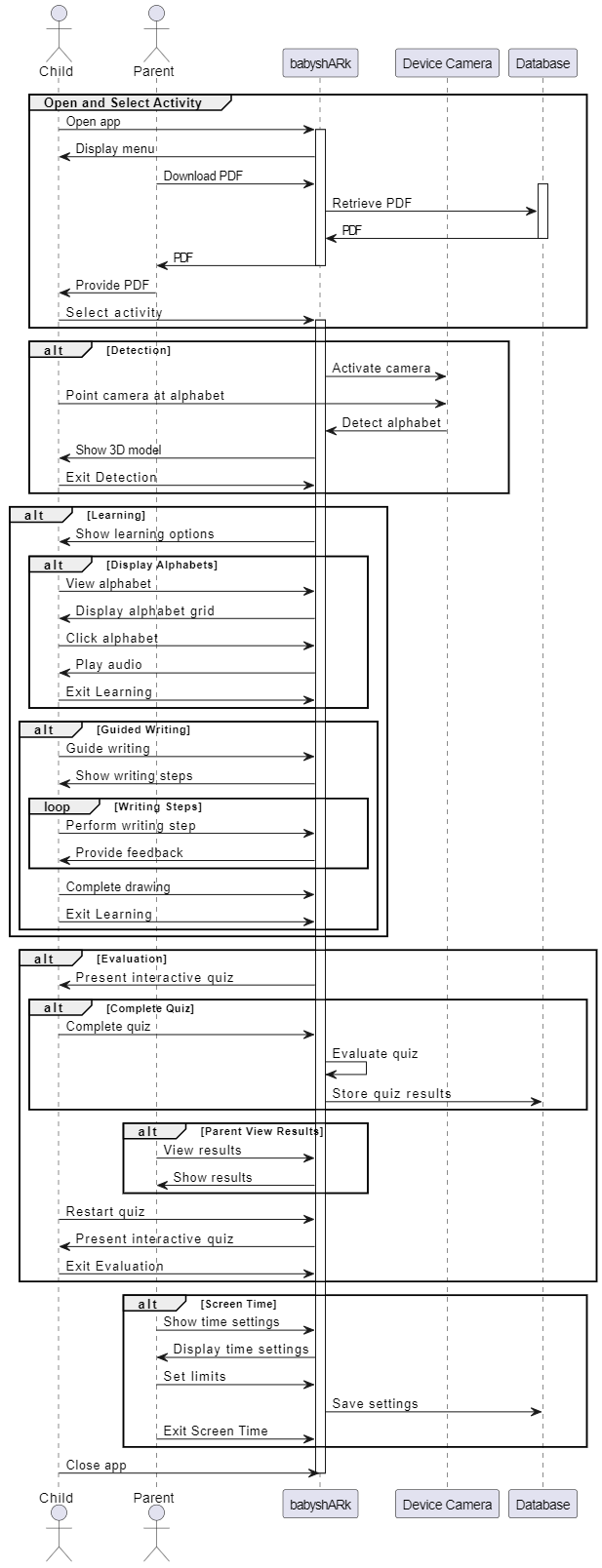
The applicable models may include:

**Use case diagram:**

**Description:**

* **Detect Alphabets**: The core functionality of detecting alphabets using augmented reality.
* **Present 3D Models**: Displays 3D models of the alphabets detected by the application.
* **Play Sounds**: Plays associated sounds for the detected alphabets.
* **Learn Alphabets**: The child learns the alphabet through various activities.
* **Guide Writing**: Provides guided writing exercises for the child to practice writing the alphabet.
* **Take Quiz**: The child can take quizzes to test their knowledge of the alphabet.
* **View Result**: Allows the parents to view the results of their child quizzes.
* **Set Screen Time Limil**: Allows the parent to set screen time limits for the child.
* **Manage Screen Time**: The parent manages the overall screen time for the application.
* **View Screen Time Limit**: Allows the parent to view the set screen time limits.
* **View Alphabets**: The child can view the alphabet in the application.
* **Play Pronunciation**: Plays the pronunciation of the alphabets for the child to learn.
* **View Result History**: The parent can view the history of the results from the quizzes taken by the child.

