# ROLL NO: A034

**CONTROL ID:** 2019060356

**NAME: Ajay Balaji Pawar**

**Break The Brick**

**THE APPROVAL PROJECT PROPOSAL**

***(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)***

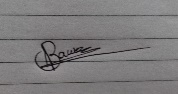
PNR **No:**

|  |  |
| --- | --- |
| 1. Name of the Student: Ajay Pawar | |
|  | |
| 2. Title of the Project: Break The Brick | |
|  | |
| 3. Name of the Guide: Mrs. Mohini Bhole. | |
|  | |
| 4. Is this your first submission? Yes | |
|  | |
| Student Name: | Ajay Pawar |
| Student Control ID: | 2019060356 |
| Student Roll No.: |  |

**DECLARATION**

I hereby declare that the project entitled,” Break The Brick” done at place where the project is done, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university. The project is done in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY) to be submitted as final semester project as part of our curriculum.

**Name: Ajay Pawar**

****

**Signature:**

**ACKNOWLEDGEMENT**

I am glad to say that, I have satisfactorily reached my aims and intentions, to make this project a success. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

I am highly indebted of my guide, Mrs. Mohini Bhole for her guidance and constant supervision as well as for providing necessary information regarding the project.

I would also like to extend my gratitude towards the Head of the Department, Mrs. Pournima Bhangae for providing me with all the facilities that was required.

This opportunity has given me a valuable experience about software development for which I shall be thankful for the years to come.

Ajay Pawar

**INDEX**

|  |  |  |
| --- | --- | --- |
| Chapter No. | Name | Page No. |
|  | **SYNOPSIS** | 1 |
| 1 | **Chapter 1** | 3 |
|  | 1.1 Background | 3 |
|  | 1.2 Objectives | 3 |
|  | 1.3 Purpose, Scope and Applicability | 3 |
|  | 1.4 Achievements | 4 |
| 2 | **Chapter 2:**  **(Survey Of Technologies)** | 5 |
| 3 | **Chapter 3:** | 8 |
|  | **3.1 Requirements & Analysis** | 8 |
|  | 3.1.1 Problem Definition | 8 |
|  | 3.2 Requirement Specification | 8 |
|  | 3.2.1 Requirement Gathering | 8 |
|  | 3.2.1.1 Information gathering | 8 |
|  | 3.2.2. Requirement Analysis | 8 |
|  | 3.2.2.1 Functional Requirements | 8 |
|  | 3.2.2.2 Non-Functional Requirements | 8 |
|  | 3.2.2.3System requirement | 9 |
| 4 | Chapter 4 | 10 |
|  | 4.1 Use Case Diagram | 10 |
|  | 4.2 Activity Diagram | 12 |
|  | 4.3 Sequence Diagram | 13 |
|  | 4.4 UI Designs | 16 |
|  | 4.5 Test Cases | 18 |
| 7 | Bibliography &References | 21 |
|  | Web References | 21 |

**ANNEXURES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Fig. Name** | **Page No.** |
| 1 | Architecture Design | 1 |
| 2 | CLIENT-SERVER MODEL | 2 |
| 3 | USE CASE DIAGRAM | 10 |
| 4 | Notation for Activity Diagram | 12 |
| 5 | Activity diagram | 13 |
| 6 | Notation for Sequence Diagram | 14 |
| 7 | Sequence diagram for Play | 14 |
| 8 | Sequence diagram for Setting | 15 |
| 9 | Sequence diagram for Avatar | 15 |
| 10 | User Interface Design Start UI | 16 |
| 11 | User Interface Design Setting | 16 |
| 12 | User Interface Design Levels | 17 |
| 13 | User Interface Design On Play | 18 |

**Synopsis**

**1.Title:**

**Break The Brick**

**2. Statement about Problem:**

My project is Computer Game which is built in the C# Programing language.

**3. Why this topic:**

The reason for this topic is Aptitude and Interest.

Aptitude means I want to build the natural ability in this field and I have interest to build a game as it gives me important experience.

**4. Objective and the scope:**

Well Objective of this project is to gain experience in game building.

For user it way to play simple mini game at very basic level.

Scope of project: it is can work mobile devices.

