****

**COMSATS University Islamabad Abbottabad, Pakistan**

**babyshARk**

***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 Software Engineering (2021-2024)***

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

****

**COMSATS University Islamabad Abbottabad, Pakistan**

**babyshARk**

**A project presented to**

**COMSATS Institute of Information Technology, Islamabad**

**In partial fulfillment**

**of the requirement for the degree of**

***Bachelor of Science in Software Engineering (2021-2024)***

**By**

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

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

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

**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.

Mubeen Khalid Syed Faizan Haider

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

Omiya Jadoon

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

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (CS) “babyshARk” was developed by   
**Mubeen Khalid (CIIT/SP21-BSE-015)** , **Syed Faizan Haider (CIIT/SP21-BSE-001)** and **Omiya Jadoon** **(CIIT/SP21-BSE-021)** under the supervision of “MAM BUSHRA MUSHTAQ” and that in her opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Software Engineering.

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

**Supervisor**

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

**External Examiner**

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

**Head of Department**

**(Department of Computer Science)**

**EXECUTIVE SUMMARY**

babyshARk is a fun and educational app designed for young children aged 3 to 6. Using augmented reality (AR), this app makes learning more exciting by letting kids see letters come to life on the screen. The app is available for both Android and iOS, so parents can download it on most mobile devices. babyshARk helps kids learn the alphabet, practice writing letters, test their knowledge, and even gives parents control over screen time.

The app has several main features. In the **Alphabet Detection and 3D Models** section, children can learn letters by pointing the device’s camera at a printed alphabet. A PDF with all the letters is provided in the app, which parents must download and print. When a child points the camera at a letter on the printout, a 3D model of that letter appears on the screen, making learning both fun and interactive. In the **Learning** module, children can also tap on each letter to hear how it sounds, helping them learn to pronounce letters correctly.

The **Step-by-Step Writing** module guides kids in learning how to write each letter. It doesn’t teach drawing but helps them practice the alphabet in a fun way. When children are ready, they can test their skills in the **Evaluation** section, where they’ll answer questions and interact with the 3D letters. There is also a history feature in this module that keeps track of past quiz scores, allowing parents and kids to see their progress.

Finally, babyshARk has a **Screen Time Management** feature to help parents control how long their child spends on the app. Parents can set a time limit for app use, and if kids want extra time, they can solve simple math problems to earn it. This ensures that children get a balanced experience, mixing learning with playtime in a safe way.

Overall, babyshARk is a helpful, interactive tool for early childhood learning, making the alphabet learning fun and engaging.

**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 **“Mam Bushra Mushtaq”**. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

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

Mubeen Khalid Syed Faizan Haider

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

Omiya Jadoon

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

**ABBREVIATIONS**

|  |  |
| --- | --- |
| **AR** | Augmented Reality |
| **SDK** | Software Development Kit |
| **UC** | Use Case |
| **3D** | 3 Dimensional |
| **SRS** | Software Requirement Specification |
| **SDD** | Software Design Document |

**TABLE OF CONTENTS**

[**1 Introduction** 11](#_Toc520773734)

[1.1 Brief 11](#_Toc520773735)

[1.2 Relevance to Course Modules 11](#_Toc520773736)

[1.3 Project Background 11](#_Toc520773737)

[1.4 Literature Review 11](#_Toc520773738)

[1.5 Analysis from Literature Review (in the context of your project) 11](#_Toc520773739)

[1.6 Methodology and Software Lifecycle for This Project 11](#_Toc520773740)

[1.6.1 Rationale behind Selected Methodology 11](#_Toc520773741)

[**2** **Problem Definition** 12](#_Toc520773742)

[2.1 Problem Statement 12](#_Toc520773745)

[2.2 Deliverables and Development Requirements 12](#_Toc520773746)

[2.3 Current System (if applicable to your project) 12](#_Toc520773747)

[**3** **Requirement Analysis** 13](#_Toc520773748)

[3.1 Use Cases Diagram(s) 13](#_Toc520773750)

[3.2 Detailed Use Case 13](#_Toc520773751)

[3.3 Functional Requirements 13](#_Toc520773752)

[3.4 Non-Functional Requirements 13](#_Toc520773753)

[**4 Design and Architecture** 14](#_Toc520773754)

[4.1 System Architecture 14](#_Toc520773759)

[4.2 Data Representation [Diagram + Description] 14](#_Toc520773760)

[4.3 Process Flow/Representation 14](#_Toc520773761)

[4.4 Design Models [along with descriptions] 14](#_Toc520773762)

[**5** **Implementation** 15](#_Toc520773763)

[5.1 Algorithm 15](#_Toc520773765)

[5.2 External APIs 15](#_Toc520773766)

[5.3 User Interface 15](#_Toc520773767)

[**6 Testing and Evaluation** 16](#_Toc520773768)

[6.1 Manual Testing 16](#_Toc520773771)

[6.1.1 System testing 16](#_Toc520773772)

[6.1.2 Unit Testing 16](#_Toc520773773)

[6.1.3 Functional Testing 17](#_Toc520773774)

[6.1.4 Integration Testing 17](#_Toc520773775)

[6.2 Automated Testing: 18](#_Toc520773776)

[Tools used: 18](#_Toc520773777)

[**7** **Conclusion and Future Work** 19](#_Toc520773778)

[7.1 Conclusion 19](#_Toc520773780)

[7.2 Future Work 19](#_Toc520773781)

[**8** **References** 20](#_Toc520773782)

**LIST OF FIGURES**

Fig 3.1 Use Case Diagram 9

Fig 4.1 Activity Diagram 9

Fig 4.3 Class Diagram 9

**LIST OF TABLES**

Table 1.1 Table1 8

Table 2.1 Table 2 9

**1.Introduction**

This chapter gives an overview of the babyshARk project, an educational app using augmented reality (AR) designed for children aged 3 to 6. The app aims to make learning fun and interactive by using technology to help young kids learn the alphabet. With features like detecting alphabets, helping kids pronounce them, guided writing, and taking quizzes, babyshARk helps children build basic skills in an enjoyable way. By using AR, babyshARk provides a new way to teach the alphabet that is both effective and entertaining.

This chapter will also talk about the background of the project, existing research on AR in education, and the methods used to develop the app.

**1.1.Brief**

The babyshARk project aims to create an educational app for children aged 3 to 6, using augmented reality (AR) to make learning fun and interactive. The app's main goal is to help kids learn the alphabet by recognizing letters, pronouncing them correctly, and practicing writing. With features like detecting letters, pronunciation guides, guided writing, and quizzes, babyshARk offers a complete learning experience. It is designed to grab children's attention and motivate them to learn through play.

To build this exciting app, several tools and technologies are being used. Unity is the main platform for developing the AR features, allowing kids to see letters as 3D models when they point their device at printed letters. In this MS Visual Studio Code is used for writing the code and C# is chosen as the programming language because it is easy to use, and Vuforia is used for AR tracking to ensure the letter detection works smoothly.

Agile method is followed for this project, which helps to stay flexible and quickly respond to feedback from users. This approach lets us break the development into small cycles, making it easier to improve the app based on what users say.

In this report, we discuss the project's background, how AR can help kids learn, and look at existing studies that support our ideas. Each chapter covers different parts of our development process, including the decisions we made, the challenges we faced, and the solutions we found. Our goal is to create a friendly and engaging app that helps young children learn and enjoy reading.

**1.2.Relevance to Course Modules**

The babyshARk project is closely related to several courses studied during the Bachelor of Computer Science (BCS) program. Here’s how it connects to different areas:

**Software Development:** The project involves programming the app using C#, which is a key skill learned in software development courses. Students apply their knowledge of coding and software design principles to create a functional application.

