

PlayCode



Group Members

Muhammad Abdullah Butt (01-131162-036)

Adeel Ahmad (01-131162-002)

Supervisor: Engr. Iqra Basharat

A Final Year Project submitted to the Department of Software Engineering,
Faculty of Engineering Sciences, Bahria University, Islamabad in the partial
fulfillment for the award of degree in Bachelor of Software Engineering

June 2021

THESIS COMPLETION CERTIFICATE

Student Name: Muhammad Abdullah Butt Enrolment No: 01-131162-036

Student Name: Adeel Ahmad Enrolment No: 01-131162-002

Programme of Study: Bachelor of Software Engineering

Project Title: PlayCode

It is to certify that the above students' project has been completed to my satisfaction and to my belief, its standard is appropriate for submission for evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at 9 % that is within the permissible limit set by the HEC. I have also found the thesis in a format recognized by the department.

Supervisor's Signature:  _____

Date: 09-06-2021 Name: Iqra Basharat

CERTIFICATE OF ORIGINALITY

This is to certify that the intellectual contents of the project **PlayCode** are the product of my/our own work except, as cited properly and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare, extend and/or implement the earlier work. Further, this work has not been submitted by me/us previously for any degree, nor it shall be submitted by me/us in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorities the University to cancel my/our degree.

Student Name: Muhammad Abdullah Butt

Enrolment No: 01-131162-036

Signature:  _____

Date: 01-06-2021

Student Name: Adeel Ahmad

Enrolment No: 01-131162-002

Signature:  _____

Date: 01-06-2021

Abstract

Learning programming languages is becoming increasingly popular and researchers want to find ways to teach it more effectively. There is a perceived problem with programming teaching and students having difficulty understanding the reasoning behind how programs operate. Therefore, they must be encouraged in some way. Programming should be made fun, and students must feel motivated to practice. Gamification refers to repurposing the concepts of games in non-gaming environments. Businesses, health care, and education use this technology. Consequently, teaching programming concepts with famous moral stories can be an excellent tool for illustrating how they're applied in real world problems. We believe children should be taught programming at an early age. It is easy to comprehend the basic concepts of programming for the kids. In light of this, we devised a mobile app that utilizes moral stories and simple quizzes to show kids how the basics of programming work. Different levels are presented based on the age of kids on the mobile application. A six to twelve-year-old is the target audience. In the beginning of the app, moral stories are shown on-screen and a problem is presented for the children to solve. Blocks of code are displayed, and they can be dragged and dropped onto the correct position on the screen. When the elements are arranged in the correct sequence, the animation is played. The kids are able to watch the code come to life. To test the users' understanding of the information, there is also a set of quizzes. The dashboard section lets parents follow their children's progress. Our goal is to encourage the learning of programming in young people, and we will succeed in this mission.

Keywords: Programming, Block-based, Gamification

Dedication

Our most sincere thanks go out to the nurturing affection, love, and committed partnership that our families have provided for us, and for giving us the opportunity to succeed in our lives.

Acknowledgments

It is Allah Almighty's blessing that we are strong and able to understand, learn, and solve problems. Our gratitude goes out to everyone who has supported us throughout our final year project.

We give a heartfelt thank you to our supervisor, Ms. Iqra Basharat, who made our research and writing much easier. Her helpful information, practical advice, and unfailing ideas have contributed significantly to our development of the project and writing the project report.

We would like to thank all the faculty members in the Software Engineering department for their support! It is our pleasure to express our sincere gratitude to Bahria University Islamabad for the opportunity they gave us to prove ourselves. Thanks to their help, we have been able to successfully complete our final project.

Table of Contents

Thesis Completion Certificate	ii
Certificate of Originality	iii
Abstract.....	iv
Dedication	v
Acknowledgments	vi
Table of Contents	vii
List of Figures.....	xi
List of Tables	xiii
Chapter 1	1
Introduction.....	1
1.1. Motivation	1
1.2. Problem statement	1
1.3. Objectives.....	2
1.4. Main contributions	2
1.5. Report organisation	3
Chapter 2	4
Background Study/Literature Review	4
2.1. Key Concepts	4
2.2. Analysis of existing work.....	4
2.2.1. Problems with existing applications:	4
2.2.2. Advantages of existing application over books:	5
2.3. Differences and similarities	5
2.4. Research Work	5
2.5. Conclusion.....	6
Chapter 3	7
System Requirements	7
3.1. Use Case Diagram.....	7
3.1.1 Launch as Parent	8
3.1.2 Create Child Profile	8
3.1.3 Update Child Profile	9
3.1.4 View Interface Guide.....	9
3.1.5 Change Settings	10
3.1.6 Select Child Profile	11
3.1.7 Launch Tutorial.....	11
3.1.8 Launch Project	12

3.1.9	Launch Quiz.....	12
3.1.10	Submit Quiz	13
3.1.11	Select Story	13
3.1.12	Start New Project	14
3.1.13	Use Template	14
3.1.14	Save Project	15
3.1.15	Launch Guest Mode.....	15
3.2.	Functional Requirements.....	16
3.2.1.	FR # 1: Create Child Profile:.....	16
3.2.2	FR # 2: Modify Child Profile:.....	16
3.2.3	FR # 3: Launch Guest Mode:.....	16
3.2.4	FR # 4: Launch Quiz:.....	16
3.2.5	FR # 5: Launch Project:	16
3.2.6	FR # 6: Change Settings:	17
3.2.7	FR # 7: Use Template:	17
3.3.	Interface Requirements.....	17
3.3.1.	User Interfaces	17
3.3.2.	Hardware Interfaces.....	17
3.3.3.	Software Interfaces	17
3.3.4.	Communications Interfaces	17
3.4.	Non-Functional Requirements	18
3.4.1.	Performance Requirements.....	18
3.4.2.	Safety Requirements.....	18
3.4.3.	Security Requirements.....	18
3.4.4.	Software Quality Attributes	18
3.5.	Project Feasibility.....	19
3.5.1.	Technical Feasibility.....	19
3.5.2.	Operational Feasibility	19
3.5.3.	Legal and Ethical Feasibility	19
3.6.	Analysis Models.....	20
3.6.1.	Activity Diagram	20
3.6.2.	Sequence Diagram	21
3.7.	Conclusion.....	28
Chapter 4	29
System Design.....	29
4.1.	Design Approach.....	29
4.2.	Design Constraints	29

4.4. Logical Design	30
4.4.1. Domain Model	30
4.4.2. Class Diagram.....	31
4.5. Dynamic View.....	33
4.5.1. State-transition Diagram.....	33
4.6. User Interface Design.....	33
4.7. Conclusion.....	43
Chapter 5	44
System Implementation.....	44
5.1. Tools Used.....	44
5.1.1. Unity	44
5.1.2. Blender.....	44
5.1.3. C#.....	44
5.1.4. Adobe Photoshop.....	45
5.2. Algorithms.....	45
5.3. Methodologies.....	45
5.3.1. Agile Development.....	45
5.4. Conclusion.....	46
Chapter 6	47
System Testing & Evaluation.....	47
6.1 Test Strategy	47
6.2 Component Testing.....	47
6.3 Unit Testing	47
6.4 Integrated Testing.....	48
6.5 Test Cases.....	48
6.6 Integration Testing.....	52
6.7 System Testing	52
6.7.1.1. GUI Test Cases:.....	52
6.8 Load Testing.....	57
6.9 Security Testing.....	57
6.10 Results & Evaluation	58
6.11 Conclusion	58
Chapter 7	59
Conclusion	59
7.1. Contributions.....	59
7.2. Reflections.....	59
7.3. Future work	60

References.....	61
Appendix A.....	62

List of Figures

Figure 1 - Thesis Organization	3
Figure 2 - Usecase Diagram of PlayCode	7
Figure 3 - Activity Diagram of PlayCode	20
Figure 4 - Sequence diagram for “Launch as Parent”	21
Figure 5 - Sequence diagram for “Create Child Profile”	21
Figure 6 - Sequence diagram for “Update Child Profile”	22
Figure 7 - Sequence diagram for “View Interface Guide”	22
Figure 8 - Sequence diagram for “Change Setting”	23
Figure 9 - Sequence diagram for “Select Child Profile”	23
Figure 10 - Sequence diagram for “Launch Tutorial”	24
Figure 11 - Sequence diagram for “Select Story”	24
Figure 12 - Sequence diagram for “Launch Project”	25
Figure 13 - Sequence diagram for “Launch Quiz”	25
Figure 14 - Sequence diagram for “Submit Quiz”	26
Figure 15 - Sequence diagram for “Save Project”	26
Figure 16 - Sequence diagram for “Launch Guest Mode”	27
Figure 17- Sequence diagram for “Start New Project”	27
Figure 18 - Sequence diagram for “Use Template”	28
Figure 19 - Domain Model of PlayCode.....	30
Figure 20 - Class Diagram of PlayCode	31
Figure 21 - State Transition Diagram for Android Application.....	33
Figure 22 - Main Menu Interface.....	34
Figure 23 - Edit Profile Interface	34
Figure 24 – Levels Interface	35
Figure 25 – Story Interface	35
Figure 26 – Blocks Interface.....	36
Figure 27 - Quiz Interface.....	36
Figure 28 – Home screen Interface.....	37
Figure 29 - Main Menu Interface.....	37
Figure 30 - Edit Profile Interface	38
Figure 31 - Current Profile Interface	38
Figure 32 – Progress Dashboard Interface.....	39
Figure 33 – Settings Interface	40
Figure 34 – Levels Interface	40
Figure 35 – Objectives Interface.....	41

Figure 36 - Story-Based Level Interface.....	41
Figure 37 – Output Interface.....	42
Figure 38 – Quiz Interface	42
Figure 39 – Results Interface	43

List of Tables

Table 1 - Differences between PlayCode and existing applications	5
Table 2 - Launch as Parent Usecase Description	8
Table 3 - Create Child Profile Usecase Description	9
Table 4 - Update Child Profile Usecase Description	9
Table 5 - View Interface Guide Usecase Description.....	10
Table 6 - Change Settings Usecase Description	10
Table 7 - Select Child Profile Usecase Description.....	11
Table 8 - Launch Tutorial Usecase Description.....	12
Table 9 - Launch Project Usecase Description	12
Table 10 - Launch Quiz Usecase Description.....	13
Table 11 - Submit Quiz Usecase Description	13
Table 12 - Select Story Usecase Description	14
Table 13 - Start New Project Usecase Description.....	14
Table 14 - Use Template Usecase Description	15
Table 15 - Save Project Usecase Description	15
Table 16 - Launch Guest Mode Usecase Description.....	16
Table 17 - Software Quality Attributes of PlayCode	18
Table 18 - Test Case 1 Specification	48
Table 19 - Test Case 2 Specification	49
Table 20 - Test Case 3 Specification	49
Table 21 - Test Case 4 Specification	50
Table 22 - Test Case 5 Specification	50
Table 23 - Test Case 6 Specification	51
Table 24 - Test Case 7 Specification	51
Table 25 -Test Case 8 Specification	51
Table 26 - Test Case 9 Specification	52
Table 27 - Test Case 10 Specification	53
Table 28 - Test Case 11 Specification	53
Table 29 -Test Case 12 Specification	54
Table 30 - Test Case 13 Specification	54
Table 31 - Test Case 14 Specification	54
Table 32 - Test Case 15 Specification	55
Table 33 - Test Case 16 Specification	55
Table 34 - Test Case 17 Specification	56
Table 35 - Test Case 18 Specification	56

Table 36 - Test Case 19 Specification	57
Table 37 - Test Case 20 Specification	57
Table 38 - Appendix A	62

Chapter 1

Introduction

1.1. Motivation

In recent years, children have become increasingly involved in coding activities as part of K-12 education. A wide variety of new technologies and programming environments have opened up new possibilities and increased the need for producing effective learning tools. It has become more and more popular to teach students programming through block-based languages. Many studies have indicated that block-based languages are an effective way of teaching computer programming concepts [1, 2, 3, 4]. With the world becoming more technologically advanced and new technologies being launched day by day, everything is now becoming more and more automated. Programming forms the basis of all new technologies. Therefore, educating our young generation about programming is necessary. Our aim and motivation behind this project are to teach programming basics to kids of ages 6 to 12 because if kids get introduced to programming at such young age, it will become easy to understand advanced concepts at later ages. Android devices are very common these days. Nowadays, it is possible to teach kids programming through an Android app. Since they are used to using a smartphone device, it is quite possible to teach them. Despite the fact that programming is an essential skill, it doesn't seem that popular right now in our country. Our goal is to leverage motivational stories and simple quizzes to explain programming to kids through the use of an android app. The primary motivation is to contribute to the society as software engineers and make a difference.

1.2. Problem statement

Even though programming is a valuable skill, it seems not very widespread in our country right now. We believe that kids should begin learning programming at a very young age. Children can learn programming through many websites and apps available already, but these are not easy to use for kids from 6 years old and up. To help kids understand programming, we used easy-to-understand blocks, stories and scenarios.

1.3. Objectives

Some of the primary objectives of this project include:

1. To teach basics of programming such as: variables, loops and conditionals.
2. To test the concepts of users using quizzes covering all the basic knowledge.

1.4. Main contributions

- **What is new, different, better, significant?**

Over the years, many books have been written that aim to teach programming to kids. Moreover, a variety of mobile apps have been developed that aim to do so. These days, books are not very common. However, kids are very used to smartphones. The smartphone apps already available for teaching programming use a tedious approach. Most of them rely on boring puzzles. However, we decided to do something different. We utilized character-based stories and interactive blocks of code.

- **Why is the world a better place because of what I have done?**

Since there has been no product for this purpose launched in Pakistan until now, this app is a whole new concept. Education in this country fails to implement interactive teaching methods and lacks current knowledge, so this project fills a major hole in that sector. The app will also make profits, and its potential user base would grow. Parents will be able to keep an eye on their child's progress. They will be able to measure the learning of the kids by letting them take quizzes included within the app.

- **What is now known/possible/better because of this project?**

PlayCode is an interactive solution for teachers and parents who can save immensely on time and resources to help the children learn programming. Through this project, kids will be able to test their abilities, use blocks of code to learn coding, and have fundamental programming challenges presented to them. This application was developed for Android operating system, which is an open source platform that is distributed, and maintained by Google. PlayCode is potentially compatible with 2.5 billion active Android devices since Android is the world's most widely used operating system.

1.5. Report organisation

As shown in figure 1 below, the report has a clear structure.

Chapter 2 is a review of the literature and background study. Based on the existing work, we have also analysed the problem statement and the proposed solution. In this chapter, we also discuss project overview, including the objectives, vision, and scope of the project.

In **Chapter 3**, System requirements are described, including functional and non-functional requirements as well. A model analysis and project feasibility are also discussed.

In **Chapter 4**, we discuss the design of the system, the design process, constraints, logical design, and dynamic view. Additionally, prototypes of user interfaces are included.

In **Chapter 5**, we discuss how the project can be applied based on its implementation strategy, the technologies used, and related mythologies.

As part of **Chapter 6**, we discuss how the project was evaluated, including all sorts of testing such as unit testing, integration testing, and system testing.

The thesis concludes in **Chapter 7** with a summary of various aspects of the work, as well as future work to be undertaken.