**5. Methodology:**

**ITERATIVE MODEL**

In this project, the Iterative Model will be used. The **Iterative model** is a particular implementation of a software development life cycle (SDLC) that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete. The system is divided into several builds or iterations. During each iteration, the module goes mainly through four processes: Requirements, design, implementation, and testing. These four processes are repeated until the software is finally completed.

**6. Proposed Architecture:**

**MVC**

This system will work on MVC architecture, The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. MVC is one of the most frequently used industry-standard web development frameworks to create scalable and extensible projects.

**7. Requirements:**

**7.1 Software Requirements:**

* **Unity Development Tool.**
* **Adobe Photo Shop.**

**7.2 Hardware Requirements:**

* **Desktop or Laptop**

**7.3 Platform:**

* **Unity**

**Contribution:**

It’s gives experience of game building to developer.

**Conclusion:**

In conclusion the project provides the joy to the user and experience foe developer.

**BREAK THE BRICK**

**CHAPTER 1**

**Introduction**

**1.1 Background: -**

Break The Brick is a game consists of bricks on the top of the screen, a flying ball and a movable paddle on the bottom of the screen.

The paddle is controlled by the user to move either left or right. Player need to use this paddle to prevent the ball from falling out of the screen. Three unbreakable walls on the side of the screen are used to deflect the ball.

The collision between the ball and bricks can make brick downgraded or disappear. The speed of ball gradually goes up with time.

The game will continuously proceed as long as the ball is not falling out of the screen. During playing, the user is able to pause the game and continuous again.

**1.2 Objectives: -**

* My game is going to build with Graphics and if possible, sound effects that is the game consist of colorful bricks, walls and control bar (if possible, sounds also).
* Players rank and score is going to store to take records of players in my game control bar can have special abilities.

**1.3 Purpose, Scope and applicability**

**1.3.1 Purpose**: -

I want to build the natural ability in this field and I have interest to build game development logic.

This project is going to develop in C# language and UNITY as its platform.

**1.3.2 Scope: -**

Android 10 mobile device.56

**1.3.3 Applicability: -**

My project provide enjoyment to everyone.

**1.4 Achievements: -**

My project gives the experience how to developed game.

**CHAPTER 2**

**Survey of Technologies**

**Programming Framework (Front end)**

* **Java**

Java is the popular programming language is used to develop Android applications.

Java.

Firstly, Java was the official language for Android App Development (but now it was replaced by Kotlin) and consequently, it is the most used language as well. Many of the apps in the Play Store are built with Java, and it is also the most supported language by Google.

* **Python 3.0:**

Python 3.0 broke backward compatibility, and much Python 2 code does not run unmodified on Python 3. Python's dynamic typing combined with the plans to change the semantics of certain methods of dictionaries, for example, made perfect mechanical translation from Python 2.x to Python 3.0 very difficult.

* **C#:**

It is an object-oriented programming language created by Microsoft that runs on the .NET Framework.

C# has roots from the C family, and the language is close to other popular languages like [C++](https://www.w3schools.com/cpp/default.asp) and [Java](https://www.w3schools.com/java/default.asp).

The first version was released in year 2002. The latest version, **C# 8**, was released in September 2019.

C# used for development of Games, Console app, Web app.

**Database (Back-end):**

* **Firebase Realtime Database**

Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android and [Web apps](https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app). Firebase provides tools for tracking analytics, reporting and fixing app crashes, creating marketing and product experiment.

* **SQLite**

SQLite is an opensource SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation.

SQLite supports all the relational database features. In order to access this database, you don't need to establish any kind of connections for it like JDBC, ODBC etc.

* **MySQL:**

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. It is released under an open-source license. It works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. It also works very quickly and works well even with large data sets.

* **Oracle:**

Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information.  Oracle is fully scalable relational database architecture and is often used by global enterprises, which manage and process data across wide and local area networks. The Oracle database has its own network component to allow communications across networks

* **Game Engine:**

A game engine is the architecture that developers use to run the game. Average game engine provides developers with a way to add things like:

* physics
* input
* rendering
* scripting
* collision detection
* artificial intelligence
* and more without the need to program them
* **Graphics:**

A capable graphic design tool not only enhances efficiency, but it also helps in putting stunning creativity into real-world design.