**Mobile Application Development:** Since babyshARk is designed to run on Android and iOS platforms, it directly relates to mobile app development courses. Students learn about the unique challenges of developing apps for different operating systems and the best practices for user interface design.

**Augmented Reality (AR):** The app utilizes augmented reality features, linking it to courses focused on AR technology. Students explore how to integrate AR functionalities into applications using tools like the Vuforia SDK, which is essential for creating immersive learning experiences.

**Database Management:** The project incorporates a local database for storing and managing user data, related it to database management courses. Students apply their knowledge of data storage solutions and learn how to efficiently retrieve and update information.

**User Experience (UX) Design:** Creating an engaging and user-friendly app requires an understanding of UX design principles. This aspect of the project connects to courses that teach students how to design applications that meet user needs and enhance overall satisfaction.

**Project Management:** The development of babyshARk follows an agile methodology, which relates to project management courses. Students learn how to break down tasks, work in teams, and adapt to changes throughout the development process.

Overall, the babyshARk project integrates various concepts from the BCS curriculum, allowing students to apply their theoretical knowledge to a practical, real-world application.

**1.3.Project Background**

The *babyshARk* project is focused on creating an educational app that uses augmented reality (AR) to help young children learn the alphabet in an engaging way. The idea behind the *babyshARk* project is based on the fact that early childhood is a very important time for learning. During these years, children are eager to learn and easily pick up new skills. Research shows that what kids learn when they are young can affect their future success in school and life. By using fun and interactive methods like augmented reality, *babyshARk* wants to encourage young children to explore and enjoy learning. The project aims to make learning letters exciting, helping kids build important skills in a way that they will remember and love.

Augmented reality is a technology that overlays digital information, like images and sounds, onto the real world. In the case of *babyshARk*, when children point their device’s camera at a printed letter, the app recognizes it and displays a fun 3D model related to that letter. For example, pointing at the letter "A" might show an animated apple. This interactive experience helps children connect letters with words and sounds, making it easier for them to remember and understand the alphabet.

Research shows that young children are naturally curious and eager to learn through play. By incorporating AR into the learning experience, *babyshARk* aims to captivate children's attention and motivate them to explore letters in a playful environment. The app also allows parents to monitor their child's learning journey, ensuring a balance between screen time and educational activities.

Overall, the project seeks to combine education and technology to create a fun and effective tool for early literacy development.

**1.4.Literature Review**

The *babyshARk* project fits well with current trends in educational technology, especially using augmented reality (AR) for young children's learning. Research shows that AR can make learning more exciting and help kids pay attention better. A study by Akçayir and Akçayir (2017) found that AR apps can help children understand and remember information more easily because they provide fun and interactive experiences that regular teaching methods often lack. When children can see and touch 3D objects, they are more likely to remember what they learn, making AR a great tool for teaching basic skills like reading.

Many existing products use AR to help children learn. For example, apps like *Osmo* and *ABCmouse* mix physical and digital learning to teach kids important skills. These apps often include games and activities that keep children interested while they learn. The success of these products shows that technology is becoming more accepted in education, and there is a growing need for tools that can engage young learners.

Recent research also highlights how important play is for early childhood education. The National Association for the Education of Young Children (NAEYC) states that learning through play is vital for children's growth because it encourages them to explore, be creative, and think critically. The *babyshARk* project aims to include these play-based elements by making learning the alphabet fun and enjoyable. By combining playful learning with AR technology, the project hopes to provide a new and exciting way for young children to learn and develop important skills.

**1.5.Analysis from Literature Review (in the context of your project)**

There are some systems that already exist that offer some of the functionalities that this project has to offer. Some of them are:

1. **Color Quest AR**: This application is only available on IOS, and the kid can color different fruits which then can be augmented, and children can dance and interact with them.
2. **AR Drawing App:** This application lets you draw sketches, and you can upload your own images and trace them using your camera.
3. **English Alphabet AR:** It’s an AR application that shows images and asks questions related to the images along with some AR based flash cards.
4. **AR Flashcards**: This app offers basic flashcards for learning letters but lacks interactive activities, making it less engaging for kids.
5. **ABC Alphabet Tracing**: This app lets children trace letters but does not include interactive AR features, making the tracing experience feel boring.

|  |  |  |
| --- | --- | --- |
| **Application Name** | **Weakness** | **Proposed Project Solution** |
| Color Quest AR | Although the children can color different fruits and interact with them there are certain limitations such as you must pay for some fruits also the coloring is not AR based you can see the fruit afterwards. | No such payment is required for unlocking any features the project offers all the functionalities completely free of cost. |
| AR Drawing App | On opening the app, the interface is not appealing and even though you can upload images that you want to trace the quality is not that good, it feels like the image is faded and transparent and nothing else. | This application provides a kid-friendly and appealing interface with high quality models and tracing sequences where kids can easily trace and draw without any difficulty. |
| English Alphabet AR | There are complete sentences that some children are not able to understand. Even though the questions are for children they are still hard to understand and if you want to draw or interact with the elements in AR you have to go their website and scan their flash cards. This process is rather time consuming and not easy to understand. | No such complexity is involved in this project. Everything is visible and easy to understand providing its kids an easy and exciting kid interface. |
| AR Flashcards | This app provides flashcards for learning letters, but the interactions are limited. Kids can only see the flashcard images and do not have engaging activities linked to them. | In *babyshARk*, we include interactive quizzes that let kids apply what they learn. This approach helps reinforce their understanding while making learning fun. |
| ABC Alphabet Tracing | This app allows children to trace letters but does not offer interactive AR experiences. The tracing can feel dull because there is no immediate feedback or engagement. | Our project combines tracing with AR elements, so when children trace a letter, they can see a related 3D model and hear its sound. This added interaction keeps kids engaged and motivated to learn. |

**1.6.Methodology and Software Lifecycle for This Project**

The methodology and SDLC model selected for babyshARk is:

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

**Software Process Model Choice:** Agile.

**1.6.1.Rationale behind Selected Methodology**

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.

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.



**2.Problem Definition**

The problematic statement and its solutions are given in this chapter. Also, deliverable and development requirements are explained below in this chapter.



**2.1.Problem Statement**

The proposed software aims to address the challenge of engaging children in educational activities that are both entertaining and informative. Traditional learning methods like books can often fail to capture the attention of kids, leading to a lack of interest. Also, many young children face challenges when learning the alphabet because most educational apps available today are not engaging or easy for them to use. Some apps require payments to access all features, which can make it difficult for families who need free learning options. Additionally, many apps rely on simple pictures and basic activities, which may not hold children’s attention or keep them excited about learning. Without interactive features or easy designs, children can quickly lose interest or feel confused, making it hard for them to build early reading skills. We are developing "babyshARk" to fill the gap in AR educational tools that are truly made for kids.

Other apps might offer some features like AR based drawing or alphabet learning, ours is designed to be more interactive, with a focus on simplicity and ease of use. Through re-implementation, we expect to learn more about children's interactions with AR technology, the process of creating content that balances education with excitement, and the importance of intuitive design in app development.

Our system will solve the identified problems by introducing an AR-based platform that transforms learning into an interactive experience. Step-by-step drawing modules will guide children through drawing lessons. Alphabet recognition will be taught using interactive 3D models, while different models based on the alphabets will also be presented. Evaluations based on these models will reinforce learning outcomes and provide feedback. Screen time management features will ensure a balanced approach to learning that parents can set, while audio-assisted modules will teach kids to pronounce different alphabets. By integrating these elements, our system will offer a comprehensive educational tool that is both enjoyable and effective for kids.