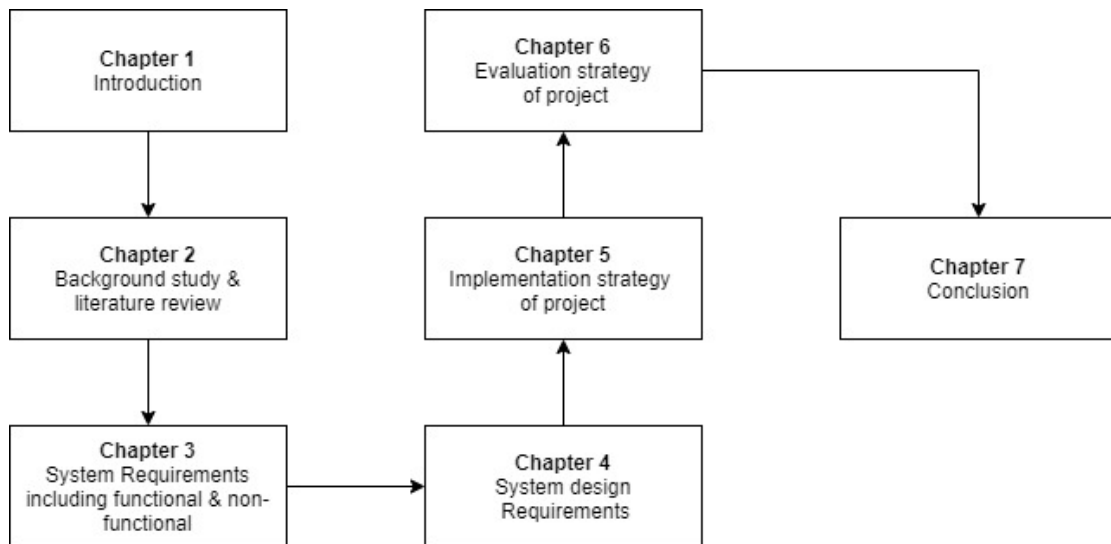


Figure 1 - Thesis Organization

Chapter 2

Background Study/Literature Review

2.1. Key Concepts

The background review will give us a better understanding of work related to our project. This will highlight existing applications that serve the same purpose. Furthermore, it will allow us to understand why applications using these existing technologies need modification for better results.

2.2. Analysis of existing work

Teachers are using old books which is not a good approach since kids these days are fond of smartphones. Kids won't be able to learn with full interest using books. Additionally, books can only be used to understand a concept, but they cannot be used to test the concept in real-time by practically doing it. It's not possible to update the books with latest knowledge. Therefore, it is not cost effective to buy revised versions of a book with updated knowledge on a regular basis.

2.2.1. Problems with existing applications:

There are a variety of applications already available on the play store.

Comparatively, these applications have these weaknesses:

- The major problem with most of them is that they demand the need of a parent that will assist the child in navigating across the application.
- Moreover, a lot of apps demand an active internet connection to store the app data on the cloud and keep the profiles updated.
- The apps already available are very costly. The costs are either on monthly subscription or cost per feature. However, PlayCode is currently 100% free, making it very cost-effective.
- Most of the apps require large screen devices and preferably tablets. However, PlayCode can run on any Android device either of large or small screen.

2.2.2. Advantages of existing application over books:

Despite the problems, all of the existing applications are better than the books on the following counts:

- It is easy and quick to update an android application with latest information at any time. The user can automatically get the app updated with latest information.
- Among the most popular operating systems on the planet, Android has gained worldwide recognition. As a consequence, Android apps have the potential to work on 2.5 billion active Android devices around the world.
- A book cannot test students' knowledge and ability. There are, however, applications that allow the user to take quizzes and see the results right away.

2.3. Differences and similarities

To overcome the problems encountered in already available apps and methods for teaching basic programming concepts to kids, we have created an app named as “PlayCode”.

Our Project's Characteristics	Other apps
No Internet connectivity required	Required
Cost effective	No
No need of a helper to help in usage	Helper Needed
Guest Mode	No
Usable on all screen sizes	No

Table 1 - Differences between PlayCode and existing applications

2.4. Research Work

In order to understand what past work has been done in this area, we have examined different research articles. It has been interesting to learn about the technologies that they have or are considering using. Through this research, we were able to identify the appropriate technologies for our project.

The idea that K-12 students should be taught computer science and computational thinking is gaining momentum [5]. There have been several organizations that have developed standards applied to coding education [5]. These organizations are Computer

Science Teachers Association (CSTA), Informatics Europe and the Cyber Innovation Center.

Additionally, Code.org and Codecademy have played a large role in fostering interest in coding among young people in recent years. These organizations believe that coding is a valuable skill that will open up a wide range of professional opportunities to people.

Across many countries, coding is taught in schools as part of the curriculum. For instance, the United Kingdom requires students to take computer programming courses in primary school on a regular basis [6]. The Danish government promotes computer literacy by focusing on creating technologies that advance knowledge [7].

2.5. Conclusion

Through the course of this chapter, we have reviewed the background and literature of this thesis. We reviewed previous applications similar to ours when discussing this. The differences between our app and those of other apps were also discussed. Research papers we have read in order to gain a deeper understanding of the technological capabilities available to our project are also included.

Chapter 3

System Requirements

Throughout this chapter, we'll talk about our project's requirements. We will cover both the functional and non-functional aspects of this project. Aside from hardware and software requirements, we will also discuss interface requirements. Analysing the project's legal, operational, and technical feasibility, we will see whether or not it can be realized. With our analysis model and activity diagram, we will be able to visualize the flow of our project.

3.1. Use Case Diagram

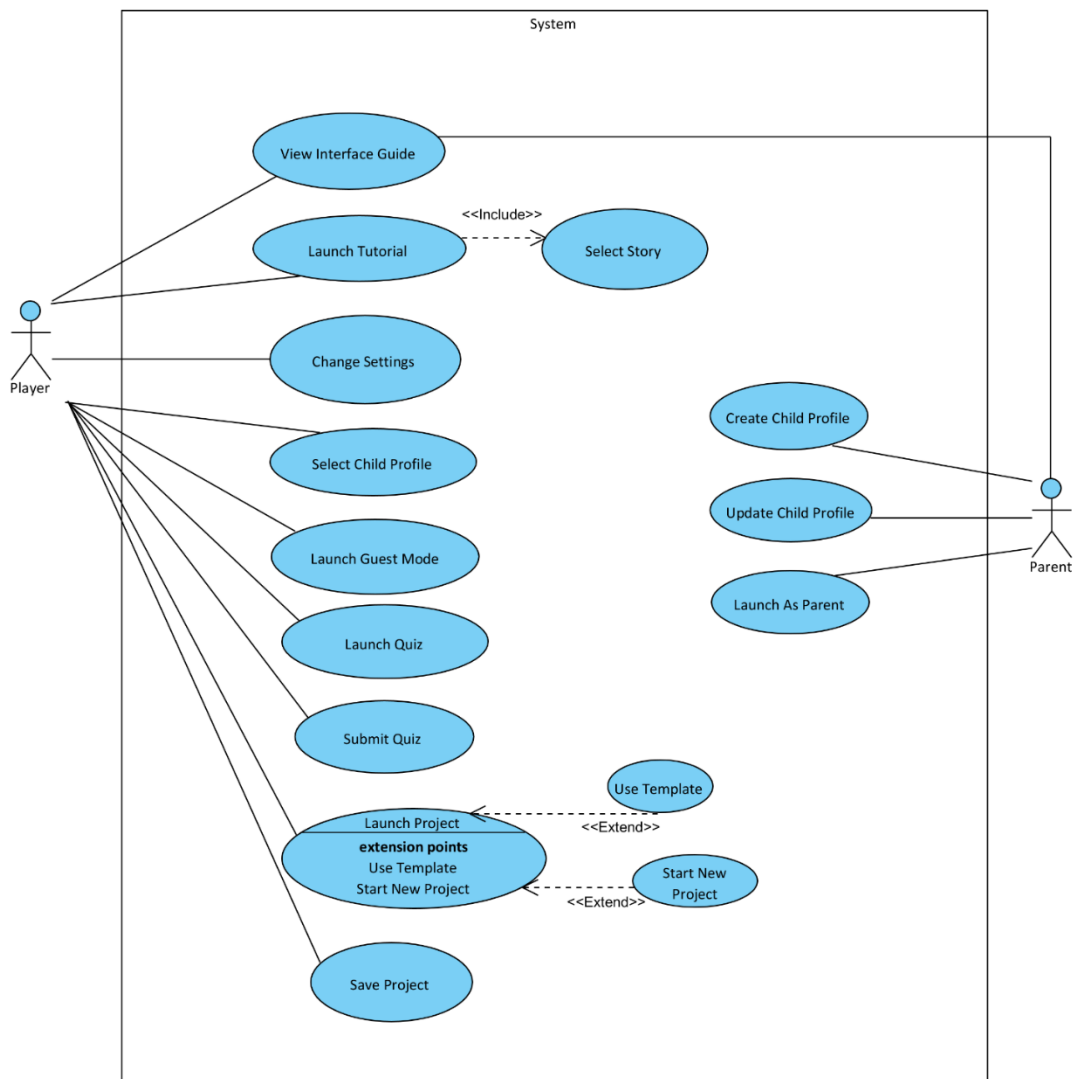


Figure 2 - Usecase Diagram of PlayCode

3.1.1 Launch as Parent

Use Case ID:	UC1	
Use Case Name:	Launch as Parent	
Actor(s):	Parent	
Pre-Conditions:	The application should have launched properly	
Priority:	High	
Basic Flow:	The user enters the dashboard for parents.	
Actor Actions		System Response
	Parent opens the app for the first time.	2. System shows two the main menu
	Parent taps the dashboard option.	4 The application takes them to the “Parent’s Dashboard”.
Alternative Course of Action (if any)		
Actor Action		System Response
	Parent taps the quit button.	2a The app is closed.

Table 2 - Launch as Parent Usecase Description

3.1.2 Create Child Profile

Use Case ID:	UC2	
Use Case Name:	Create Child Profile	
Actor(s):	Parent	
Pre-Conditions:	The parent should have entered “Parent’s Dashboard” successfully.	
Priority:	High	
Basic Flow:	The user enters the dashboard for parents and creates a child profile.	
Actor Actions		System Response
1	Parent enters the “Parent’s Dashboard”.	2. System shows options including: <ul style="list-style-type: none"> • Create Child Profile • Update Child Profile
3	Parent taps the “Create Child Profile” option.	4 The application asks the parent to enter the following details: <ul style="list-style-type: none"> • Child Name • Child Age
5	Parent enters the details and clicks “Create.”	6 System creates the child profile.
Alternative Course of Action (if any)		
Actor Action		System Response

3a	On the “Parent’s Dashboard”, Parent taps the “Update Child Profile” option.	4a System asks to create a child profile first.
5a	Parent enters only child name and clicks “Create”.	6a System shows error message “Enter all the details”.

Table 3 - Create Child Profile Usecase Description

3.1.3 Update Child Profile

Use Case ID:	UC3	
Use Case Name:	Update Child Profile	
Actor(s):	Parent	
Pre-Conditions:	A Child Profile should already have been created.	
Priority:	High	
Basic Flow:	The user enters the dashboard for parents and updates a child profile.	
Actor Actions		System Response
	Parent enters the “Parent’s Dashboard”.	2. System shows options including: <ul style="list-style-type: none"> • Create Child Profile • Update Child Profile
	Parent taps the “Update Child Profile” option.	4 The application asks the parent to modify the following details: <ul style="list-style-type: none"> • Child Name • Child Age
	Parent modifies the details and clicks “Create.”	6 System updates the child profile.
Alternative Course of Action (if any)		
Actor Action		System Response
	On the “Parent’s Dashboard”, Parent taps the “Create Child Profile” option.	4a System asks to create another child profile.
	Parent taps the back button.	6a System shows the main menu.

Table 4 - Update Child Profile Usecase Description

3.1.4 View Interface Guide

Use Case ID:	UC4
Use Case Name:	View Interface Guide
Actor(s):	Player
Pre-Conditions:	The application should have launched properly
Priority:	High

Basic Flow:	The player launches the application and views the interface guide.	
Actor Actions		System Response
	Player launches the application.	2. System shows the main menu.
	Player taps the “View Interface Guide” option.	4 The application the Interface guide.
Alternative Course of Action (if any)		
Actor Action		System Response
	On the main menu, player taps the “Exit” option.	4a The application is closed.
	Player taps the back button.	6a System shows the main menu.

Table 5 - View Interface Guide Usecase Description

3.1.5 Change Settings

Use Case ID:	UC5	
Use Case Name:	Change Settings	
Actor(s):	Player	
Pre-Conditions:	The application should have launched properly	
Priority:	High	
Basic Flow:	The player launches the application and changes the settings.	
Actor Actions		System Response
	Player launches the application.	2. System shows the main menu.
	Player taps the “Change Settings” option.	4 The application shows the following options: <ul style="list-style-type: none"> • Toggle Vibration • Toggle Sound
	Player changes the settings and clicks “Close”.	6 The changes are saved and main menu is shown.
Alternative Course of Action (if any)		
Actor Action		System Response
	On the main menu, player taps the “Exit” option.	4a The application is closed.
	Player taps the back button.	6a System shows the main menu.

Table 6 - Change Settings Usecase Description

3.1.6 Select Child Profile

Use Case ID:	UC6	
Use Case Name:	Select Child Profile	
Actor(s):	Player	
Pre-Conditions:	A Child Profile should have been created already.	
Priority:	High	
Basic Flow:	The player launches the application and selects a child profile.	
Actor Actions		System Response
	Player launches the application.	2. System shows the main menu.
	Player taps the “Select Child Profile” option.	4 The application shows the available Child profile.
	Player taps on a Child Profile to open.	6 The selected Child Profile is opened.
Alternative Course of Action (if any)		
Actor Action		System Response
	On the main menu, player taps the “Exit” option.	4a The application is closed.
	Player taps the back button.	6a System shows the main menu.

Table 7 - Select Child Profile Usecase Description

3.1.7 Launch Tutorial

Use Case ID:	UC7	
Use Case Name:	Launch Tutorial	
Actor(s):	Player	
Pre-Conditions:	A Child Profile should have been selected already.	
Priority:	High	
Basic Flow:	The player launches a tutorial from the available tutorials.	
Actor Actions		System Response
	Player selects a child profile.	2. System shows the following options: <ul style="list-style-type: none"> • Launch Tutorial • Launch Project • Launch Quiz
	Player selects the “Launch Tutorial” option.	4 The application shows the available tutorials.
	Player selects a tutorial to open.	6 The selected tutorial is opened.
Alternative Course of Action (if any)		
Actor Action		System Response
	Player selects the “Launch Project” option.	4a The application launches the project mode.

	Player selects the “Launch Quiz” option.	4b The application launches the quiz mode.
	Player taps the back button.	6a System shows the main menu.

Table 8 - Launch Tutorial Usecase Description

3.1.8 Launch Project

Use Case ID:	UC8	
Use Case Name:	Launch Project	
Actor(s):	Player	
Pre-Conditions:	A Child Profile should have been selected already.	
Priority:	High	
Basic Flow:	The player launches the project mode.	
Actor Actions		System Response
	Player selects a child profile.	2. System shows the following options: <ul style="list-style-type: none"> • Launch Tutorial • Launch Project • Launch Quiz
	Player selects the “Launch Project” option.	4 The application launches the project mode.
Alternative Course of Action (if any)		
Actor Action		System Response
	Player selects the “Launch Tutorial” option.	4a The application launches the tutorial mode.
	Player selects the “Launch Quiz” option.	4b The application launches the quiz mode.
	Player taps the back button.	6a System shows the main menu.

Table 9 - Launch Project Usecase Description

3.1.9 Launch Quiz

Use Case ID:	UC9	
Use Case Name:	Launch Quiz	
Actor(s):	Player	
Pre-Conditions:	A Child Profile should have been selected already.	
Priority:	High	
Basic Flow:	The player launches the quiz mode.	
Actor Actions		System Response
	Player selects a child profile.	2. System shows the following options: <ul style="list-style-type: none"> • Launch Tutorial

		<ul style="list-style-type: none"> • Launch Project • Launch Quiz
	Player selects the “Launch Quiz” option.	4 The application launches the quiz mode.
Alternative Course of Action (if any)		
Actor Action		System Response
	Player selects the “Launch Tutorial” option.	4a The application launches the tutorial mode.
	Player selects the “Launch Project” option.	4b The application launches the project mode.
	Player taps the back button.	6a System shows the main menu.