**Sequence Diagram:**

****

Description:

#### 1. ****Open and Select Activity****

* **Child Opens App**: The child initiates the interaction by opening the "babyshARk" app.
* **Display Menu**: The app displays a menu to the child.
* **Parent Downloads PDF**: The parent downloads a PDF from the app.
  + **Retrieve PDF**: The app retrieves the PDF from the database and sends it to the parent.
* **Parent Provides PDF**: The parent gives the downloaded PDF to the child.
* **Select Activity**: The child selects an activity from the displayed menu.

#### 2. ****Detection****

* **Activate Camera**: The app activates the device camera.
* **Point Camera at Alphabet**: The child points the camera at an alphabet.
* **Detect Alphabet**: The camera detects the alphabet, and the app shows a 3D model of it.
* **Exit Detection**: The child exits the detection activity.

#### 3. ****Learning****

* **Show Learning Options**: The app displays various learning options to the child.

##### Display Alphabets

* **View Alphabet**: The child selects to view an alphabet.
* **Display Alphabet Grid**: The app shows a grid of alphabets.
* **Click Alphabet**: The child clicks on an alphabet.
* **Play Audio**: The app plays the corresponding audio for the selected alphabet.
* **Exit Learning**: The child exits the learning activity.

##### Guided Writing

* **Guide Writing**: The child selects the guide writing option.
* **Show Writing Steps**: The app shows the steps for writing the alphabet.
* **Perform Writing Step**: The child performs each writing step.
* **Provide Feedback**: The app provides feedback for each step.
* **Complete Drawing**: The child completes the drawing.
* **Exit Learning**: The child exits the learning activity.

#### 4. ****Evaluation****

* **Present Interactive Quiz**: The app presents an interactive quiz to the child.

##### Complete Quiz

* **Complete Quiz**: The child completes the quiz.
* **Evaluate Quiz**: The app evaluates the quiz.
* **Store Quiz Results**: The app stores the quiz results in the database.

##### Parent View Results

* **View Results**: The parent views the results of the quiz.
* **Show Results**: The app displays the results to the parent.
* **Restart Quiz**: The child decides to restart the quiz.
* **Present Interactive Quiz**: The app presents the quiz again.
* **Exit Evaluation**: The child exits the evaluation activity.

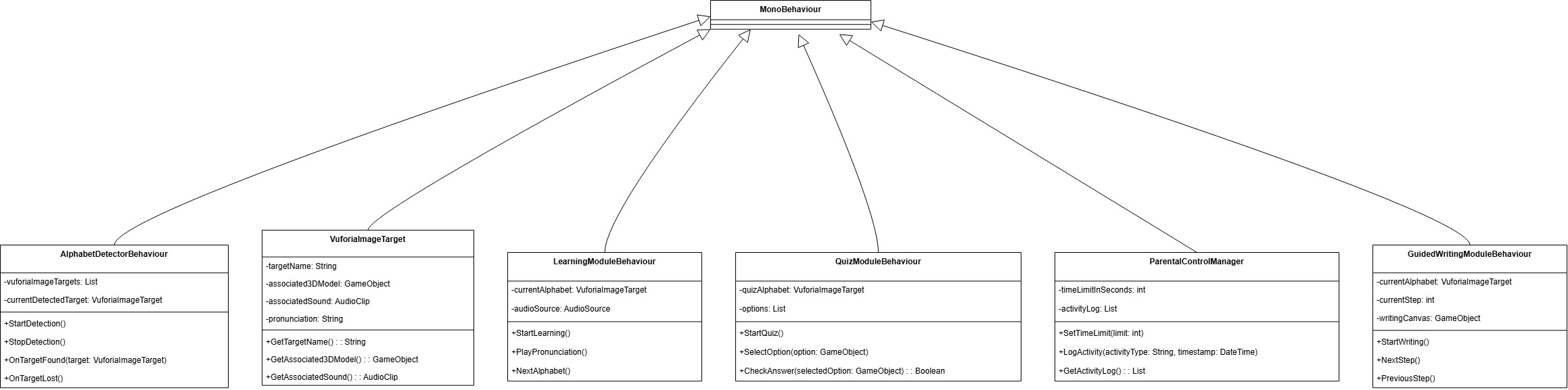
#### 5. ****Screen Time****

* **Show Time Settings**: The parent accesses the screen time settings.
* **Display Time Settings**: The app displays the time settings to the parent.
* **Set Limits**: The parent sets screen time limits.
* **Save Settings**: The app saves these settings to the database.
* **Exit Screen Time**: The parent exits the screen time settings.