2.2.Deliverables and Development Requirements

Deliverables

* **Completed *babyshARk* App:** A fully functional augmented reality educational app available on both Android and iOS platforms.
* **Alphabet Detection Module:** Recognizes letters and displays corresponding 3D models.
* **Step-by-Step Drawing Module:** Guides children in writing letters with AR assistance.
* **Interactive Quiz Module:** Offers quizzes to test knowledge.
* **History Module:** Shows previous quiz scores and history for tracking progress.
* **Screen Time Management:** Allows parents to set limits on app usage, including a reward system for additional time through educational challenges.
* **SRS:** Software Requirement Specification document.
* **SDD:** Software Design Document.

**Development Requirements:**

* **Unity:** For developing the AR experience and overall app functionalities.
* **C#:** The primary programming language for coding the app features.
* **Vuforia:** For augmented reality tracking and interactions.
* **MS Visual Studio Code:** The IDE used for writing and debugging code.
* **Agile Methodology:** To allow flexibility in development, enabling quick adjustments based on user feedback and ongoing testing.
* **User Testing:** Conduct regular testing sessions with children and parents to gather feedback on usability and engagement.
* **Functionality Testing:** Ensure all app features work correctly and smoothly before the final launch.

**2.3.Current System (if applicable to your project)**

One existing system that is applicable to our project is **English Alphabet AR**. This app is designed to help children learn the alphabet by providing images and asking questions related to each letter. It utilizes augmented reality to enhance the learning experience by displaying images associated with letters. However, the app has several limitations that hinder its effectiveness.

The **English Alphabet AR** app presents complete sentences and questions, which can be confusing for young children who are still developing their language skills. Additionally, to interact with the AR elements, users must scan flashcards from the app's website, making the process time-consuming. This extra step can frustrate both children and parents, leading to a less engaging experience.

In comparison, our project, *babyshARk*, aims to address these shortcomings by simplifying the learning process.



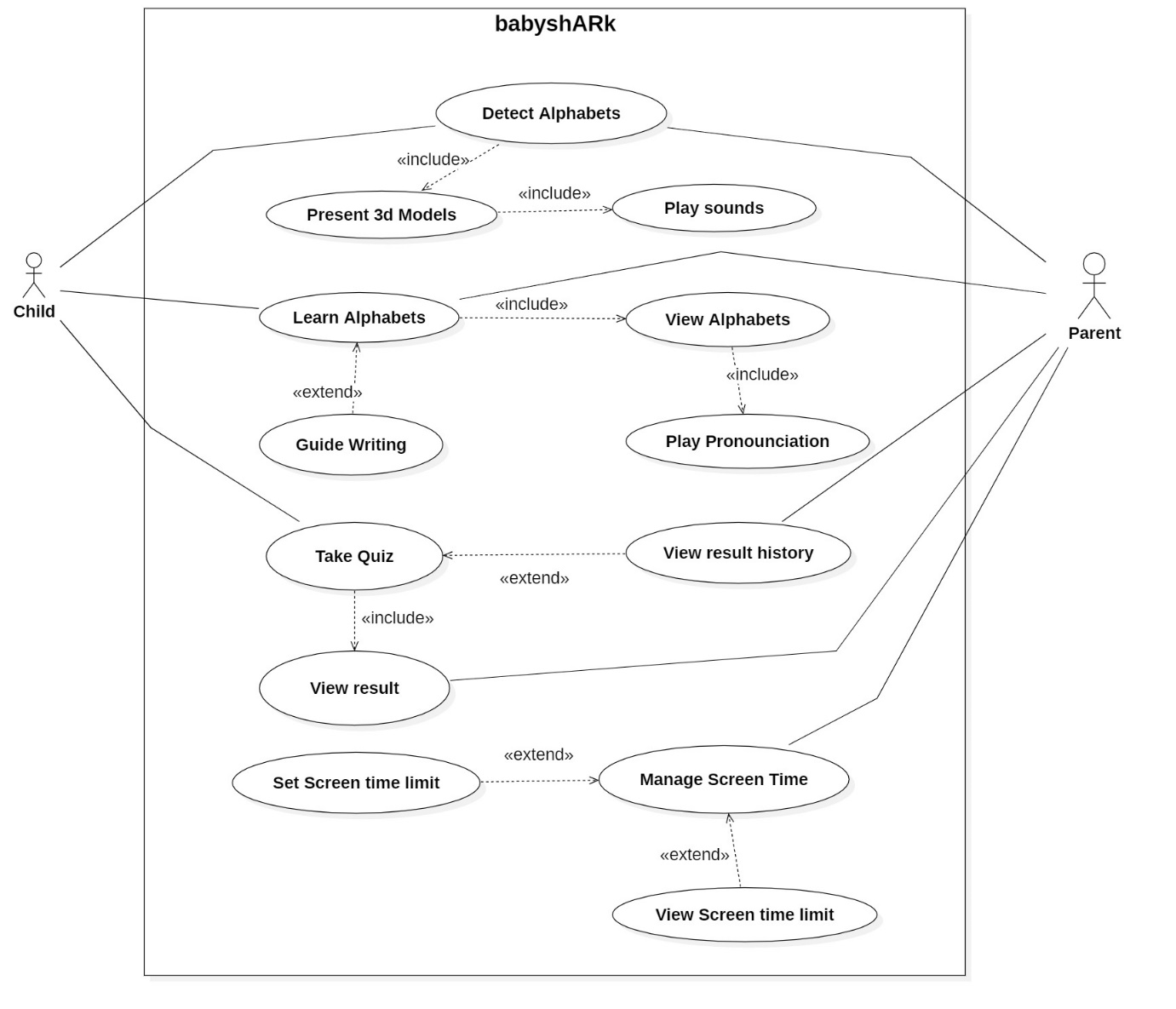
*Figure 2.1: English Alphabet AR Figure 2.2: English Alphabet AR*

**3.Requirement Analysis**

The following parts of the Software Requirements Specification (SRS) report should be included in this chapter.



**3.1.Use Cases Diagram(s)**



*Figure 3.1: Use case Diagram*

**3.2.Detailed Use Case**

Use Case UC-1: Detect Alphabets

|  |  |
| --- | --- |
| **Use Case ID** | UC-1 |
| **Use Case Name** | Detect Alphabets |
| **Actors** | Primary actor: Child |
| **Description** | The child uses the AR environment to identify and interact with alphabets. The system processes the camera input to recognize the shapes of alphabets using image targets and provides feedback. |
| **Trigger** | The child points the device camera at printed alphabet target. |
| **Precondition** | 1. The camera is active and pointing at an appropriate target.  2. The camera is only being accessed by the current app.  3. The environment light is good enough. |
| **Postconditions** | The system successfully detects and identifies the alphabet viewed through the camera. |
| **Normal Flow** | 1. The child opens the app.  2. The child points the device camera at the alphabets  3. The system processes the camera input and detects the alphabet.  4. The system highlights or labels the detected alphabets on the screen.  5. The child receives visual feedback of the identified alphabets. |
| **Assumption** | The child has access to a device with a camera and the app is installed. |

**Use Case UC-2: Present 3D Models**

|  |  |
| --- | --- |
| **Use Case ID** | UC-2 |
| **Use Case Name** | Present 3D Models |
| **Actors** | Primary actor: Child |
| **Description** | The child views corresponding 3D models triggered by detected alphabets. The system uses the detected alphabet target to present an interactive 3D model related to that alphabet. |
| **Trigger** | Detection of alphabet target in AR environment. |
| **Precondition** | 1. Detection of alphabets in the AR environment.  2. The camera is not shaky. |
| **Postconditions** | The 3D model is successfully augmented. |
| **Normal Flow** | 1. The system detects an alphabet in the AR environment.  2. The system presents the 3D model on the screen. |