Table 10 - Launch Quiz Usecase Description

3.1.10 Submit Quiz

Use Case ID:	UC10	
Use Case Name:	Submit Quiz	
Actor(s):	Player	
Pre-Conditions:	The player should have completed a quiz.	
Priority:	High	
Basic Flow:	The player finishes the quiz and submits it.	
Actor Actions		System Response
	Player enters the quiz mode.	2. System displays the quiz questions.
	Player selects the appropriate answers and clicks “Submit”.	4 The application displays the result.
Alternative Course of Action (if any)		
Actor Action		System Response
	Player clicks “Submit” button without attempting all the questions.	4a The application warns to complete all the questions or marks will be deducted.
	Player taps the back button.	6a The application shows a dialog confirming to close quiz without completing.

Table 11 - Submit Quiz Usecase Description

3.1.11 Select Story

Use Case ID:	UC11
Use Case Name:	Select Story
Actor(s):	Player
Pre-Conditions:	The player should have entered the tutorial mode.
Priority:	High

Basic Flow:	The player launches a story from the available options.	
Actor Actions		System Response
	Player enters the tutorial mode.	2. System shows available stories.
	Player selects a story.	4 The application launches the selected story.
Alternative Course of Action (if any)		
Actor Action		System Response
	Player taps the back button.	4a System shows the main menu.

Table 12 - Select Story Usecase Description

3.1.12 Start New Project

Use Case ID:	UC12	
Use Case Name:	Start New Project	
Actor(s):	Player	
Pre-Conditions:	The player should have entered the project mode.	
Priority:	High	
Basic Flow:	The player launches a new project.	
Actor Actions		System Response
	Player enters the project mode.	2. System shows the following options: <ul style="list-style-type: none"> • Start New Project • Use template
	Player selects “Start New Project” option.	4 The application launches a new project.
Alternative Course of Action (if any)		
Actor Action		System Response
	Player selects “Use template” option.	4a The application shows available templates.
	Player taps the back button.	4b System shows the main menu.

Table 13 - Start New Project Usecase Description

3.1.13 Use Template

Use Case ID:	UC13
Use Case Name:	Use Template
Actor(s):	Player
Pre-Conditions:	The player should have entered the project mode.
Priority:	High

Basic Flow:	The player launches a project from given templates.	
Actor Actions		System Response
	Player enters the project mode.	2. System shows the following options: <ul style="list-style-type: none"> • Start New Project • Use template
	Player selects “Use template” option.	4 The application shows available templates.
	Players selects a template.	5 The selected template is launched.
Alternative Course of Action (if any)		
Actor Action		System Response
	Player selects “Start New Project” option.	4a The application launches a new project.
	Player taps the back button.	4b System shows the main menu.

Table 14 - Use Template Usecase Description

3.1.14 Save Project

Use Case ID:	UC14	
Use Case Name:	Save Project	
Actor(s):	Player	
Pre-Conditions:	The player should have launched a new project.	
Priority:	High	
Basic Flow:	The player edits a project and saves it.	
Actor Actions		System Response
	Player launches a new project.	2. System shows the project screen.
	Player edits the project and selects “Save Project” option.	4 The application saves the project.
Alternative Course of Action (if any)		
Actor Action		System Response
	Player taps the back button.	4a System shows a dialog confirming to exit without saving the project.

Table 15 - Save Project Usecase Description

3.1.15 Launch Guest Mode

Use Case ID:	UC15	
Use Case Name:	Launch Guest Mode	
Actor(s):	Player	
Pre-Conditions:	The application should have launched properly	
Priority:	High	

Basic Flow:	The user enters the guest mode.	
Actor Actions		System Response
	Parent opens the app for the first time.	2. System shows the main menu
	Parent selects the “Guest Mode” option.	4 The application launches the Guest Mode.
Alternative Course of Action (if any)		
Actor Action		System Response
	Parent taps the quit button.	2a The app is closed.

Table 16 - Launch Guest Mode Usecase Description

3.2. Functional Requirements

3.2.1. FR # 1: Create Child Profile:

The first module of this project. The user starts the inputs name and age and a profile is created. The application gives the user guidelines so that the user can start learning.

3.2.2 FR # 2: Modify Child Profile:

At any time, the user/child profile can be updated such as the name of the child and age.

3.2.3 FR # 3: Launch Guest Mode:

In Guest Mode, the user can use the different modes of the app without creating a profile.

3.2.4 FR # 4: Launch Quiz:

Quiz mode can be used to test users. In the quiz mode, all the concepts of programming taught in other modes of the app are assessed.

3.2.5 FR # 5: Launch Project:

In this module, the user is presented with different levels according to the age specified in the child profile. The user can select a level and start learning. The user must finish all the levels to fully learn the concepts.

3.2.6 FR # 6: Change Settings:

The settings allow the user to turn the sound and music on and off.

3.2.7 FR # 7: Use Template:

The user can use different templates to modify ready-made programs and learn how the change in some values affect the whole program

3.3. Interface Requirements

3.3.1. User Interfaces

PlayCode game has a Graphical User Interface with simple and self-explanatory icons and buttons and has the following screens:

- Launch Project
- User template
- Launch Quiz
- Change Settings
- Guest Mode
- Manage Child Profile

3.3.2. Hardware Interfaces

- Android Phone

3.3.3. Software Interfaces

- **Operating System:** Android
- **IDE:** Android Studio
- **Technology:** The project is programmed in C#.

3.3.4. Communications Interfaces

There are no communication interfaces since our mobile application works offline.

3.4. Non-Functional Requirements

3.4.1. Performance Requirements

The game shall not incur any glitch on a device with at least of 2GB RAM. The application runs at 30 frames per second.

3.4.2. Safety Requirements

“PlayCode” shall not cause any overheating issues in the player’s device. It shall not cause any damage to the hardware of the device or to other applications installed in the Player’s device.

3.4.3. Security Requirements

“PlayCode” shall not store any personal information of the player other than their name and age. Moreover, this information shall be stored locally on the device of the player. There shall be no sharing of this information with any server.

3.4.4. Software Quality Attributes

The simplicity, understandability and well-designed user-friendly interface of PlayCode makes it accessible to all types of users.

Reliability	"PlayCode" responds to the player's commands in 1 second. When the player drags the blocks of code, the output is generated within 1 second.
Availability	The mobile application is available 24/7.
Maintainability	The app is easily be maintainable by any technical person having knowledge of mobile application development.
User-friendliness	The app has a consistent UI (in all the views and dialogs, the UI elements behave and are placed in a similar way.)

Table 17 - Software Quality Attributes of PlayCode

3.5. Project Feasibility

3.5.1. Technical Feasibility

The project involves the development of an Android application. A device that runs Android is all that is needed to run an Android application. These requirements can all be managed easily.

3.5.2. Operational Feasibility

The project PlayCode is operationally feasible as it can be installed on thousands of android devices. Android devices such as tablets can be installed in schools with PlayCode installed. Therefore, kids can easily learn anywhere.

3.5.3. Legal and Ethical Feasibility

No legal or ethical barriers stand in our way when developing the application. In this case, it is legal and ethically permissible, since there are no legal or ethical concerns in its development.

3.6. Analysis Models

3.6.1. Activity Diagram

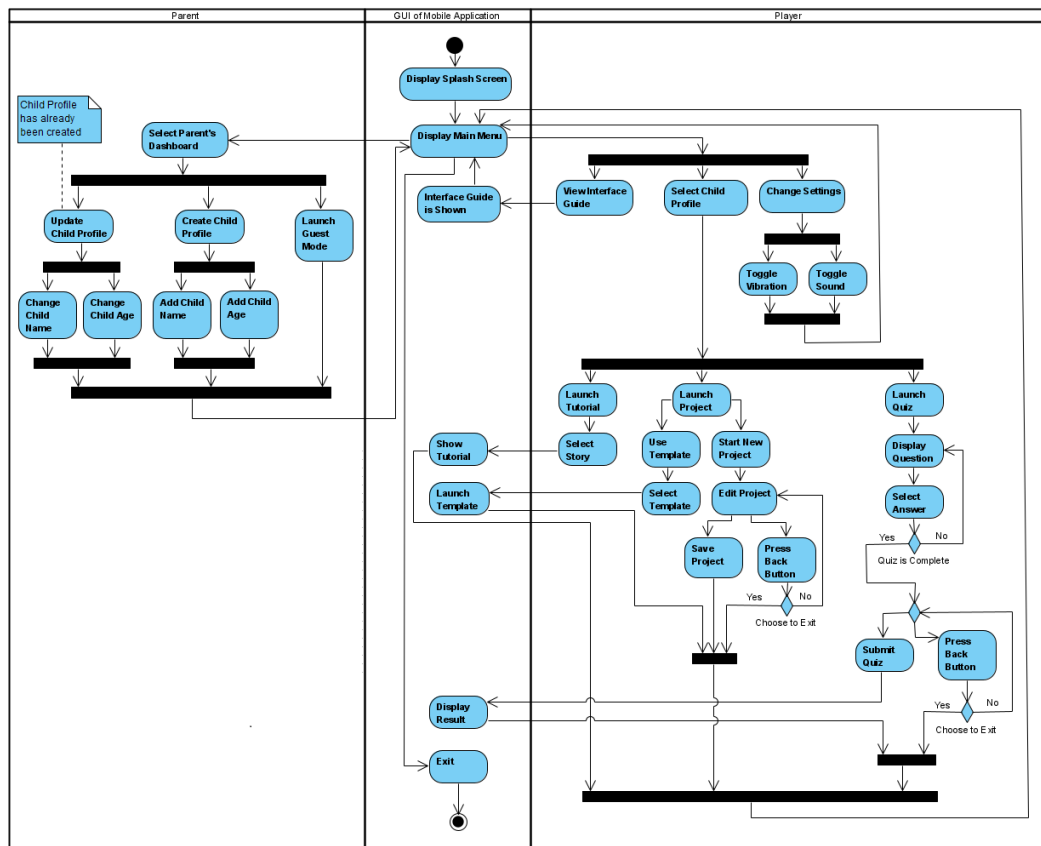


Figure 3 - Activity Diagram of PlayCode

1. **Parent** includes all the activities that will be performed by the parent. When parent opens the mobile application, the following activities can be performed:
 - Create Child Profile
 - Update Child Profile
 - Launch Guest Mode
2. **GUI of Mobile Application** includes following activities:
 - Display Splash Screen
 - Display Main Menu
 - Exit
3. **Player** includes the following activities:
 - View Interface Guide
 - Select Child Profile
 - Change Settings
 - Launch Tutorial
 - Launch Project

- Launch Quiz

3.6.2. Sequence Diagram

- Launch as Parent

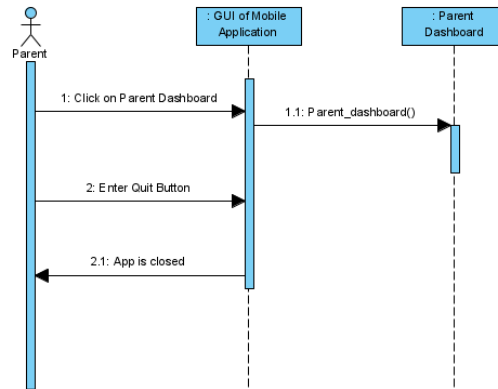


Figure 4 - Sequence diagram for “Launch as Parent”

- Create Child Profile

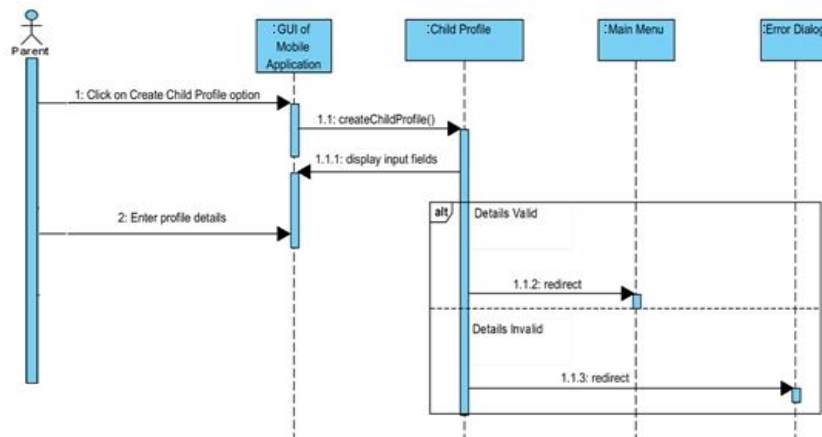


Figure 5 - Sequence diagram for “Create Child Profile”

- **Update Child Profile**

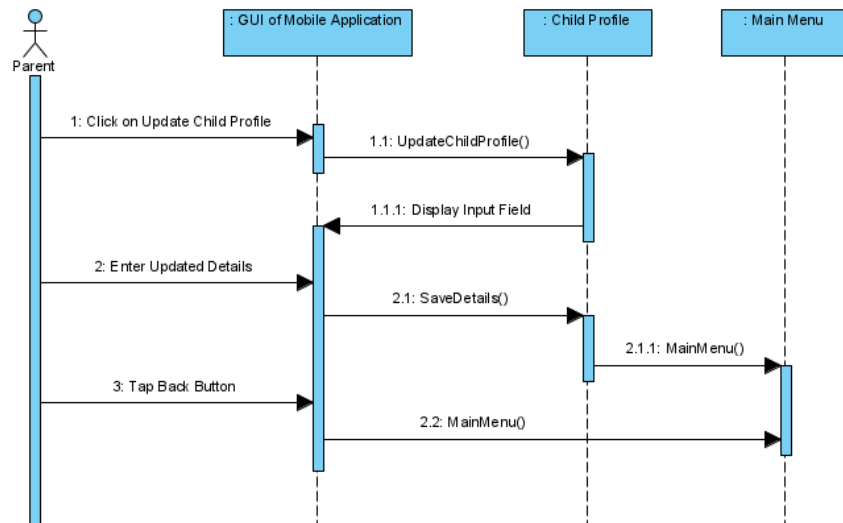


Figure 6 - Sequence diagram for “Update Child Profile”

- **View Interface Guide**

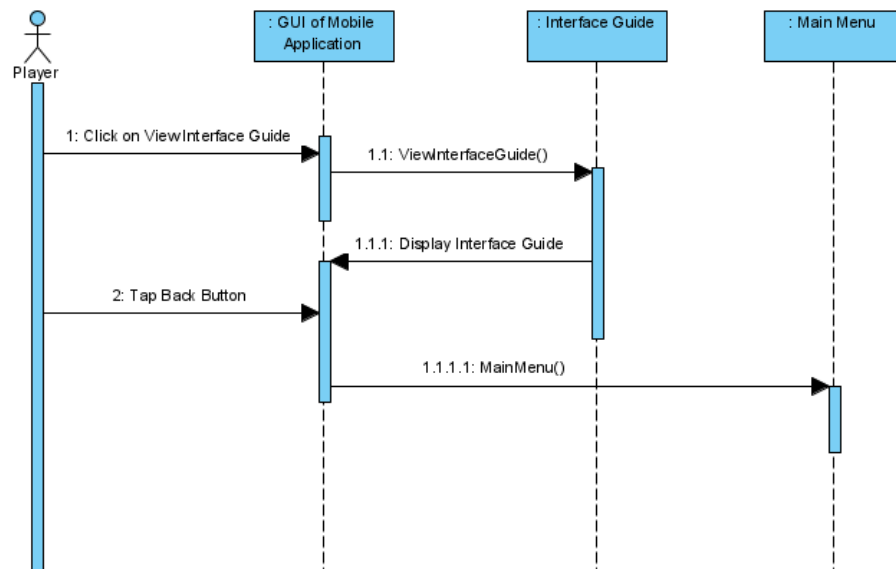


Figure 7 - Sequence diagram for “View Interface Guide”