Average game engine provides developers with a way to add things like:

* Adobe Photoshop
* Adobe InDesign
* CorelDraw

**Technology used by me:**

For this project, I will be using **Unity Engine** and **C#**  as its scripting langauge, **Adobe Photoshope** as Graphics Design

**Chapter 3:**

**Requirements and analysis**

**3.1 Problem Definition:**

This project is about to make a mobile game name “Break The Brick” which is very simple game to play on android device.

Provide good graphic and user interface to user.

* 1. **Requirement Specification:**
     1. **Requirement Gathering:**

**Information gathering**

* **Interviewing the users:**
* Which type of music do you prefer in game?
* How many levels do you prefer in game?
* What types of level do you like?
* What kind of User Interface do you prefer?
* What type of Controls do you prefer in my game?

**3.2.2 Requirement Analysis:**

1. **Functional Requirement**

* Admin must be able to update the game.
* Game must run properly until user (Player) terminates it.
* Game has to install and uninstall as per choice of user.
* Game UI must support the user’s device.

1. **Non-functional Requirement**

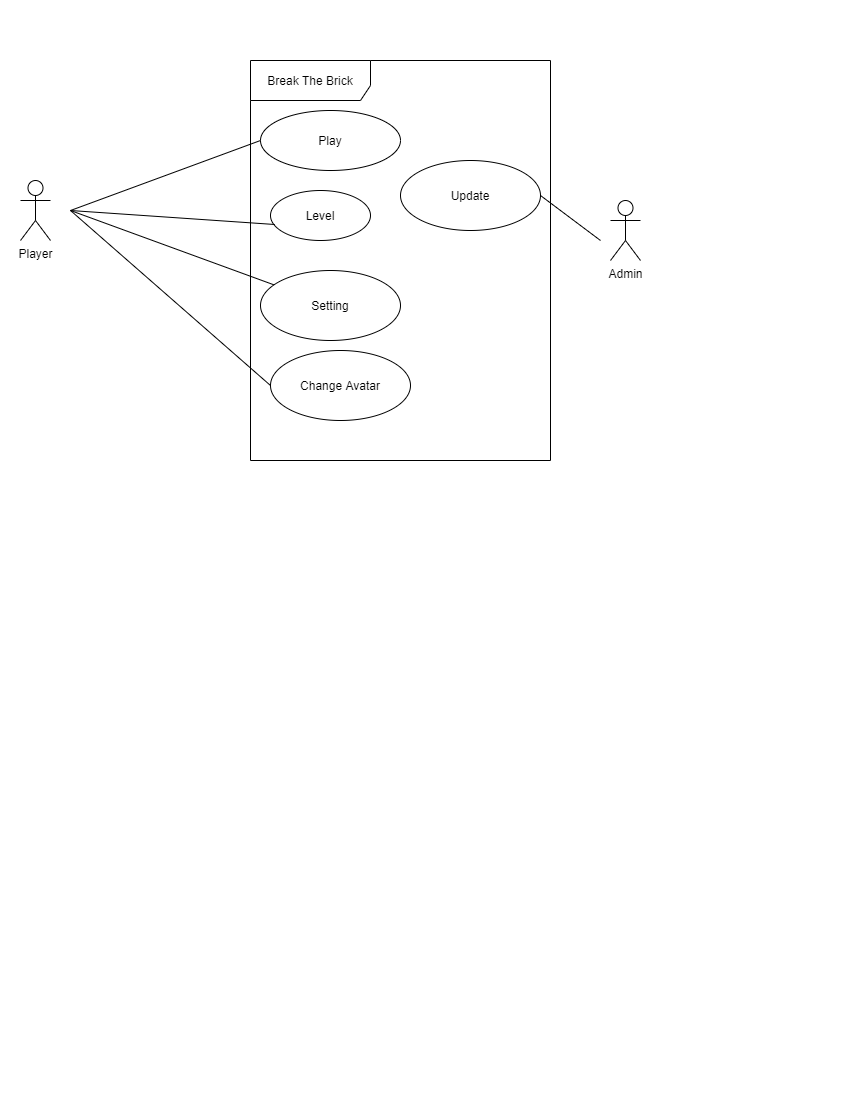
* Portability: The game is portable as it can run on any android device
* Scalability: The game can further modify into the future.
* Performance verifiable: appropriate, fast, portable, lightweight, small, large, maximize, minimize also include in it.