**Use Case UC-3: Learn Alphabets**

|  |  |
| --- | --- |
| **Use Case ID** | UC-3 |
| **Use Case Name** | Learn |
| **Actors** | Primary actor: Child |
| **Description** | The child learns the alphabet by interacting with the app. The app provides visual and auditory cues to help the child understand and memorize each alphabet. |
| **Trigger** | The child selects the learning module from the app menu. |
| **Precondition** | The application is running and the learning module is accessed. |
| **Postconditions** | All alphabets are displayed with options to view and hear pronunciation. |
| **Normal Flow** | 1. The child opens the app and selects the learning mode.  2. The app displays alphabets on the screen.  3. The child interacts with an alphabet (e.g., clicking on it).  4. The app provides visual and auditory feedback.  5. The child repeats the interaction with other alphabets. |

**Use Case UC-4: View Alphabets**

|  |  |
| --- | --- |
| **Use Case ID** | UC-4 |
| **Use Case Name** | View Alphabets |
| **Actors** | Primary actor: Child |
| **Description** | The child views the list of alphabets in the learning module. |
| **Trigger** | The child accesses the learning module. |
| **Precondition** | The application is running, and the learning module is accessed. |
| **Postconditions** | A list of all alphabets is displayed. |
| **Normal Flow** | 1.The child accesses the learning module.  2.The application displays a list of all alphabets. |
| **Assumption** | The application data is correctly loaded. |

**Use Case UC-5: Play Pronunciation**

|  |  |
| --- | --- |
| **Use Case ID** | UC-5 |
| **Use Case Name** | Play Pronunciation |
| **Actors** | Primary actor: Child |
| **Description** | The child learns the pronunciation of alphabets by clicking on them. The system plays a voice line for the selected alphabet. |
| **Trigger** | The child clicks on an alphabet in the learning module. |
| **Precondition** | The app must have audio capabilities. |
| **Postconditions** | The child hears the correct pronunciation of the alphabet. |
| **Normal Flow** | 1. The child selects an alphabet in the learning module.  2. The system plays the pronunciation audio for the selected alphabet.  3. The child listens to the pronunciation. |
| **Assumption** | The app includes prerecorded pronunciations for each alphabet. |

**Use Case UC-6: Guide Writing**

|  |  |
| --- | --- |
| **Use Case ID** | UC-6 |
| **Use Case Name** | Guide Writing |
| **Actors** | Primary actor: Child |
| **Description** | The child is guided step-by-step to write alphabets in the AR environment. |
| **Trigger** | The child selects the writing module from the app menu. |
| **Precondition** | The application is running, and the writing module is accessed. |
| **Postconditions** | The child successfully learns to write the selected alphabets. |
| **Normal Flow** | 1. The child opens the app and selects the writing mode.  2. The app displays the first step of writing an alphabet.  3. The child follows the guided instructions to write the alphabet.  4. The app proceeds to the next step until the alphabet is completed.  5. The child repeats the process for other alphabets. |

**Use Case UC-7: Take Quiz**

|  |  |
| --- | --- |
| **Use Case ID** | UC-7 |
| **Use Case Name** | Take Quiz |
| **Actors** | Primary actor: Child |
| **Description** | The child participates in interactive quizzes to test their knowledge of the alphabet. |
| **Trigger** | The child selects the quiz module from the app menu. |
| **Precondition** | The application is running, and the quiz module is accessed. |
| **Postconditions** | Child successfully completes quizzes.  The results of the quiz are recorded. |
| **Normal Flow** | 1. The child opens the app and selects the quiz mode.  2. The app presents quiz questions related to the alphabet.  3. The child answers the quiz questions.  4. The app displays the final quiz score. |

**Use Case UC-8: View Result**

|  |  |
| --- | --- |
| **Use Case ID** | UC-8 |
| **Use Case Name** | View Result |
| **Actors** | Primary actor: Parents  Secondary actor: Child |
| **Description** | The parent views the results of the quizzes taken by the child. |
| **Trigger** | The parent selects the result view option from the app menu. |
| **Precondition** | 1. The application is running, and the results module is accessed. |
| **Postconditions** | The results of the child's quizzes are displayed. |
| **Normal Flow** | 1. The parent selects the result view option from the app menu.  2. The application displays the results of the child's quizzes. |

**Use Case UC-9: View Result History**

|  |  |
| --- | --- |
| **Use Case ID** | UC-9 |
| **Use Case Name** | View Result History |
| **Actors** | Primary actor: Parents |
| **Description** | The parent views the history of all quiz results. |
| **Trigger** | The parent selects the result history option from the app menu. |
| **Precondition** | 1. The application is running, and the results history module is accessed.  2. The child has attempted 1 or more quizzes earlier. |
| **Postconditions** | The history of all quiz results is displayed. |
| **Normal Flow** | 1. The parent selects the result history option from the app menu.  2. The application displays the history of all quiz results. |
| **Assumption** | The application has access to all stored quiz results. |

**Use Case UC-10: Manage Screen Time**

|  |  |
| --- | --- |
| **Use Case ID** | UC-10 |
| **Use Case Name** | Manage Screen Time |
| **Actors** | Primary actor: Parents |
| **Description** | Parent manages screen time limits for the child. |
| **Trigger** | Parent accesses the screen time management feature. |
| **Precondition** | The application is running, and the screen time management module is accessed. |
| **Postconditions** | Parent successfully manages screen time limits. |
| **Normal Flow** | 1. The parent selects the screen time management option from the app menu.  2. The application displays the current screen time settings and options to manage screen time. |

**Use Case UC-11: Set Screen Time Limits**

|  |  |
| --- | --- |
| **Use Case ID** | UC-11 |
| **Use Case Name** | Set Screen Time Limit |
| **Actors** | Primary actor: Parent |
| **Description** | The parent sets specific time limits on how long the child can use the app. |
| **Trigger** | Parent accesses the screen time setting feature. |
| **Precondition** | The application is running, and the screen time settings are accessed. |
| **Postconditions** | Parent successfully sets screen time limits. |
| **Normal Flow** | 1. The parent accesses the screen time settings.  2. The application displays options to set time limits.  3. The parent sets the desired screen time limit.  4. The application saves the new screen time limit. |

**Use Case UC-12: View Screen Time**

|  |  |
| --- | --- |
| **Use Case ID** | UC-12 |
| **Use Case Name** | View Screen Time Limit |
| **Actors** | Primary actor: Parent |
| **Description** | The parent views the current screen time of the application. |
| **Trigger** | The parent selects the option to view screen time from the app menu. |
| **Precondition** | The application is running, and the screen time viewing option is accessed. |
| **Postconditions** | Current screen time settings and usage are displayed. |
| **Normal Flow** | 1. The parent selects the option to view screen time from the app menu.  2. The application displays the current screen time settings and usage. |

**Use Case UC-13: Play Sounds**

|  |  |
| --- | --- |
| **Use Case ID** | UC-13 |
| **Use Case Name** | Play Sounds |
| **Actors** | Primary actor: Child |
| **Description** | The system plays the sounds related to different objects which are augmented that makes the interaction more fun. |
| **Trigger** | Detection of alphabet target in AR environment. |
| **Precondition** | 1. Detection of alphabets in the AR environment.  2. The camera is not shaky. |
| **Postconditions** | The sound is successfully played. |
| **Normal Flow** | 1. The system detects an alphabet in the AR environment.  2. The system presents the 3D model on the target image.  3. The system plays a sound related to the presented model. |