- **Change Setting**

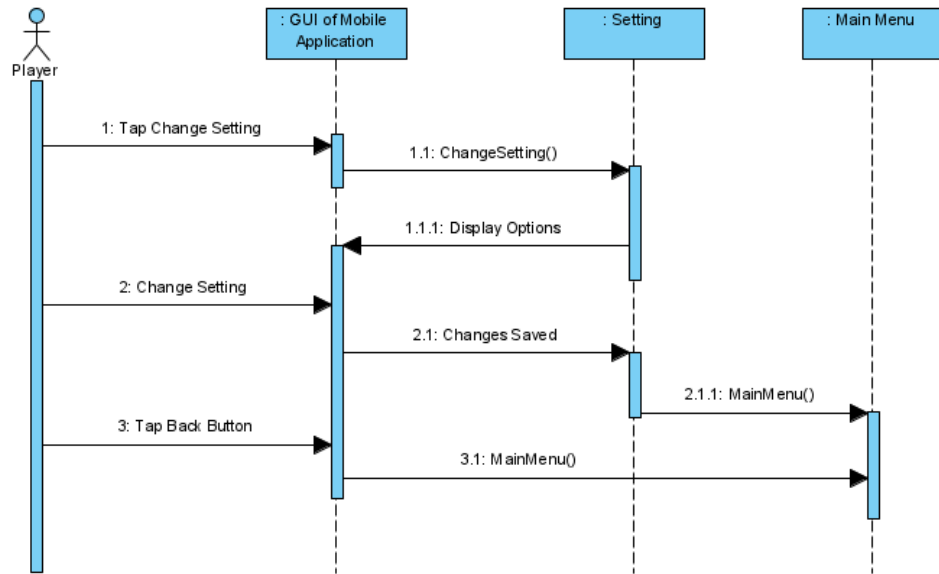


Figure 8 - Sequence diagram for “Change Setting”

- **Select Child Profile**

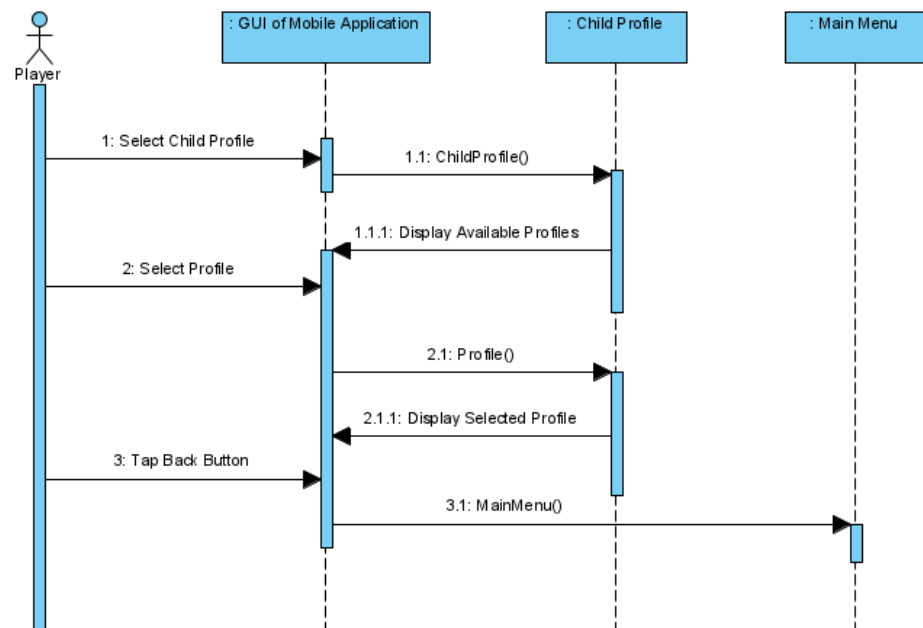


Figure 9 - Sequence diagram for “Select Child Profile”

- **Launch Tutorial**

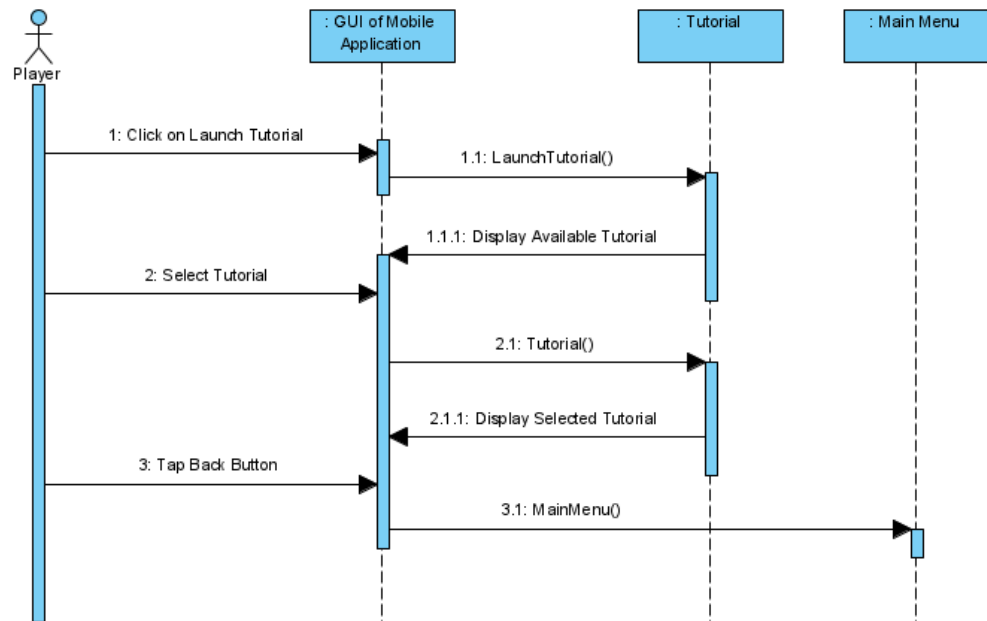


Figure 10 - Sequence diagram for “Launch Tutorial”

- **Select Story**

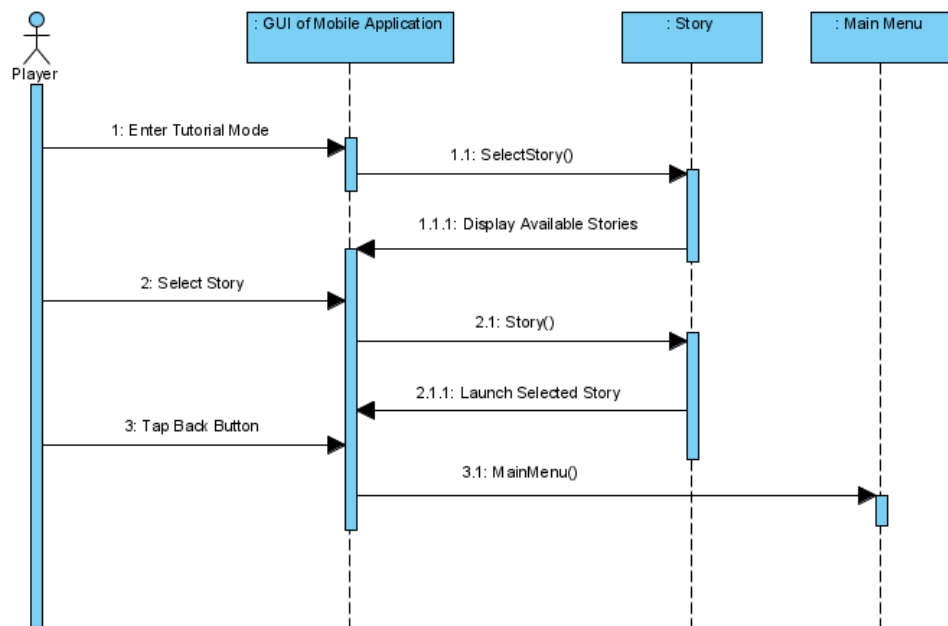


Figure 11 - Sequence diagram for “Select Story”

- **Launch Project**

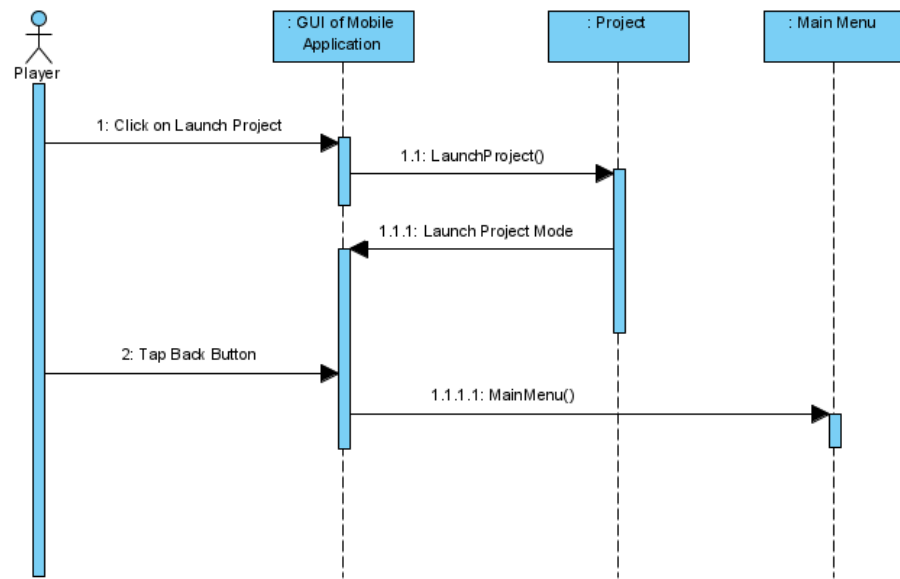


Figure 12 - Sequence diagram for “Launch Project”

- **Launch Quiz**

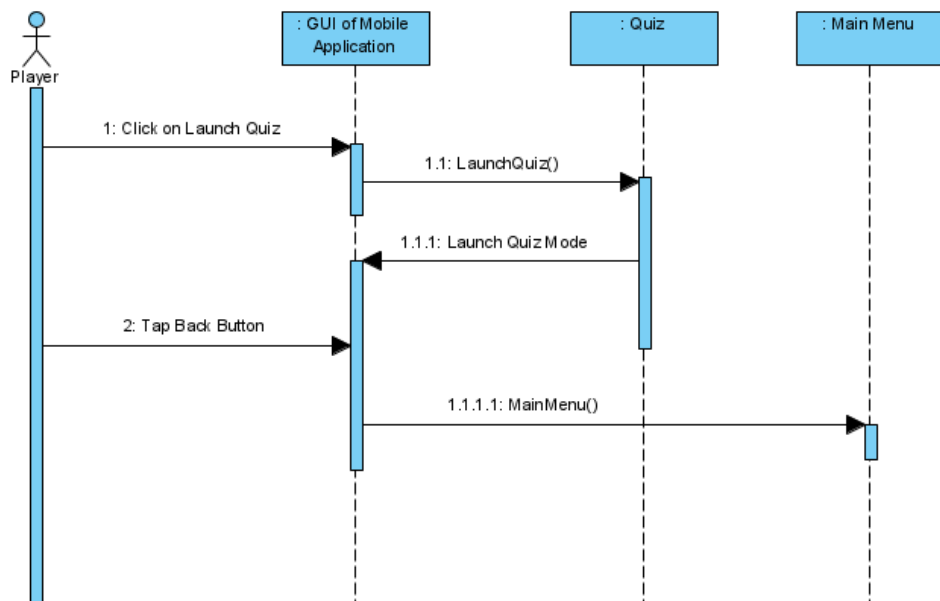


Figure 13 - Sequence diagram for “Launch Quiz”

- **Submit Quiz**

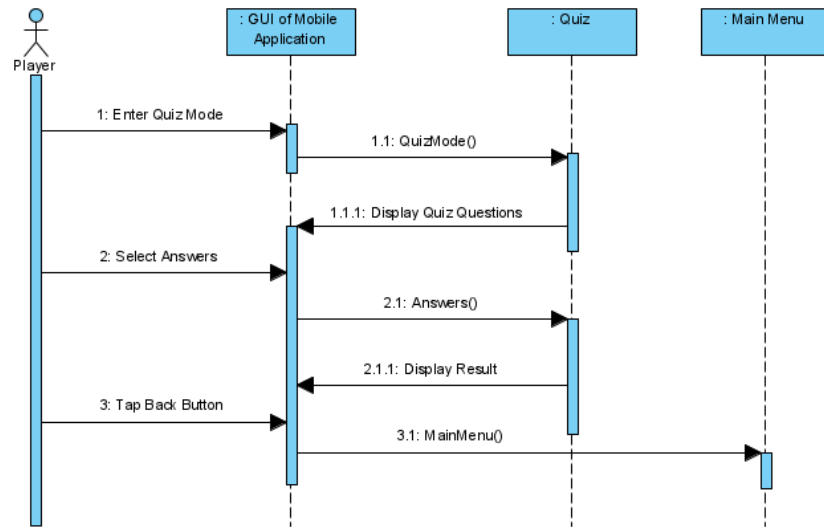


Figure 14 - Sequence diagram for “Submit Quiz”

- **Save Project**

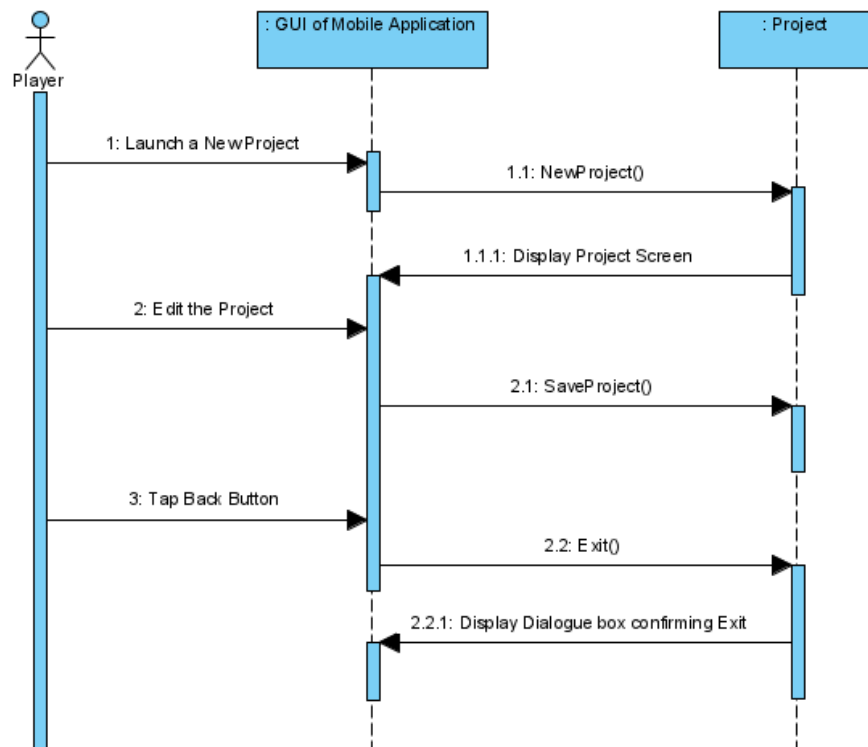


Figure 15 - Sequence diagram for “Save Project”

- **Launch Guest Mode**

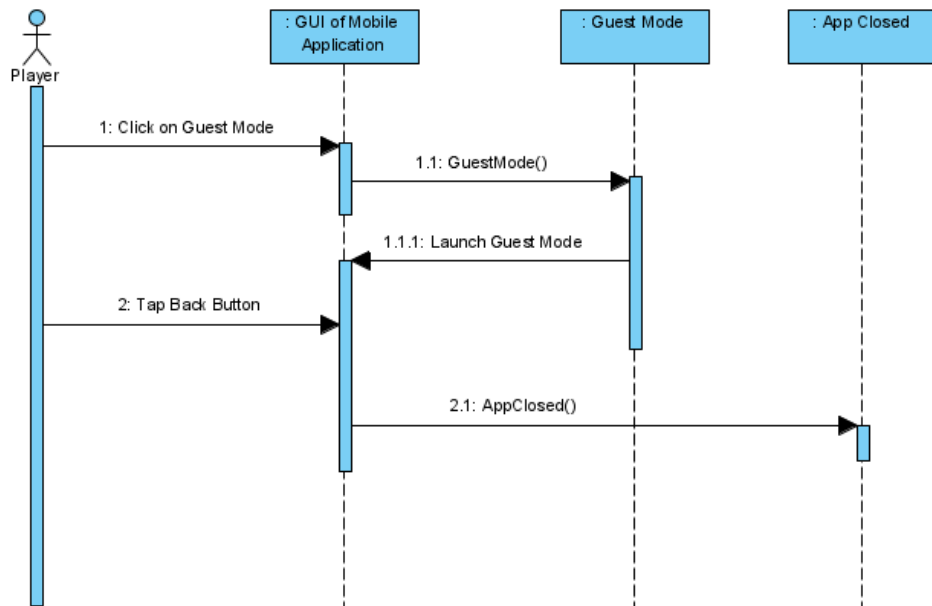


Figure 16 - Sequence diagram for “Launch Guest Mode”