#### 6. ****Close App****

* **Close App**: Finally, the child closes the app.

**Class Diagram:**

****

**Description:**

1. **AlphabetDetectorBehaviour (inherits from MonoBehaviour)**
   * **Attributes:**
     + vuforiaImageTargets: List
     + currentDetectedTarget: VuforiaImageTarget
   * **Methods:**
     + StartDetection()
     + StopDetection()
     + OnTargetFound(target: VuforiaImageTarget)
     + OnTargetLost()
2. **VuforiaImageTarget**
   * **Attributes:**
     + targetName: String
     + associated3DModel: GameObject
     + associatedSound: AudioClip
     + pronunciation: String
   * **Methods:**
     + GetTargetName() :: String
     + GetAssociated3DModel() :: GameObject
     + GetAssociatedSound() :: AudioClip
3. **LearningModuleBehaviour (inherits from MonoBehaviour)**
   * **Attributes:**
     + currentAlphabet: VuforiaImageTarget
     + audioSource: AudioSource
   * **Methods:**
     + StartLearning()
     + PlayPronunciation()
     + NextAlphabet()
4. **QuizModuleBehaviour (inherits from MonoBehaviour)**
   * **Attributes:**
     + quizAlphabet: VuforiaImageTarget
     + options: List
   * **Methods:**
     + StartQuiz()
     + SelectOption(option: GameObject)
     + CheckAnswer(selectedOption: GameObject) :: Boolean
5. **ParentalControlManager (inherits from MonoBehaviour)**
   * **Attributes:**
     + timeLimitInSeconds: int
     + activityLog: List
   * **Methods:**
     + SetTimeLimit(limit: int)
     + LogActivity(activityType: String, timestamp: DateTime)
     + GetActivityLog() :: List
6. **GuidedWritingModuleBehaviour (inherits from MonoBehaviour)**
   * **Attributes:**
     + currentAlphabet: VuforiaImageTarget
     + currentStep: int
     + writingCanvas: GameObject
   * **Methods:**
     + StartWriting()
     + NextStep()
     + PreviousStep()

### Relationships:

* All specific module classes (AlphabetDetectorBehaviour, LearningModuleBehaviour, QuizModuleBehaviour, ParentalControlManager, and GuidedWritingModuleBehaviour) inherit from MonoBehaviour.
* The AlphabetDetectorBehaviour class works with VuforiaImageTarget to detect and manage image targets.
* The LearningModuleBehaviour class uses VuforiaImageTarget to represent the current alphabet being learned and interacts with an AudioSource to play pronunciations.
* The QuizModuleBehaviour class uses VuforiaImageTarget for the quiz content and manages options for the quiz.
* The ParentalControlManager class handles time limits and activity logging for the system.
* The GuidedWritingModuleBehaviour class uses VuforiaImageTarget to guide writing steps on a writingCanvas.

**Human interface design**

**Screen images**

****

****

****

****

****

****

**Appendix I**

* How to design using UML (OOP): For guidance please follow the instructions mentioned in the link: http://agilemodeling.com/artifacts/
* How and when to design ER diagrams: For guidance please follow the instructions mentioned in the link:

<http://people.inf.elte.hu/nikovits/DB2/Ullman_The_Complete_Book.pdf>

* Data flow diagrams: For guidance please follow the instructions mentioned in the link and book:
  + http://www.agilemodeling.com/artifacts/dataFlowDiagram.htm
  + Software Engineering –A Practitioner’s approach by Roger Pressman
* Architecture diagram: For guidance please follow the instructions mentioned in the link and book:
  + Ian Sommerville – Software Engineering 9th Edition– Chapter 6