**3.3.Functional Requirements**

**Functional Requirement for UC-1: Detect Alphabets**

|  |  |
| --- | --- |
| **Identifier** | FR-UC1-1 |
| **Title** | Detect Alphabets |
| **Requirement** | The system shall detect and identify alphabets when the child points the device camera at printed alphabet targets. |
| **Source** | Child |
| **Rationale** | To allow children to interact with and learn alphabets using AR technology. |
| **Business Rule (if required)** | - |
| **Dependencies** | - |
| **Priority** | High |

**Functional Requirement for UC-2: Present 3D Models**

|  |  |
| --- | --- |
| **Identifier** | FR-UC2-1 |
| **Title** | Present 3D Models |
| **Requirement** | Upon detecting an alphabet, the system shall augment the corresponding 3D model on the target. |
| **Source** | System |
| **Rationale** | To provide a visual representation of objects to learn alphabets. |
| **Business Rule (if required)** | - |
| **Dependencies** | FR-UC1-1 (Detect Alphabets) |
| **Priority** | High |

**Functional Requirement for UC-3: Learn Alphabets**

|  |  |
| --- | --- |
| **Identifier** | FR-UC3-1 |
| **Title** | Learn Alphabets |
| **Requirement** | The system shall display alphabets and allow children to interact with them to learn their shapes and sounds. |
| **Source** | Child |
| **Rationale** | To facilitate the learning of alphabets through methods like playing pronunciations, and guided writing. |
| **Business Rule (if required)** | - |
| **Dependencies** | - |
| **Priority** | High |

**Functional Requirement for UC-4: View Alphabets**

|  |  |
| --- | --- |
| **Identifier** | FR-UC4-1 |
| **Title** | View alphabets |
| **Requirement** | The system shall display a list of all alphabets in alphabetical order. |
| **Source** | Child |
| **Rationale** | To provide an easy navigation of alphabets |
| **Business Rule (if required)** | - |
| **Dependencies** | - |
| **Priority** | High |

**Functional Requirement for UC-5: Hear Pronunciation**

|  |  |
| --- | --- |
| **Identifier** | FR-UC5-1 |
| **Title** | Hear Pronunciation |
| **Requirement** | When the child clicks on an alphabet, the system shall play the corresponding pronunciation audio. |
| **Source** | Child |
| **Rationale** | To help children learn the correct pronunciation of alphabets. |
| **Business Rule (if required)** | - |
| **Dependencies** | FR-UC3-1 (Learn Alphabets) |
| **Priority** | High |

**Functional Requirement for UC-6: Guide Writing**

|  |  |
| --- | --- |
| **Identifier** | FR-UC6-1 |
| **Title** | Guide Writing |
| **Requirement** | The system shall provide step-by-step instructions for drawing or writing alphabets. |
| **Source** | System |
| **Rationale** | To teach children how to draw or write alphabets accurately. |
| **Business Rule (if required)** | - |
| **Dependencies** | - |
| **Priority** | Medium |

**Functional Requirement for UC-7: Take Quiz**

|  |  |
| --- | --- |
| **Identifier** | FR-UC7-1 |
| **Title** | Take Quiz |
| **Requirement** | The system shall present interactive quizzes to the child and provide feedback on their answers. |
| **Source** | Child |
| **Rationale** | To test the child's knowledge of alphabets. |
| **Business Rule (if required)** | - |
| **Dependencies** | FR-UC3-1 (Learn Alphabets) |
| **Priority** | Medium |

**Functional Requirement for UC-8: View Result**

|  |  |
| --- | --- |
| **Identifier** | FR-UC8-1 |
| **Title** | View result |
| **Requirement** | The system shall allow the parent to access the results view option. |
| **Source** | Parent |
| **Rationale** | To enable parents to monitor their child's progress |
| **Business Rule (if required)** | - |
| **Dependencies** | FR-UC7-1 (Take Quiz) |
| **Priority** | Medium |

**Functional Requirement for UC-9: View Result History**

|  |  |
| --- | --- |
| **Identifier** | FR-UC9-1 |
| **Title** | View Result History |
| **Requirement** | The system shall allow the parent to access the result history option. |
| **Source** | Parent |
| **Rationale** | To provide a comprehensive view of the child's progress over time |
| **Business Rule (if required)** | - |
| **Dependencies** | FR-UC7-1 (Take Quiz) |
| **Priority** | Medium |

**Functional Requirement for UC-10: Manage Screen Time**

|  |  |
| --- | --- |
| **Identifier** | FR-UC10-1 |
| **Title** | Manage Screen Time |
| **Requirement** | The system shall allow parents to manage and monitor screen time limits for their children. |
| **Source** | Parent |
| **Rationale** | To help parents control and limit the amount of time children spend on the app |
| **Business Rule (if required)** | - |
| **Dependencies** | - |
| **Priority** | High |

**Functional Requirement for UC-11: Set Screen Time Limits**

|  |  |
| --- | --- |
| **Identifier** | FR-UC11-1 |
| **Title** | Set Screen Time Limits |
| **Requirement** | The system shall allow parents to set specific screen time limits for their children. |
| **Source** | Parent |
| **Rationale** | To ensure children use the app within the allocated screen time limits. |
| **Business Rule (if required)** | - |
| **Dependencies** | - |
| **Priority** | High |

**Functional Requirement for UC-12: View Screen Time**

|  |  |
| --- | --- |
| **Identifier** | FR-UC12-1 |
| **Title** | View Screen Time |
| **Requirement** | The system shall allow the parent to access the screen time viewing option. |
| **Source** | Parent |
| **Rationale** | To monitor the child's screen time usage |
| **Business Rule (if required)** | - |
| **Dependencies** | - |
| **Priority** | High |

**Functional Requirement for UC-13: Play Sound**

|  |  |
| --- | --- |
| **Identifier** | FR-UC13-1 |
| **Title** | Play Sound |
| **Requirement** | The system plays the sounds related to different objects which are augmented that makes the interaction more fun. |
| **Source** | System |
| **Rationale** | To make learning more entertaining for the child. |
| **Business Rule (if required)** | - |
| **Dependencies** | - |
| **Priority** | Medium |

**3.4.Non-Functional Requirements**

**Useability Requirement**

USE-1: The system shall provide a child-friendly interface with easy-to-understand icons and minimal text to facilitate navigation for kids.

USE-2: The system shall provide simple, one-touch controls for accessing main features like detecting alphabets, viewing 3D models, and taking quizzes.

**Performance Requirements**

PER-1: The system shall quickly detect and identify alphabets when the child points the device camera at an appropriate target.

PER-2: The system shall promptly load and present 3D models after an alphabet is detected, audio pronunciation shall play immediately when the user clicks on an alphabet, quiz results shall be displayed promptly after the quiz is completed by the child.

**Reliability Requirements**

REL-1: The system shall be available 99.9% of the time during usage hours.

REL-2: The system shall ensure that no data, including quiz results and learning progress, is lost during interactions.

**Portability Requirements**

POR-1: The system shall be compatible with major mobile operating systems, specifically iOS and Android.

**4.Design and Architecture**

Th Software Design Description (SDD) is explained in this chapter.