- **Start New Project**

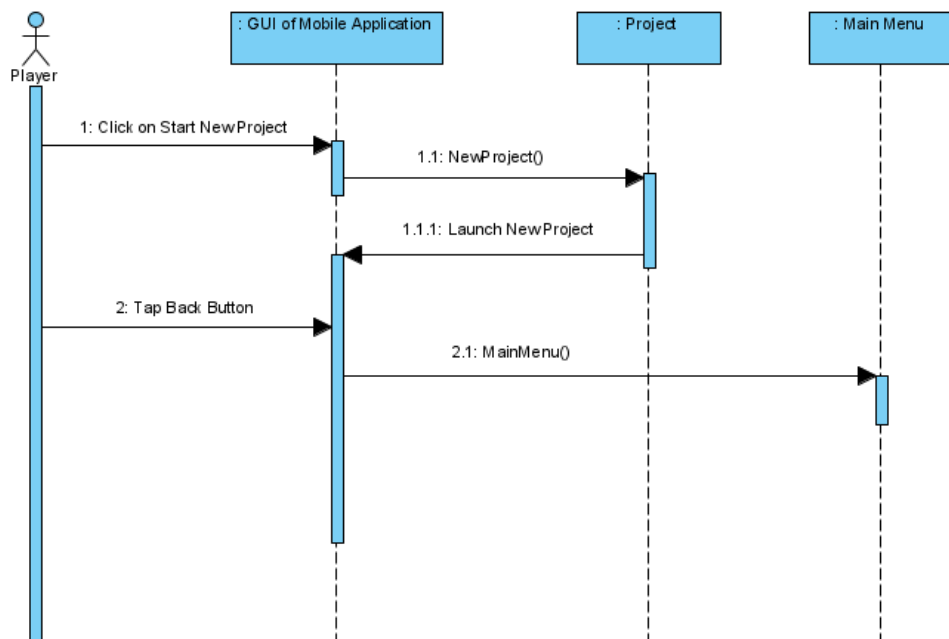


Figure 17- Sequence diagram for “Start New Project”

- **Use Template**

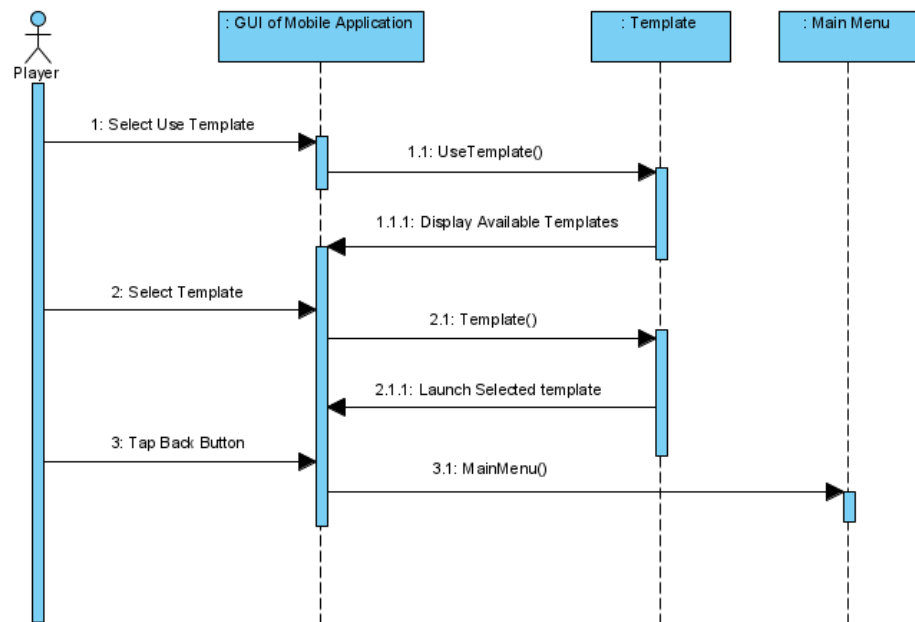


Figure 18 - Sequence diagram for “Use Template”

3.7. Conclusion

The requirements of our project have been discussed in this chapter. This project has been covered from both a functional and non-functional perspective. In addition to the hardware and software requirements, we discussed the interface requirements. By analysing the legal, operational, and technical feasibility, we have explored the project's feasibility. An analysis model for our project has been included, including an activity diagram and a sequence diagram.

Chapter 4

System Design

4.1. Design Approach

In the development of our android app, we utilized a functional approach to make sure that it is easy to understand and that the interface is intuitive. Our application is designed to provide a variety of features under one platform. Modules provide different functionality, each one providing a specific function. We consider each feature for one specific area as a submodule to minimize complexity. One module consists of all sub modules designed and implemented together. A fully functional application is made by integrating all the completed modules.

4.2. Design Constraints

We have listed the implementation limitations and constraints of this system below.

- 3D elements must be done using Unity 3D.
- A key constraint for implementation is that models must be thoroughly tested before being integrated into the game.
- The interaction constraint is that no more than 0.1 seconds should elapse between the user providing inputs and the inputs reflecting in the environment.
- Among the display constraints is a minimum frame rate of 20 frames per second for the visual images.

4.4. Logical Design

4.4.1. Domain Model

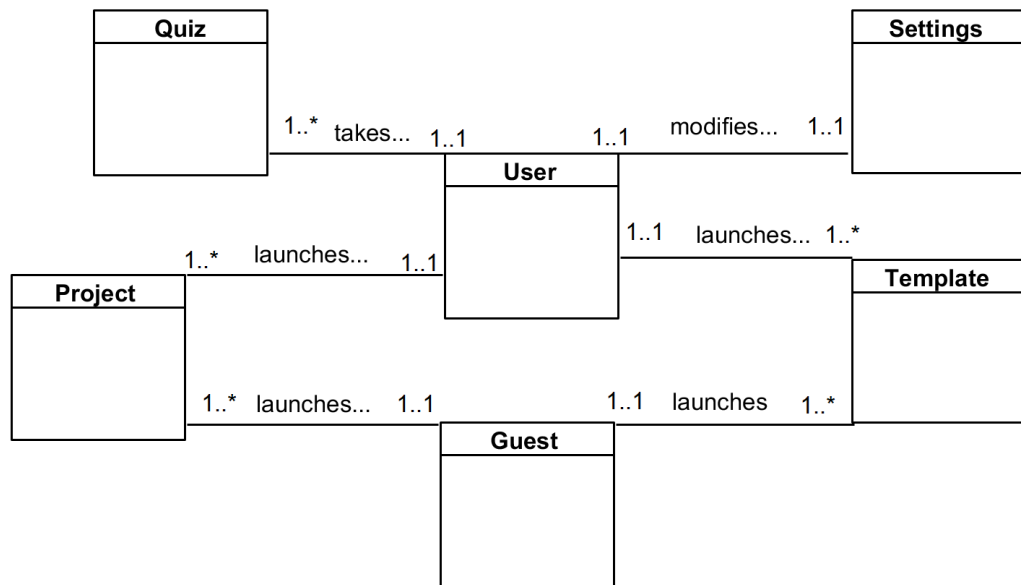


Figure 19 - Domain Model of PlayCode

The domain model shown in figure 20 includes:

- **User class**
- **Guest class**
- **Project class**
- **Template class**
- **Quiz class**
- **Settings class**

4.4.2. Class Diagram

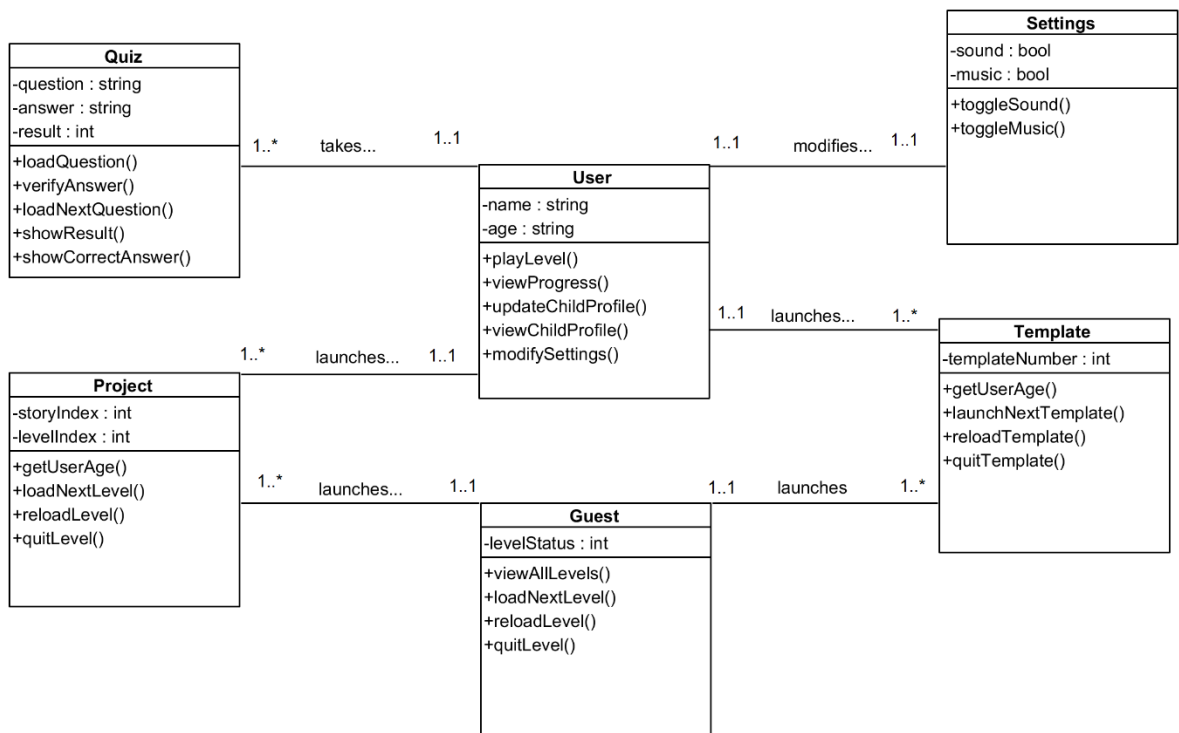


Figure 20 - Class Diagram of PlayCode

- **User class** includes
 - **Attribute:** name, age of data type string
 - **Functions:** playLevel(), viewProgress(), updateChildProfile(), viewChildProfile() and modifySettings()
 - Settings class provides settings status.
- **Settings class** includes
 - **Attribute:** sound, music of bool data type
 - **Operations:** toggleSound() and toggleMusic()
- **Guest class** includes
 - **Attribute:** levelStatus of int data type
 - **Operations:** viewAllLevels(), loadNextLevel(), reloadLevel(), quitLevel().
- **Project class** includes
 - **Attribute:** storyIndex and levelIndex. Both are of int data type
 - **Operations:** getUserAge() to get user age, loadNextLevel(), reloadLevel() and quitLevel().
 - User class provides user age.

- **Template class** includes
 - **Attribute:** templateNumber of int data type.
 - **Operations:** getUserAge(), launchNextTemplate(), reloadTemplate() and quitTemplate().
 - User class provides user age.
- **Quiz class** includes
 - **Attribute:** question, answer of string data type and “result” of int data type.
 - **Operations:** loadQuestion(), verifyAnswer(), showCorrectAnswer(), loadNextQuestion() and showResult()

4.5. Dynamic View

4.5.1. State-transition Diagram

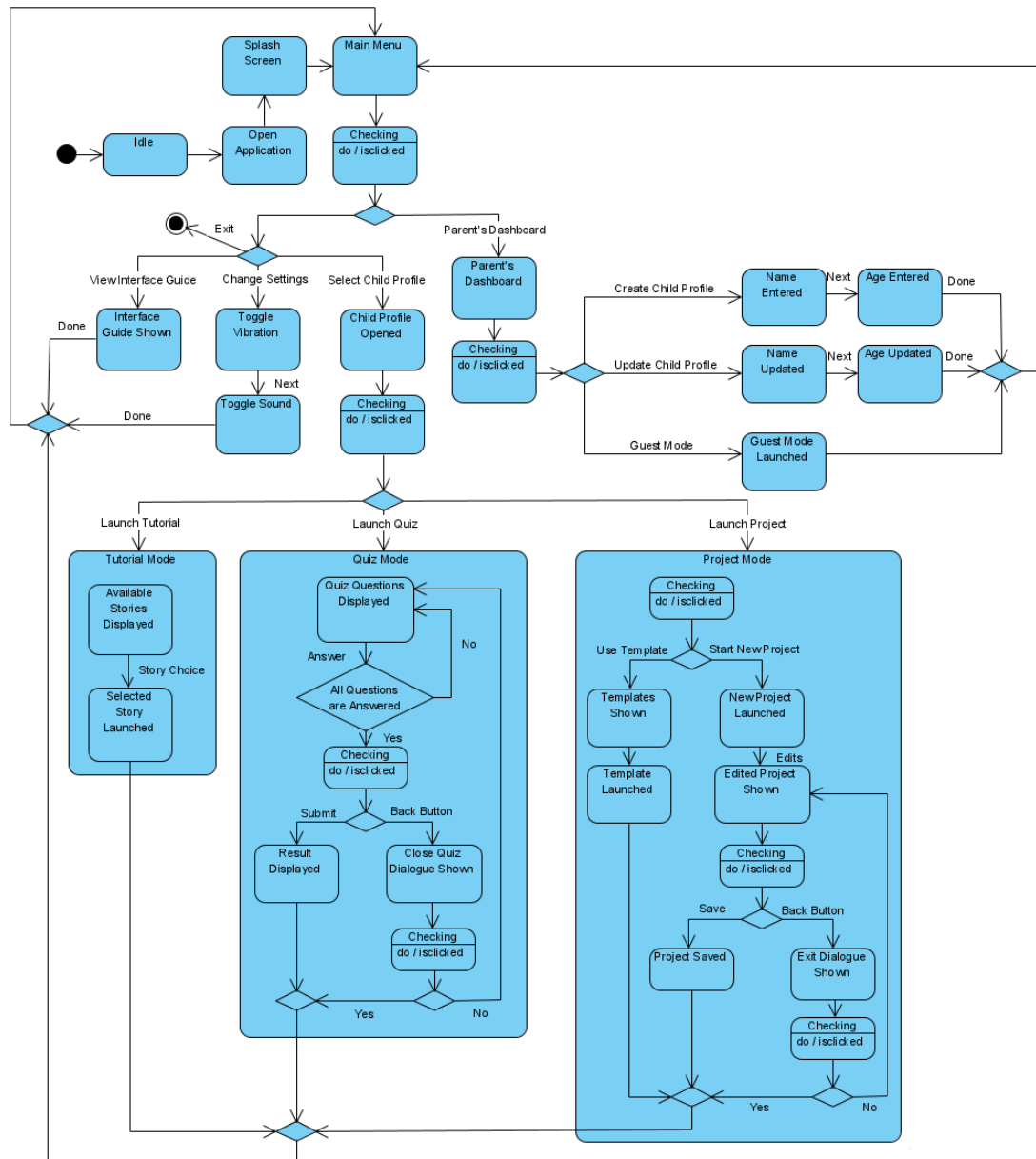


Figure 21 - State Transition Diagram for Android Application

4.6. User Interface Design

4.6.1. Low Fidelity Prototype

Creating low-fidelity prototypes is a quick and easy way to translate concepts into physical artifacts.

Basic models of the product are used for prototypes at low levels of fidelity. You can make them quickly, easily, and inexpensively. All these are used for is showing

developers and all other parties involved in the development process the basic look and feel of the functionality of a software product.