**4.1.Data Representation [Diagram + Description]**

*Figure 4.1: Class diagram*

**Description:**

**4.2. Process Flow/Representation**

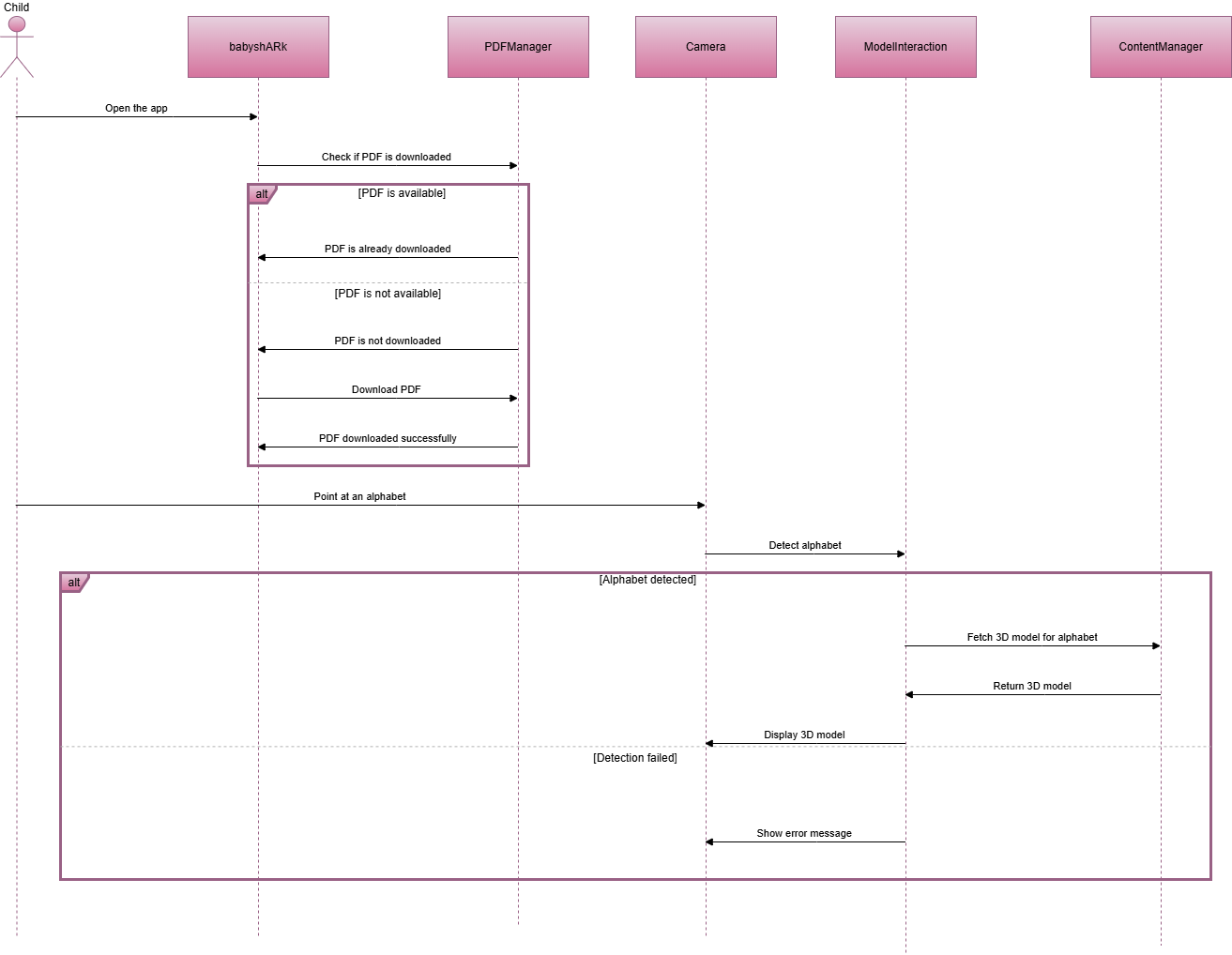
A diagram of a company

Description automatically generated

*Figure 4.2: Activity Diagram*

**4.3. Design Models [along with descriptions]**

**Sequence diagram for Alphabet Detection:**

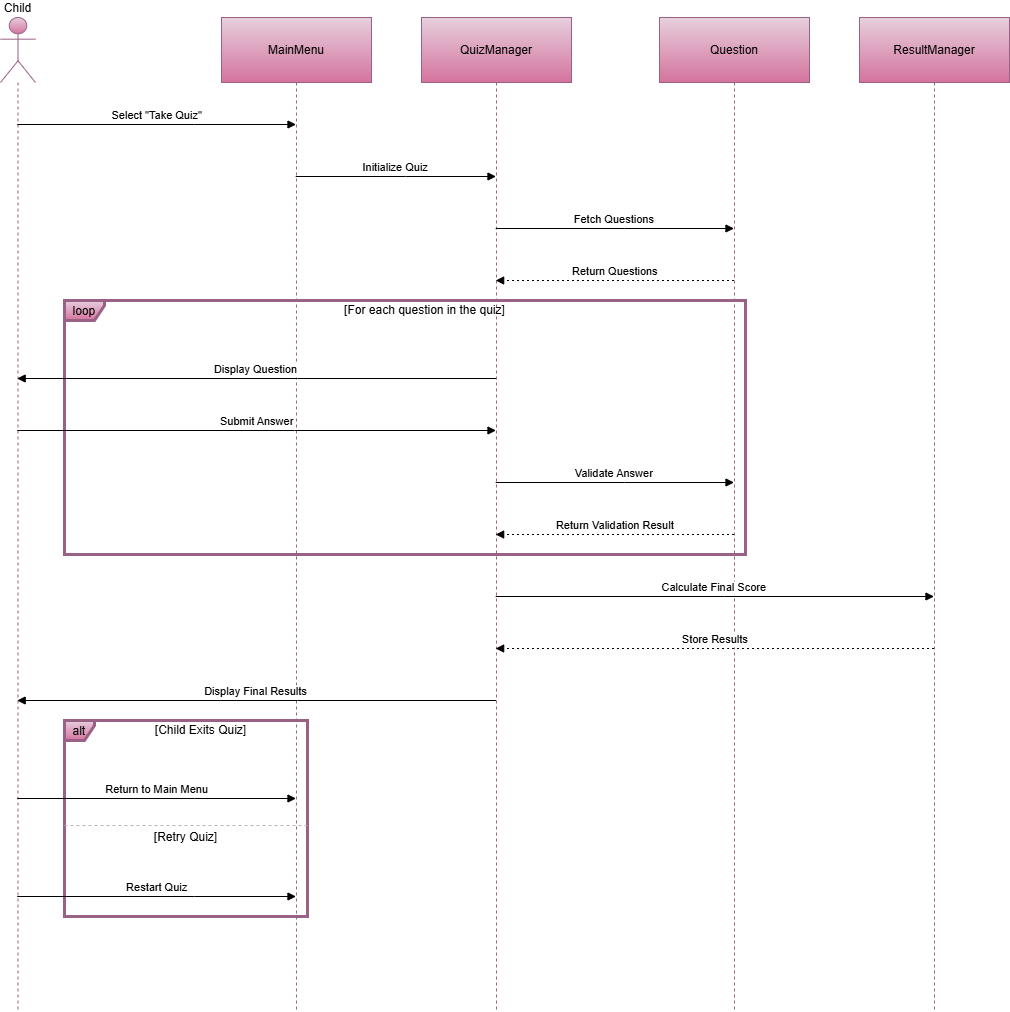
*Figure 4.3: Sequence diagram for Alphabet Detection*

**Description:**

In this sequence diagram, the child first launches the app. The app checks if the PDF, which contains instructions or content, has already been downloaded. If the PDF is available, the app confirms that it is already downloaded. If not, the app initiates a download of the PDF and confirms once the download is complete.