- **Home screen Interface**

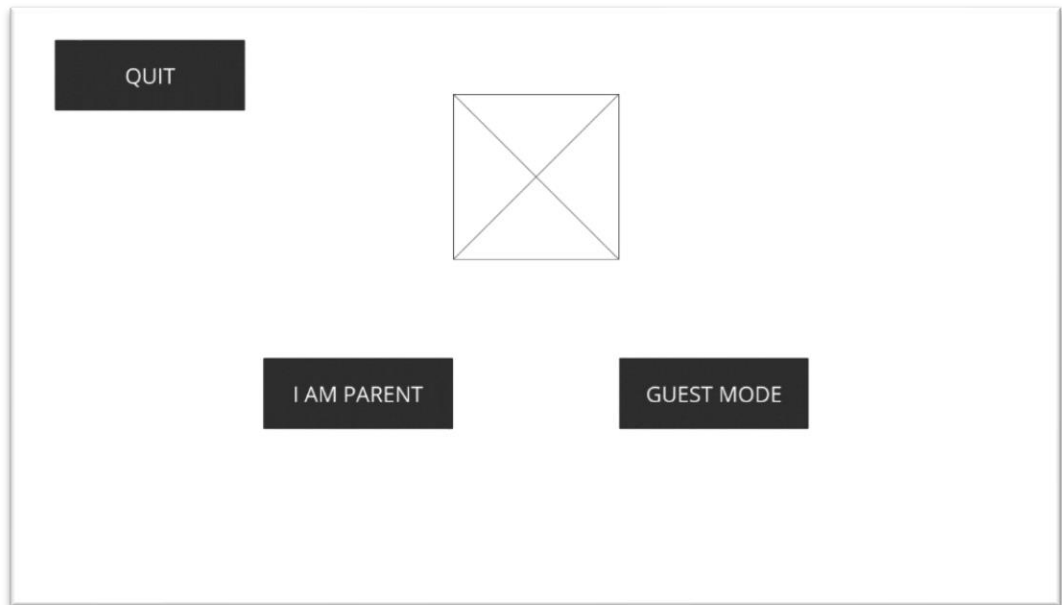


Figure 22 - Main Menu Interface

- **Edit Profile Interface**

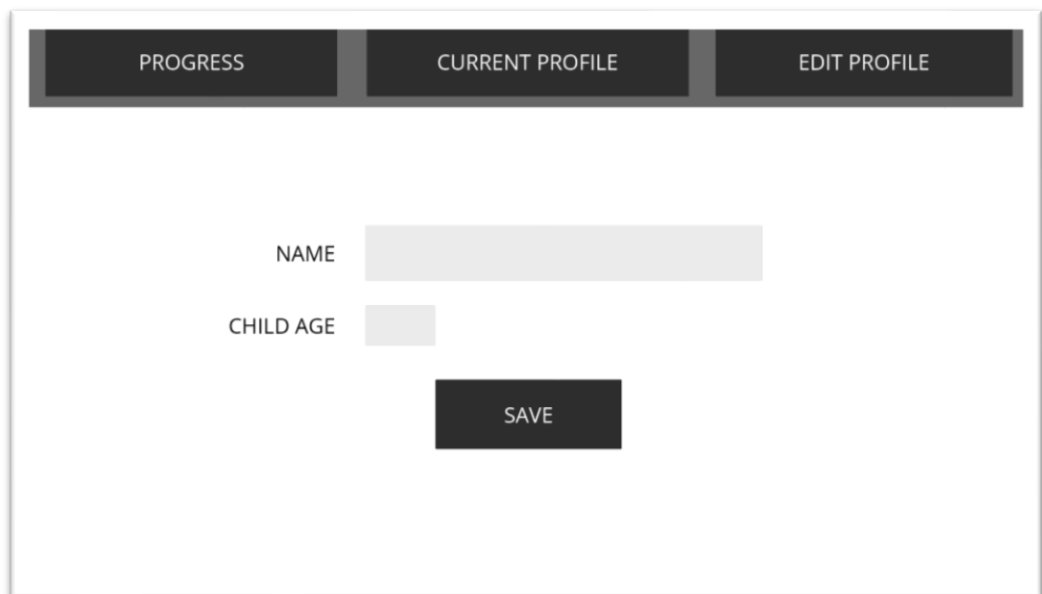


Figure 23 - Edit Profile Interface

- **Levels Interface**

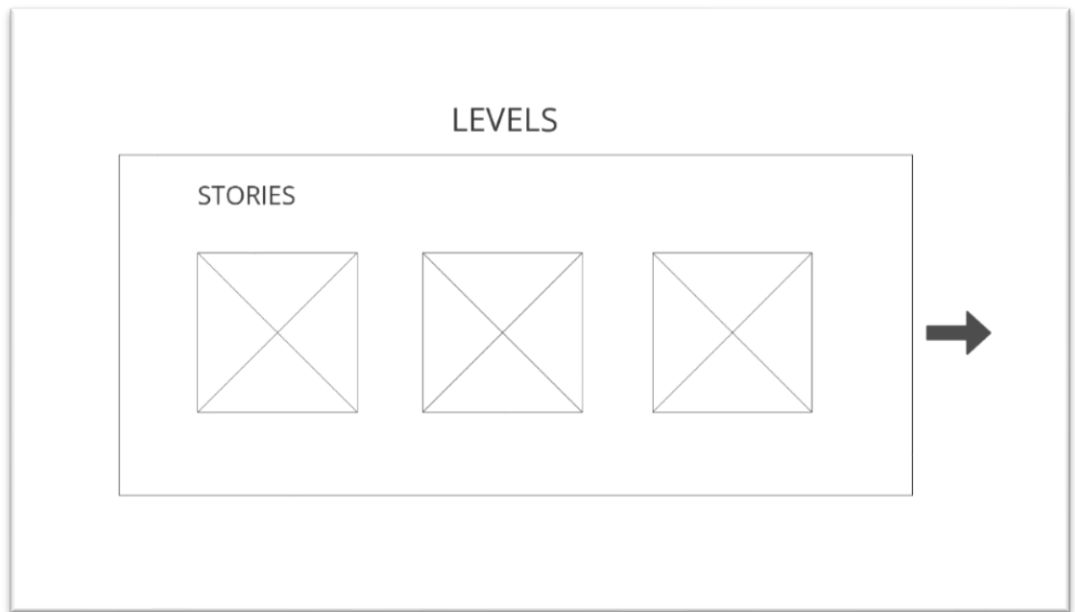


Figure 24 – Levels Interface

- **Story Interface**



Figure 25 – Story Interface

- **Blocks Interface**

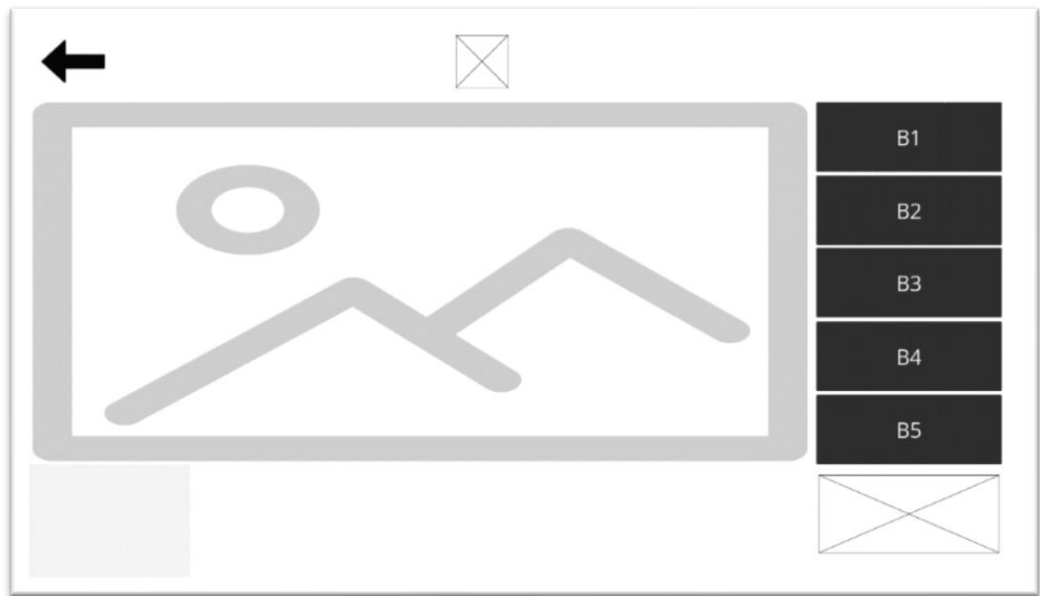


Figure 26 – Blocks Interface

- **Quiz Interface**



Figure 27 - Quiz Interface

4.6.2 High Fidelity Prototype

- **Home screen Interface**

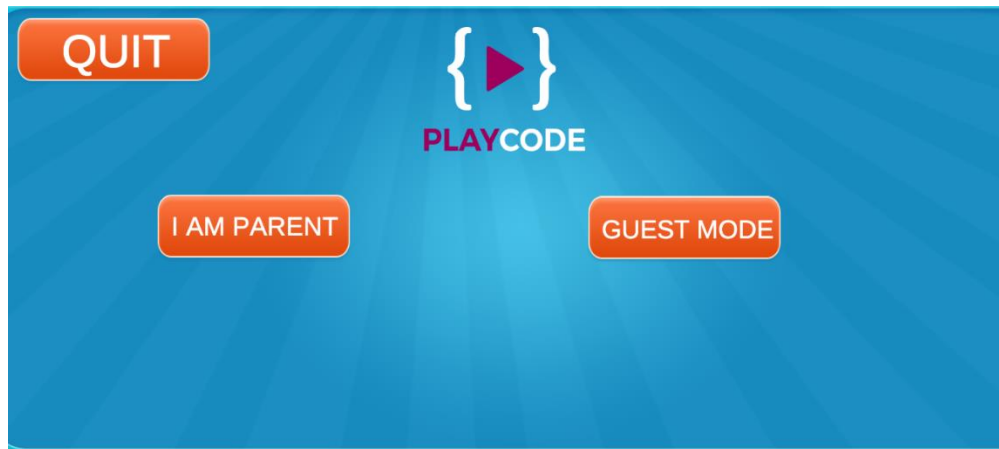


Figure 28 – Home screen Interface

This is the first screen presented to the user as soon as the application is opened. The user is able to use either guest mode or they can create a child profile by selecting “I am Parent” button.

- **Main Menu Interface**

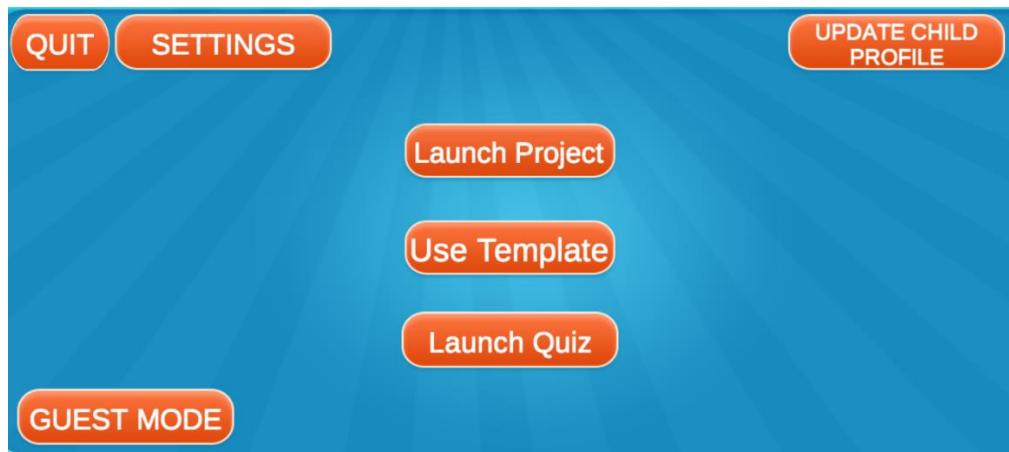


Figure 29 - Main Menu Interface

After a profile has been created, the user is presented with the Main Menu where different modes of the application can be accessed such as the Launch Project, Use template or Launch Quiz.

- **Edit Profile Interface**



Figure 30 - Edit Profile Interface

Upon clicking the “Update Child Profile” button, the user can change the name and age of the child. The difficulty level of the game is determined by the age inputted here.

- **Current Profile Interface**

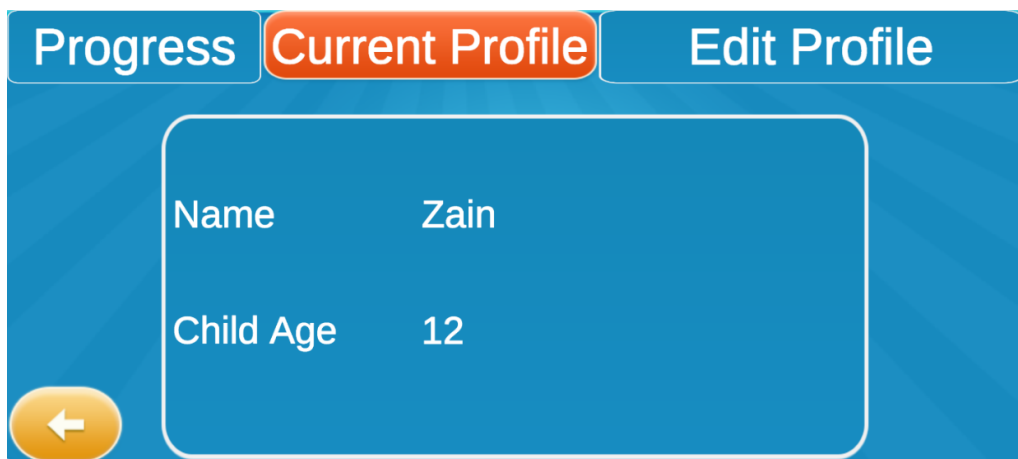


Figure 31 - Current Profile Interface

The user can see the information about the current child profile by clicking the “Current Profile” button. The information displayed is name and age of the registered child.

- **Progress Dashboard**

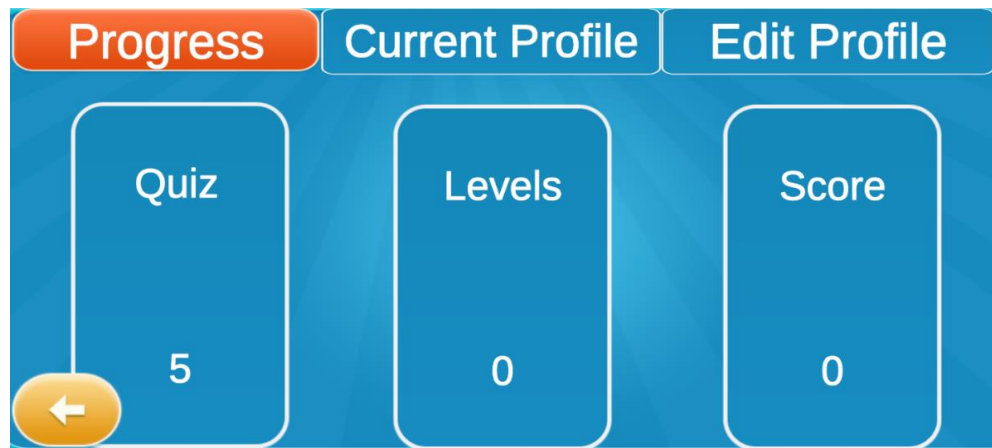


Figure 32 – Progress Dashboard Interface

The user can see the information about the progress of the child by clicking the “Progress” button. The information displayed the number of quizzes attempted, the number of levels finished and the overall scores of the registered child.

- **Settings Interface**

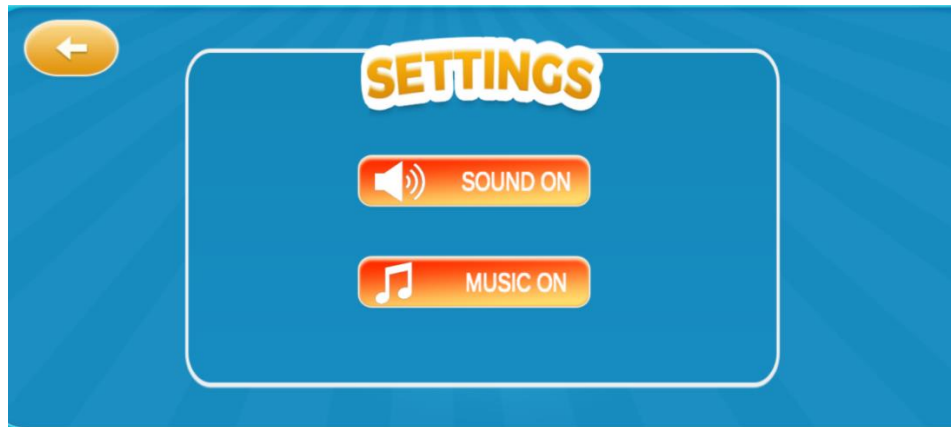


Figure 33 – Settings Interface

The settings menu can be accessed from the main menu where the user can toggle sound and music as they like.

- **Levels Interface**

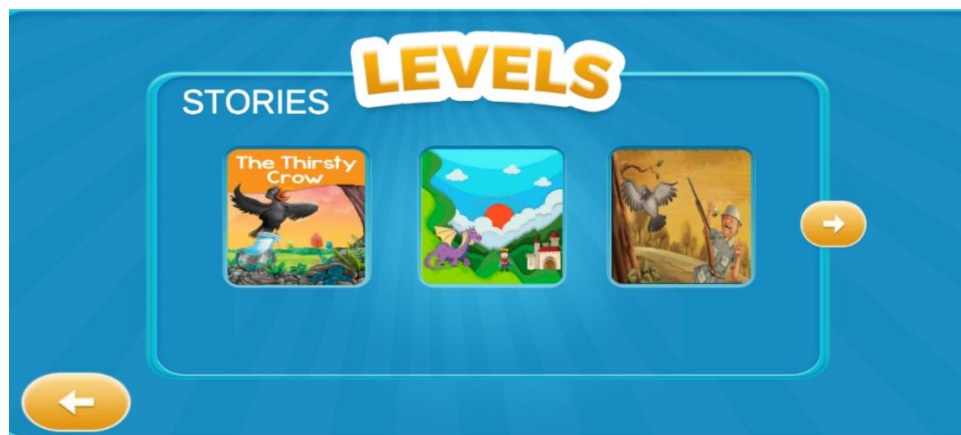


Figure 34 – Levels Interface

Upon selecting the “Launch Project” button on the main menu, the user is presented with the “Levels” screen where they can browse and choose a level to begin learning. The levels are character-based stories and short programs.

- **Objectives Interface**

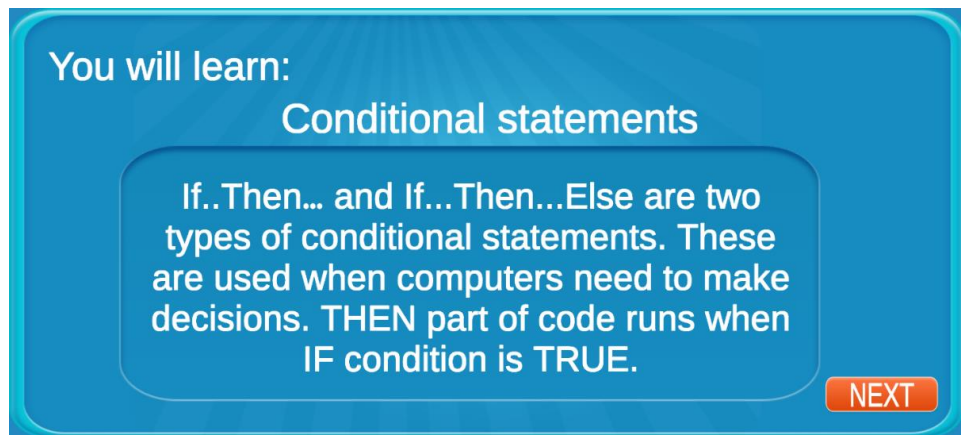


Figure 35 – Objectives Interface

Whenever the user selects a level to play, an objectives screen is displayed. It tells the user what they will learn at each level.

- **Story-Based Level Interface:**



Figure 36 - Story-Based Level Interface

The story-based level screens include animated characters. Upon correct placing of the blocks of code, the characters to the specified task.

- **Output Interface**

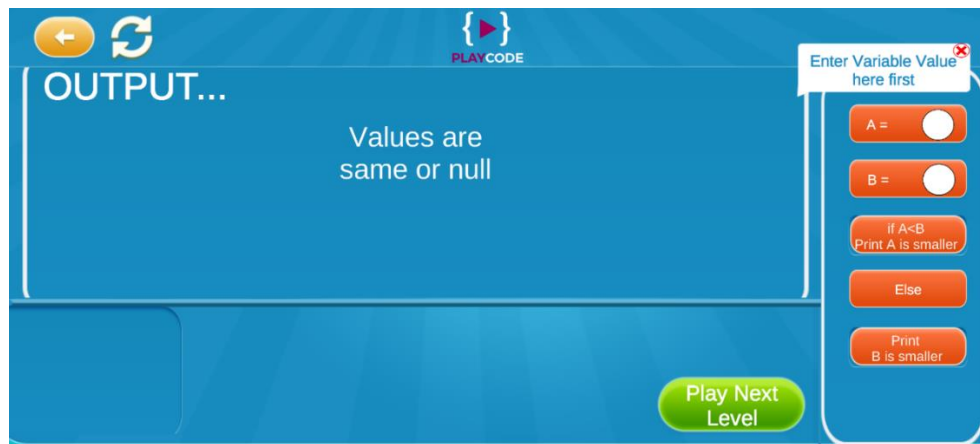


Figure 37 – Output Interface

A section on the left side of the screen displays the result of the input program that makes it interactive for short programs based on levels.

- **Quiz Interface**

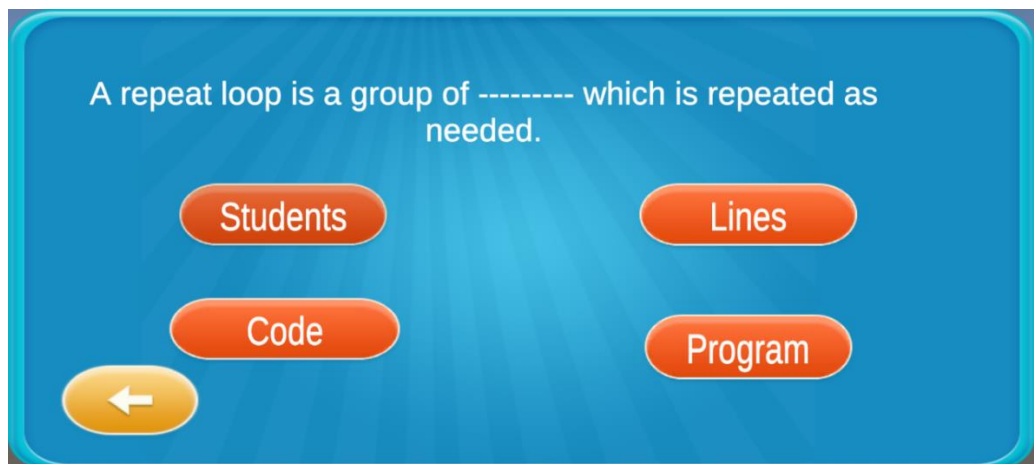


Figure 38 – Quiz Interface

On the main menu, the user can launch the quiz mode to test their knowledge. The quizzes include all the relevant learning material covered in all of the exercises within the application.

- **Result Interface**

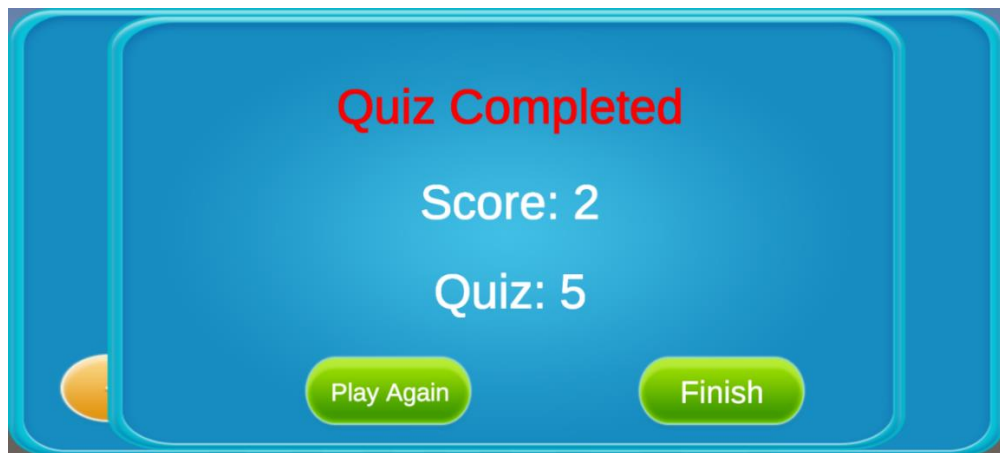


Figure 39 – Results Interface

Upon completing a quiz, the score and the number of the total quizzes taken till that time is displayed. The user can take another quiz right from here or they can finish by clicking the “Finish” button.

4.7. Conclusion

This chapter describes how we approached the design for our project. In addition, we discussed constraints related to system design and architecture. The logical design of our system has been described using the domain model and the class diagram. Furthermore, we have included a state and sequence diagram that shows the dynamic and interactive nature of our project.

Chapter 5

System Implementation

5.1. Tools Used

5.1.1. Unity

In June 2005, Unity Technologies announced and released a Mac OS X-only version of the Unity engine on Apple Inc.'s Worldwide Developers Conference. The engine supports over 19 platforms as of 2021. Games, simulations, and other interactive experiences can be created with the engine, including three- and two-dimensional games, virtual reality, and augmented reality. Many industries outside video gaming have adopted the engine, such as film, automobile, architecture, engineering, and construction for the development and implementation of android application. Interface designing, coding, debugging, and testing are all available within an easy-to-use drag-and-drop interface of Android Studio. It is free to download with its latest version.

5.1.2. Blender

Blender is a free and open-source 3D computer graphics program. We used it to create the 3D models used in our application. This software is also used in the creation of animated films, visual effects, art, 3D printed models, motion graphics, interactive 3D applications, and computer games. Its features include 3D modeling, UV unwrapping, texturing, raster graphics editing, rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, animation, match moving, rendering, motion graphics, video editing, and compositing.

5.1.3. C#

The C# programming language is a general-purpose, multi-paradigm language that embraces strong typing, lexically scoped languages, imperative ones, declarative ones, functional ones, generic ones, object-oriented (class-oriented), and component-oriented disciplines. Around 2000, Microsoft developed the technology as part of the .NET initiative, and Ecma (ECMA-334) and ISO (ISO/IEC 23270) later approved it as an international standard. Mono is a free and open-source project to develop a compiler and runtime for the language. It's a programming language designed for the Common Language Infrastructure (CLI).

5.1.4. Adobe Photoshop

Developed by Adobe Inc., Adobe Photoshop is a raster graphics editor which is available for both Windows and macOS. Thomas and John Knoll created it in 1988. In the years since, Not only has the software become the industry standard in raster graphics editing, but also in digital art as a whole. Photoshop allows the creation of layers for raster images. A number of colour models are supported including RGB, CMYK, CIELAB, spot colour, and duotone. The PSD format and PSB format are used in Photoshop to support these features. Along with raster graphics, Photoshop can also edit and render text and vector graphics. Plug-ins, which run inside Photoshop and give it access to new features, can expand Photoshop's feature set.

5.2. Algorithms

The features and functionality defined above need to be incorporated into an application in a flexible manner, so a few approaches need to be taken. We followed an object-oriented approach to design our application. With a object-oriented approach, we will be able to define classes in a similar manner to what we created when designing.

The classes for each entity are separated out using the object-oriented approach, and all the functionality related to that entity is contained in the classes defined for those entities. The encapsulation of all related attributes and functions is thus achieved. As a result, cohesion is promoted because each class represents all aspects related to its characteristics, such that extra features are not added to the class, thus making it more responsibility-driven.

We have made sure that the classes are not interconnected so that changes in any of them do not affect the others. As a result, our application supports low coupling and high cohesion. In Blender, we designed the 3D models, then we incorporated the 3D models in Unity and coded the functionality in C#. The graphics of different parts of the user interface were created using Adobe Photoshop.

5.3. Methodologies

5.3.1. Agile Development

We used agile software development lifecycle methodology in order to implement our project efficiently. A number of advantages have led to Agile Software Development becoming more popular in various fields. The process of building agile software

lifecycles occurs in such a way that all requirements are gathered, and then iterative evaluations are conducted continuously until the product is complete. Iterations are evaluated accordingly as they are completed on the basis of the requirements you specified. Upon successful completion, the results are moved to the next iteration. Otherwise, the same iteration is followed. Our project has also used this method, as we have gathered all requirements right from the very beginning. The features were then classified into modules. Iterations are then carried out to develop each module. We have moved from one module to the next and so on. In this life cycle, it is possible to change requirements or designs even mid-process or whenever such a need arises. As well, we are able to maintain the code in a much easier manner than waterfall development. The waterfall method does not allow us to change our requirements or designs once we begin developing the project. Programmers working on waterfall projects need to understand most of the project's workflow during the early phases. In addition, another disadvantage is that we cannot begin building our project until our design process has been completed. As agile methods are computed in terms of iterations, these drawbacks can be removed as each module can be created and tested before progressing to the next iteration, thus making the process more efficient. Using agile methodology and applying its principles, we have approached this project.

5.4. Conclusion

Throughout this chapter, we have explored our project's thorough implementation. Our discussion has focused on the programming languages and development environment used. Also, the methodology used for developing our project has been discussed.

Chapter 6

System Testing & Evaluation

6.1 Test Strategy

In order to ensure that the software performs as intended, each piece of software must be tested prior to deployment. In addition to this, it ensures the system's reliability, as well as its usability. A proper testing procedure ensures that every possible bug is found in the system.

The code, implementation, and testing strategy guided us in testing our system. To ensure that the module describes the intended behaviour, we code it first, and then we test it. Once the module is implemented in the overall system, it is subjected to intense testing to ensure optimal functionality when used with the overall system. Depending on when and what type of module we were testing, we performed different types of tests.

6.2 Component Testing

In order to ensure that the software performs as intended, each piece of software must be tested prior to deployment. In addition to this, it ensures the system's reliability, as well as its usability. A proper testing procedure ensures that every possible bug is found in the system.

The code, implementation, and testing strategy guided us in testing our system. To ensure that the module describes the intended behaviour, we code it first, and then we test it. Once the module is implemented in the overall system, it is subjected to intense testing to ensure optimal functionality when used with the overall system. Depending on when and what type of module we were testing, we performed different types of tests.

6.3 Unit Testing

Testing a software unit is the initial phase of the testing process. Unit testing is a process by which the smallest parts of a system, referred to as units, are scrutinized individually for characteristics and behaviour. Performing unit tests manually or automatically is possible.

We adopted a testing strategy known as Use Case Testing to conduct unit tests. Please refer to the Figure 1 use case diagram for details.

6.4 Integrated Testing

An integrated test integrates all modules and features. During this process, various components or modules interact with each other. Integration's main purpose is to check whether each module is operating in accordance with its specifications or not. We have conducted integrated testing in our project by integrating "Launch Project" with "User Age". By displaying levels according to their age, the user is guaranteed to receive content tailored to their age group. A similar test was done with the "Use template" feature, where we tested that the templates are displayed properly based on the age of the player. All modules were then tested as a whole.

6.5 Test Cases

Testing using Use Cases involves checking the correctness of the system's functionality in accordance with the use case. Various modules of the system are evaluated with varying inputs, and their actual performance is compared with that predicted.