Next, the child points the camera at an alphabet. The camera attempts to detect the alphabet and sends this information to the ModelInteraction module. If the alphabet is successfully detected, the ModelInteraction module requests the corresponding 3D model from the ContentManager, which then returns the model. The 3D model is displayed by the camera, allowing the child to interact with it. If the alphabet is not detected, an error message is shown to the child, indicating the failure. This flow ensures that the app provides a smooth, interactive experience for the child when learning the alphabet

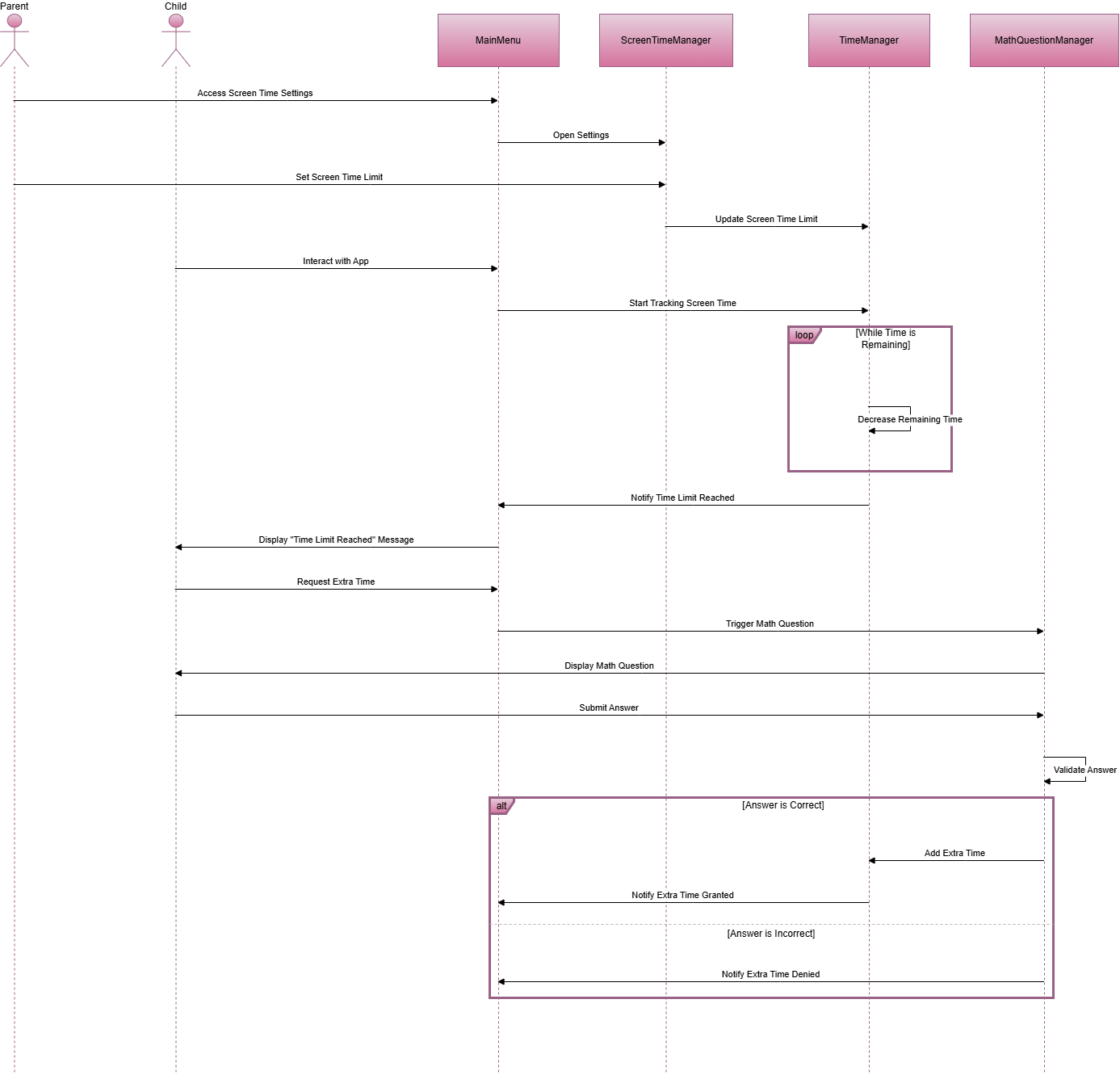


**Sequence diagram for Evaluation:**

*Figure 4.4: Sequence diagram for Evaluation*

**Description:**

The sequence begins when the child selects the "Take Quiz" option from the Main Menu, prompting the Main Menu to ask the QuizManager to start the quiz. The QuizManager then requests the set of questions from the Question module, which returns the questions. For each question in the quiz, the QuizManager displays it to the child, who then submits their answer. The QuizManager checks the answer by sending it to the Question module, which validates it and sends back whether the answer is correct or incorrect. After all questions are answered, the QuizManager asks the ResultManager to calculate the final score, and the results are stored. Finally, the QuizManager shows the child their final score. The child can choose to either return to the Main Menu or restart the quiz, depending on their choice. This flow ensures that the quiz process is smooth and organized, giving the child an interactive and engaging experience.

**Sequence diagram for Screen Time management:**

*Figure 4.5: Sequence diagram for Screen time management*

**Description:**

The process begins when the parent accesses the Main Menu to adjust the screen time settings. The parent sets a screen time limit, which is then updated in the ScreenTimeManager and passed to the TimeManager to track the remaining time. When the child uses the app, the TimeManager starts tracking the screen time and decreases the remaining time as the child interacts with the app.

Once the screen time runs out, the TimeManager notifies the Main Menu, which then displays a "Time Limit Reached" message to the child. The child can request extra time, prompting the Main Menu to ask the MathQuestionManager to present a math question. The child attempts to solve the math question, and the MathQuestionManager validates the answer. If the answer is correct, extra time is added by the TimeManager, and the Main Menu notifies the child that the extra time has been granted. If the answer is incorrect, the MathQuestionManager informs the Main Menu to deny the request for extra time. This flow ensures that the child can earn more time only by answering a math question correctly.

**Sequence diagram for Learning:**

*Figure 4.6: Sequence diagram for Learning*

**Description:**

1. **Implementation**

This chapter will discuss implementation details supported by UML diagrams (if applicable). You will not put your source code here. Any of the following sections may be included based on your project.

1. 1. **Algorithm**

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

* 1. **User Interface**

Details about user interface with descriptions.

**6.Testing and Evaluation**

The testing for the babyshARk project is performed manually and automatedly to ensure that the system functions correctly and meets all specified requirements. This chapter may include the following sections



**6.1 Manual Testing**

**6.1.1 System Testing**

System testing was performed on the **babyshARk** application to ensure it works as expected and meets the requirements. The goal was to check that all the features, such as alphabet detection, 3D model display, audio playback, quizzes, guided writing and screen time management, function correctly. System testing also helped find any hidden errors or bugs that could affect the app’s performance.

Each module was tested to verify smooth interactions. For example, pointing the camera at an alphabet was tested to ensure the correct 3D model and audio appeared. Overall, system testing confirmed that the app is ready for use.

**6.1.2 Unit Testing**

Once the system has been successfully developed, the following unit tests were performed to ensure each individual module works correctly.

* **Unit Testing 1:** PDF Download  
  **Testing Objective:** To ensure the PDF download functionality works correctly.

*Table 6.1: PDF Download Unit Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| **1** | Verify PDF download on Getting Started screen | Trigger PDF download | PDF successfully downloaded and stored locally | Pass |

* **Unit Testing 2:** Alphabet Detection  
  **Testing Objective:** To ensure the camera detects alphabets correctly and displays corresponding 3D models.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| 1 | Verify detection of alphabet 'A' via camera | Camera points at 'A' | Alphabet 'A' is detected, and 3D model appears | Pass |
| 2 | Verify detection of alphabet 'B' via camera | Camera points at 'B' | Alphabet 'B' is detected, and 3D model appears | Pass |

*Table 6.2: Alphabet Detection Unit Testcase*

* **Unit Testing 3:** Play Alphabet Pronunciation  
  **Testing Objective:** To ensure that the correct audio plays when clicking on an alphabet.