- **Test Case: 1**

Test Case No	TC1	QA Test Engineer	Muhammad Abdullah Butt
Objective	Create Child Profile		
System	PlayCode		
Precondition	"I am Parent" mode launched		
Step No.	Execution description	Procedure result	
1	User opens the app.	Expected output: Child profile is created. Actual output: Child profile is created.	
2	User selects "I am a parent"		
3	User enters child name and selects age.		
<u>Comments</u>			
The test was successfully performed.			

Table 18 - Test Case 1 Specification

- Test Case: 2

Test Case No	TC2	QA Test Engineer	Muhammad Abdullah Butt
Objective	Update Child Profile		
System	PlayCode		
Precondition	A Child profile is already created in the app.		
Step No.	Execution description	Procedure result	
1	User opens the app.	Expected output: Child profile is updated. Actual output: Child profile is updated.	
2	User selects "Update Child Profile"		
3	User enters child name and selects age.		
<u>Comments</u> The test was successfully performed.			

Table 19 - Test Case 2 Specification

- Test Case: 3

Test Case No	TC3	QA Test Engineer	Muhammad Abdullah Butt
Objective	Launch Project		
System	PlayCode		
Precondition	A child profile is set up.		
Step No.	Execution description	Procedure result	
1	User opens the app.	Expected output: Level is launched. Actual output: Level is launched.	
2	User selects "Launch Project"		
3	User selects a level		
<u>Comments</u> The test was successfully performed.			

Table 20 - Test Case 3 Specification

- Test Case: 4

Test Case No	TC4	QA Test Engineer	Muhammad Abdullah Butt
Objective	Use Template		
System	PlayCode		
Precondition	A child profile is set up.		

Step No.	Execution description	Procedure result
1	User opens the app.	Expected output: Template is launched. Actual output: Template is launched.
2	User selects “Use Template”	
<u>Comments</u> The test was successfully performed.		

Table 21 - Test Case 4 Specification

- **Test Case: 5**

Test Case No	TC5	QA Test	Muhammad Abdullah Butt Engineer
Objective	Change Settings		
System	PlayCode		
Precondition	Main menu is launched		
Step No.	Execution description	Procedure result	
1	User opens the app.	Expected output: Change in settings is updated. Actual output: Change in settings is updated.	
2	User selects “Settings” icon.		
3	User changes the settings as required.		
<u>Comments</u> The test was successfully performed.			

Table 22 - Test Case 5 Specification

- **Test Case: 6**

Test Case No	TC6	QA Test	Muhammad Abdullah Butt Engineer
Objective	Quit		
System	PlayCode		
Precondition	Application is launched first.		
Step No.	Execution description	Procedure result	
1	User opens the app.	Expected output: App is closed. Actual output: App is closed.	
2	User selects “Settings” icon.		
3	User changes the settings as required.		
4	User selects “Launch Project”.		
5	User selects a level.		

6	User clicks the back button.	
7	User clicks the “Quit” button.	
<u>Comments</u> The test was successfully performed.		

Table 23 - Test Case 6 Specification

- **Test Case: 7**

Test Case No	TC7	QA Test Engineer	Muhammad Abdullah Butt
Objective	Launch Guest Mode		
System	PlayCode		
Precondition	Application is launched		
Step No.	Execution description	Procedure result	
1	User opens the app.	Expected output: User progress is temporarily stored. Actual output: User progress is temporarily stored.	
2	User selects “Guest Mode”.		
3	User selects a level.		
4	User completes a few levels.		
<u>Comments</u> The test was successfully performed.			

Table 24 - Test Case 7 Specification

- **Test Case: 8**

Test Case No	TC8	QA Test Engineer	Muhammad Abdullah Butt
Objective	Close Guest Mode		
System	PlayCode		
Precondition	Guest Mode is launched		
Step No.	Execution description	Procedure result	
1	User opens the app.	Expected output: Guest mode is terminated. Actual output: Guest mode is terminated.	
2	User selects “Guest Mode”.		
3	User selects a level.		
4	User completes a few levels. User clicks the back button.		
<u>Comments</u> The test was successfully performed.			

Table 25 -Test Case 8 Specification

6.6 Integration Testing

Often, integration testing is referred to as an intermediate testing step. As soon as the units are complete, they are integrated with one another and tested to see if they work properly in an integrated state as well. The integration testing ensures that the individual components interact correctly, and that the requirements are met.

6.7 System Testing

6.7.1 GUI Testing

It is the primary goal of the GUI testing process to ensure that the GUI works as intended according to the specifications.

Testing the functionality can be done using Graphical User Interface testing. Also, it helps identify UI flaws. Some defects related to user interfaces are related to the layout, such as the placement of icons and UI elements. There are many UI elements, such as text boxes, icons, and buttons. Aside from that, we also look out for differences in the font style and size used in the text, the colour scheme, the quality of the text etc.

6.7.1.1. GUI Test Cases:

- **Test Case: 9**

Test Case No	TC9	QA Test Engineer	Adeel Ahmad
Objective	Playing first animation		
System	PlayCode		
Precondition	Blocks are arranged in correct order.		
Step No.	Execution description	Procedure result	
1	Drag the blocks in correct order.	Expected output: Crow animation is played. Actual output: Crow animation is played.	
2	Press the "PlayCode" button.		
<u>Comments</u> The test was successfully performed.			

Table 26 - Test Case 9 Specification

- **Test Case: 10**

Test Case No	TC10	QA Test Engineer	Adeel Ahmad
Objective	Playing second animation		
System	PlayCode		
Precondition	Blocks are arranged in correct order.		
Step No.	Execution description	Procedure result	
1	Drag the blocks in correct order.	Expected output: Animation is played. Actual output: Animation is played.	
2	Press the “PlayCode” button.		
<u>Comments</u> The test was successfully performed.			

Table 27 - Test Case 10 Specification

- **Test Case: 11**

Test Case No	TC11	QA Test Engineer	Adeel Ahmad
Objective	Playing third animation		
System	PlayCode		
Precondition	Blocks are arranged in correct order.		
Step No.	Execution description	Procedure result	
1	Drag all the blocks in correct order.	Expected output: Animation is played. Actual output: Animation is played.	
2	Press the “PlayCode” button.		
<u>Comments</u> The test was successfully performed.			

Table 28 - Test Case 11 Specification

- **Test Case: 12**

Test Case No	TC12	QA Test Engineer	Adeel Ahmad
Objective	Assigning Values to Variables		
System	PlayCode		
Precondition	Level is launched.		
Step No.	Execution description	Procedure result	

1	Enter value 9 in the variable1.	Expected output: Value assigned to variable1 and variable2. Actual output: Value assigned to variable1 and variable2.
2	Enter value 09 in the variable1.	
Comments The test was successfully performed.		

Table 29 -Test Case 12 Specification

- **Test Case: 13**

Test Case No	TC13	QA Test Engineer	Adeel Ahmad
Objective	Addition Result Correctness		
System	PlayCode		
Precondition	Level is launched.		
Step No.	Execution description	Procedure result	
1	Enter value 9 in the variable1.	Expected output: 18	
2	Enter value 09 in the variable1.	Actual output: 18	
3	Drag and drop blocks in correct order.		
4	Click “PlayCode” button.		
Comments The test was successfully performed.			

Table 30 - Test Case 13 Specification

- **Test Case: 14**

Test Case No	TC14	QA Test Engineer	Adeel Ahmad
Objective	Null variable correctness		
System	PlayCode		
Precondition	Level is launched		
Step No.	Execution description	Procedure result	
1	Leave variable A null.	Expected output: Message	
2	Leave variable B null.	“Values are same or null”	
3	Drag and drop blocks in correct order.	displayed.	
4	Click “PlayCode” button.	Actual output: Message “Values are same or null” displayed.	
Comments The test was successfully performed.			

Table 31 - Test Case 14 Specification

- **Test Case: 15**

Test Case No	TC15	QA Test Engineer	Adeel Ahmad
Objective	Result Correctness		
System	PlayCode		
Precondition	Level is launched.		
Step No.	Execution description	Procedure result	
1	Enter 4 in variable A.	Expected output: Message “Values are same or null” displayed. Actual output: Message “Values are same or null” displayed.	
2	Enter 4 in variable B.		
3	Drag and drop blocks in correct order.		
4	Click “PlayCode” button.		
<u>Comments</u> The test was successfully performed.			

Table 32 - Test Case 15 Specification

- **Test Case: 16**

Test Case No	TC16	QA Test Engineer	Adeel Ahmad
Objective	Result Correctness		
System	PlayCode		
Precondition	Level is launched		
Step No.	Execution description	Procedure result	
1	Enter 4 in variable A.	Expected output: B is smaller. Actual output: B is smaller.	
2	Enter -9 in variable B.		
3	Drag and drop blocks in correct order.		
4	Click “PlayCode” button.		
<u>Comments</u> The test was successfully performed.			

Table 33 - Test Case 16 Specification

- **Test Case: 17**

Test Case No	TC17	QA Test Engineer	Adeel Ahmad
Objective	Result Correctness		
System	PlayCode		
Precondition	Level is launched		
Step No.	Execution description	Procedure result	

1	Enter 10 in variable A.	Expected output: A is equal to 10, B is equal to 10. Actual output: A is equal to 10, B is equal to 10.
2	Enter 10 in variable B.	
3	Drag and drop blocks in correct order.	
4	Click “PlayCode” button.	
<u>Comments</u> The test was successfully performed.		

Table 34 - Test Case 17 Specification

- **Test Case: 18**

Test Case No	TC18	QA Test Engineer	Adeel Ahmad
Objective	Result Correctness		
System	PlayCode		
Precondition	Level is launched		
Step No.	Execution description	Procedure result	
1	Enter 10 in variable A.	Expected output: A is equal to 10 Actual output: A is equal to 10	
2	Enter -1 in variable B.		
3	Drag and drop blocks in correct order.		
4	Click “PlayCode” button.		
<u>Comments</u> The test was successfully performed.			

Table 35 - Test Case 18 Specification

- **Test Case: 19**

Test Case No	TC19	QA Test Engineer	Adeel Ahmad
Objective	Result Correctness		
System	PlayCode		
Precondition	Level is launched		
Step No.	Execution description	Procedure result	
1	Enter -1 in variable A.	Expected output: A & B is not equal to 10 Actual output: A & B is not equal to 10	
2	Enter 99 in variable B.		
3	Drag and drop blocks in correct order.		
4	Click “PlayCode” button.		

<u>Comments</u> The test was successfully performed.
--

Table 36 - Test Case 19 Specification

- **Test Case: 20**

Test Case No	TC20	QA Test Engineer	Adeel Ahmad
Objective	Result Correctness		
System	PlayCode		
Precondition	Level is launched		
Step No.	Execution description	Procedure result	
1	Leave variable "Ali height" null.	Expected output: Message "Fill value first" displayed. Actual output: Message "Fill value first" displayed.	
2	Leave variable "Sara height" null.		
3	Drag and drop blocks in correct order.		
4	Click "PlayCode" button.		
<u>Comments</u> The test was successfully performed.			

Table 37 - Test Case 20 Specification

6.8 Load Testing

The purpose of load testing is to measure the software's response to challenges, which is accomplished via placing demands on the system. Load tests are usually conducted to determine how a system performs during normal and peak load conditions. By doing so, the tester is able to identify what is the maximum operating capacity of the system and is able to pinpoint bottlenecks. Through the app, we simultaneously opened and closed different modules in our system to test its load-bearing capacity.

6.9 Security Testing

Tests for security reasons are conducted in order to identify vulnerabilities in software. A system passing a security test, however, does not guarantee that it will not be intruded upon or accessed by unauthorized parties. In addition to integrity and confidentiality testing, there is authentication and authorization testing. Throughout our system, security has always been a priority. Currently, we are not connected to any online

servers. In this case, the user's private information can never be shared. Furthermore, the application only needs the name and age of the user to function. The user's device stores this information locally.

6.10 Results & Evaluation

Test cases were carried out individually, and all were successfully completed. The features passed all of the tests after they were tested.

6.11 Conclusion

In this chapter, we have discussed various types of testing and the testing procedure we will utilize for our project. Testing has been done on all features of the project. Using use cases as a starting point, the test cases were created.

Chapter 7

Conclusion

7.1. Contributions

Numerous books aimed at teaching kids programming have already been written in the past. Additionally, many mobile applications have been developed that do the same. Reading books is not a favourite pastime among kids these days. Mobile phones, however, are well known to kids. In this way, kids can learn to program using smartphones. There are already smartphone apps available to teach programming, but the process is onerous. It's usually boring puzzles that they use. Our idea was to try something different. Interactive blocks of code were used as well as character-based stories.

In Pakistan, there has never been an app of this kind before. This project fills an important need in the education sector in this country because it implements interactive teaching methods and uses up-to-date information. It will be possible for parents to keep track of how their child is doing. By allowing children to take quizzes within the app, they are able to figure out how learning is going for them.

Teachers and parents can save time and resources by using PlayCode to teach their children programming. This project provides children with many challenges to evaluate their abilities. They will learn coding by using code blocks. The challenges they will have to deal with are fundamental in nature. This application was developed for the Android operating system. The platform is open source and provided by Google. The fact that Android is the most widely used operating system in the world could lead PlayCode to be compatible with Android devices used by 2.5 billion individuals.

7.2. Reflections

PlayCode is an application that is developed to help kids who aren't familiar with computer programming concepts. By using engaging and attractive short programs and stories, our project effectively conveys programming concepts. Kids can, therefore, understand key concepts such as code, algorithm, loop, bug, variable, sequence, and conditional statements. Our main competitors are computer schools with outdated

learning materials and apps. These schools lack the access to modern technology that enables them to effectively teach these subjects. Thus, we have the upper hand in this area and that is the strength of our project.

7.3. Future work

In a single application, PlayCode already offers a lot. Nevertheless, future improvements can still be made to it. With further research, more advanced and innovative approaches to teaching programming concepts can be implemented in the application. In addition, more and more programming exercises in the form of moral stories could be added. Once the app becomes more popular and its user base grows, we will consider advertising as a means of monetization. The revenue generated by these ads will be used to support the further development and maintenance of our application. Thus, this application will only continue to grow and improve.

REFERENCES

- [1] S. Cooper, W. Dann, and R. Pausch. *Teaching Objects-first in Introductory Computer Science*. In Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education, SIGCSE '03, pages 191–195, New York, NY, USA, 2003. ACM.
- [2] B. Moskal, S. Cooper, and D. Lurie. *Evaluating the Effectiveness of a New Instructional Approach*. In Proceedings of the SIGCSE technical symposium on Computer science education, 2005.
- [3] O. Meerbaum-Salant, M. Armoni, and M. M. Ben-Ari. *Learning Computer Science Concepts with Scratch*. In Proceedings of the Sixth International Workshop on Computing Education Research, ICER '10, pages 69–76, New York, NY, USA, 2010. ACM.
- [4] T. W. Price and T. Barnes. *Comparing Textual and Block Interfaces in a Novice Programming Environment*. In Proceedings of the Eleventh Annual International Conference on International Computing Education Research, ICER '15, pages 91–99, New York, NY, USA, 2015. ACM.
- [5] Hubwieser, P., Armoni, M., Giannakos, M. N., & Mittermeir, R. T. (2014). *Perspectives and visions of computer science education in primary and secondary (K-12) Schools*. ACM Transactions on Computing Education (TOCE), 14(2), 7
- [6] Jones, S. P. Computing at school in the UK. 2013. <http://research.microsoft.com/en-us/um/people/simonpj/papers/cas/computingatschoolcacm.pdf>
- [7] Tuhkala, A., Wagner, M.-L., Nielsen, N., Iversen, O. S., & Kärkkäinen, T. *Technology comprehension: Scaling making into a national discipline*. Paper presented at the proceedings of the conference on creativity and making in education, 2018

APPENDIX A

Abbreviation	Acronyms
UC	Usecase
3D	3 Dimensions
TC	Testcase
CLI	Common Language Infrastructure

Table 38 - Appendix A