*Table 6.3: Play Alphabet Pronunciation Unit Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| 1 | Verify audio for alphabet 'C' | Click on alphabet 'C' | Audio for alphabet 'C' is played successfully | Pass |
| 2 | Verify audio for alphabet 'D' | Click on alphabet 'D' | Audio for alphabet 'D' is played successfully | Pass |

* **Unit Testing 4:** Start Quiz  
  **Testing Objective:** To ensure that the quiz module starts and displays questions correctly.

*Table 6.4: Start Quiz Unit Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| 1 | Verify quiz module start | Open quiz screen | Quiz screen displays the first question | Pass |

* **Unit Testing 5:** Solve Math Question for Extra Time  
  **Testing Objective:** To ensure that solving the math question grants extra screen time correctly.

*Table 6.5: Solve Math Question Unit Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| 1 | Verify correct answer grants extra time | Solve math question: Correct Answer = 10 | Extra time granted (e.g., +5 minutes) | Pass |
| 2 | Verify incorrect answer denies extra time | Solve math question: Incorrect Answer | Access denied; no extra time granted | Pass |

* + 1. **Functional Testing**

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

* **Functional Testing 1:** Launch Main Menu  
  **Objective**: To ensure that the app launches successfully, and the Main Menu is displayed.

*Table 6.6: Launch Main Menu Functional Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| 1 | Launch the app and navigate to the Main Menu | App launch | The Main Menu screen is displayed with Interact, Quiz, and Exit buttons | Pass |

* **Functional Testing 2:** Alphabet Detection in Interact Module  
  **Objective**: To ensure that the app detects alphabets and displays corresponding 3D models.

*Table 6.7: Alphabet Detection Functional Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| 1 | Point camera at letter ‘A’ | Camera input: letter ‘A’ | The corresponding 3D model of the letter ‘A’ appears on the screen | Pass |

* **Functional Testing 3:** Alphabet Pronunciation  
  **Objective**: To ensure audio pronunciation works when tapping on a letter.

*Table 6.8: Alphabet Pronunciation Functional Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| 1 | Tap on letter ‘B’ | Tap input: letter ‘B’ | Audio pronunciation for the letter ‘B’ is played | Pass |

* **Functional Testing 4:** Quiz Module - Correct Answer  
  **Objective**: To ensure correct feedback and score update when answering a question correctly.

*Table 6.9: Quiz Correct Answer Functional Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| 1 | Answer quiz question correctly | Correct answer selected | “Correct” feedback is displayed, and the score is updated | Pass |

* **Functional Testing 5:** Quiz Module - Incorrect Answer  
  **Objective**: To ensure incorrect feedback when answering a question wrongly.

*Table 6.10: Quiz Incorrect Answer Functional Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| 1 | Answer quiz question incorrectly | Wrong answer selected | “Try Again” feedback is displayed, and no score is added | Pass |

* **Functional Testing 6:** Screen Time Management  
  **Objective**: To ensure that the screen time limit triggers the math question correctly.

*Table 6.11: Screen Time Management Functional Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| 1 | Exceed the usage time limit | Time exceeded | A math question is displayed to gain additional time | Pass |

* **Functional Testing 7:** Correct Math Question Answer  
  **Objective**: To ensure additional time is granted when answering the math question correctly.

*Table 6.12: Correct Math Question Functional Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| 1 | Answer math question correctly | Correct answer (e.g., 5+3=8) | Additional time is granted, and app usage continues | Pass |

* **Functional Testing 8:** Incorrect Math Question Answer  
  **Objective**: To ensure app usage locks when answering the math question incorrectly.

*Table 6.13: Incorrect Math Question Functional Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| 1 | Answer math question incorrectly | Wrong answer (e.g., 5+3=9) | App locks usage, and no additional time is granted | Pass |

* **Functional Testing 9:** PDF Download  
  **Objective**: To ensure that the PDF file for alphabet recognition can be downloaded successfully.

*Table 6.14: PDF Download Functional Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| 1 | Click on "Download PDF" button | Button click: Download PDF | PDF file is successfully downloaded | Pass |

* **Functional Testing 10:** Exit the App  
  **Objective**: To ensure the app exits successfully when the Exit button is clicked.

*Table 6.15: Exit App Functional Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test Case/Test Script** | **Attribute and Value** | **Expected Result** | **Result** |
| 1 | Click on the “Exit” button | Button click: Exit | App closes successfully | Pass |

* + 1. **Integration Testing**

Table 6.16 shows the integration testing

*Table 6.16: Integration Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Launch the app and display the Main Menu | App launch | The Main Menu screen is displayed with Interact, Quiz, and Exit buttons | Pass |
| 2 | Access the Interact Module | Click on the “Interact” button | The camera opens, and the alphabet detection module starts | Pass |
| 3 | Point camera at an alphabet | Camera input: letter ‘A’ | The 3D model of letter ‘A’ appears on the screen | Pass |
| 4 | Tap on a 3D alphabet model | Tap input: letter ‘A’ | Audio pronunciation for letter ‘A’ is played | Pass |
| 5 | Access the Quiz Module | Click on the “Quiz” button | The quiz interface opens with the first question | Pass |
| 6 | Answer quiz question correctly | Correct answer selected | “Correct” feedback is displayed, and the score is updated | Pass |
| 7 | Answer quiz question incorrectly | Incorrect answer selected | “Try Again” feedback is displayed, and no score is updated | Pass |
| 8 | Exceed the screen time limit | Time exceeded | A math question is displayed to gain additional time | Pass |
| 9 | Answer math question correctly | Correct answer provided | Additional screen time is granted | Pass |
| 10 | Answer math question incorrectly | Incorrect answer provided | App usage is locked until next session | Pass |
| 11 | Download PDF for alphabet recognition | Click on “Download PDF” button | The PDF file is successfully downloaded | Pass |
| 12 | Exit the app | Click on “Exit” button | The app closes successfully | Pass |

* 1. **Automated Testing:**

Automated testing tools are used to ensure the babyshARk application performs consistently across its modules. These tools help validate functional requirements, non-functional requirements, and specific test cases, reducing manual testing effort and increasing efficiency.

* + 1. **Tools used:**

Table 6.17 shows the

*Table 6.17: Tools used*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tool Name** | **Tool Description** | **Applied on [list of related tests cases / FR / NFR]** | **Results** |
| Selenium | A web automation tool used for functional testing of UI components. | Main Menu navigation, Interact Module, Quiz Module (FR1, FR2, FR3) | Pass |
| JUnit | A Java-based unit testing framework for testing individual classes/methods. | Alphabet Detection logic, Math Question logic (Test Cases 3, 8, 9) | Pass |
| Appium | An open-source tool for automating mobile applications. | Alphabet detection via camera input, 3D models rendering (FR2) | Pass |
| Postman | A tool used to test APIs for backend integration and data retrieval. | Screen Time data validation, PDF download verification (FR4, NFR1) | Pass |
| TestComplete | A UI testing tool for functional and regression testing. | User navigation, Quiz Module feedback (Test Cases 5, 6, 7) | Pass |
| Jenkins | A CI/CD tool used for automating the testing pipeline. | Integration of all modules during deployment (All FRs) | Pass |

**7.Conclusion and Future Work**

This chapter concludes the project and highlights future work.



**7.1.Conclusion**

**7.2.Future Work**

**8.References**

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

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