1. **System requirement:**

* *Levels:* The game should display the levels which are already cleared by user or user’s current level.
* *Play:* The game should redirect control to level UI after clicking on play button.
* *Setting:* Changed settings should apply on Game objects like music and sound.
* *Skin:* Changed skin should apply on Game objects like ball and slider.

**CHAPTER 4**

***4.1 USE CASE DIAGRAM:***

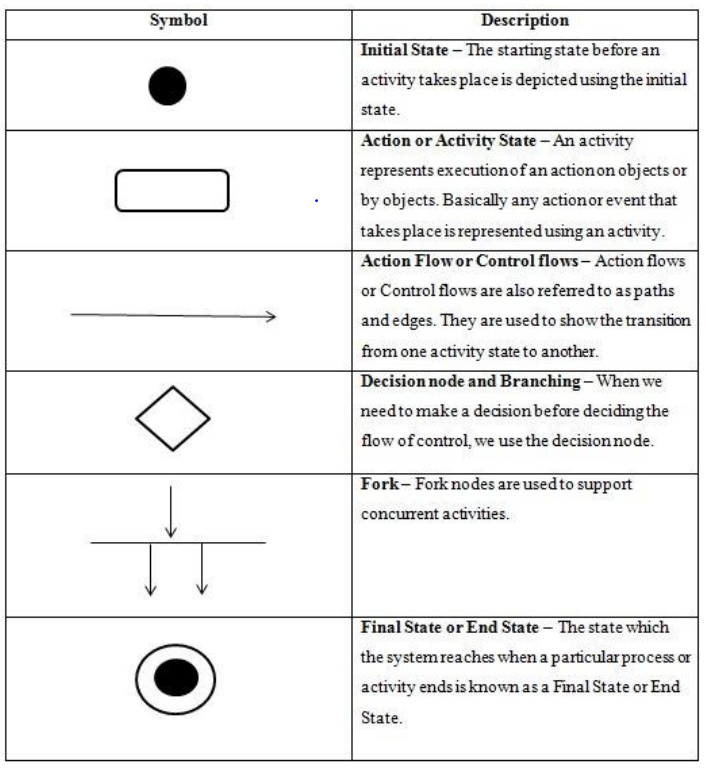
******

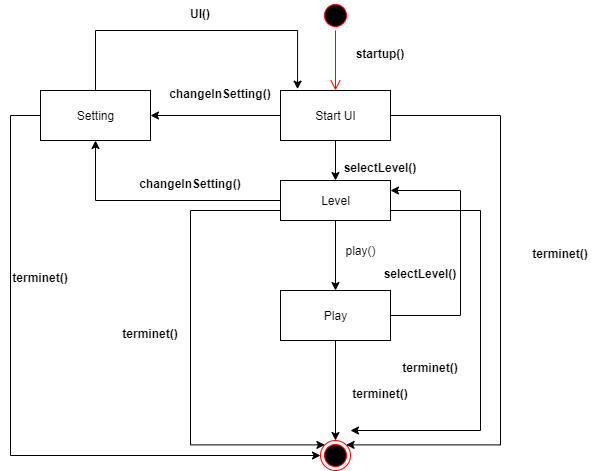
* ***Use Case: Play***
* *Description: On click the game is going to Play.*
* *Actor: Player.*
* *Pre-Condition: Game must be running.*
* *Post-Condition: Player have can select Level of choice.*
* ***Use Case: Level***
* *Description: Player can select the level of choice.*
* *Actor: Player*
* *Pre-Condition: Player must have selected to play and level must have been cleared or current level*
* *Post-Condition: Player have can continue play.*
* ***Use Case: Setting***
* *Description: on click player have access to game control settings.*
* *Actor: Player*
* *Pre-Condition: Game must be running.*
* *Post-Condition: Player can change the game user settings.*
* ***Use Case: Change Avatar***
* *Description: on click player can change the avatar of slider.*
* *Actor: Player*
* *Pre-Condition: Game must be running.*
* *Post-Condition: Player continue on playing with new avatar.*
* ***Use Case: Update***
* *Description: Update and modify game and maintains of game*
* *Actor: Admin*

**4.2 Activity diagram:**

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

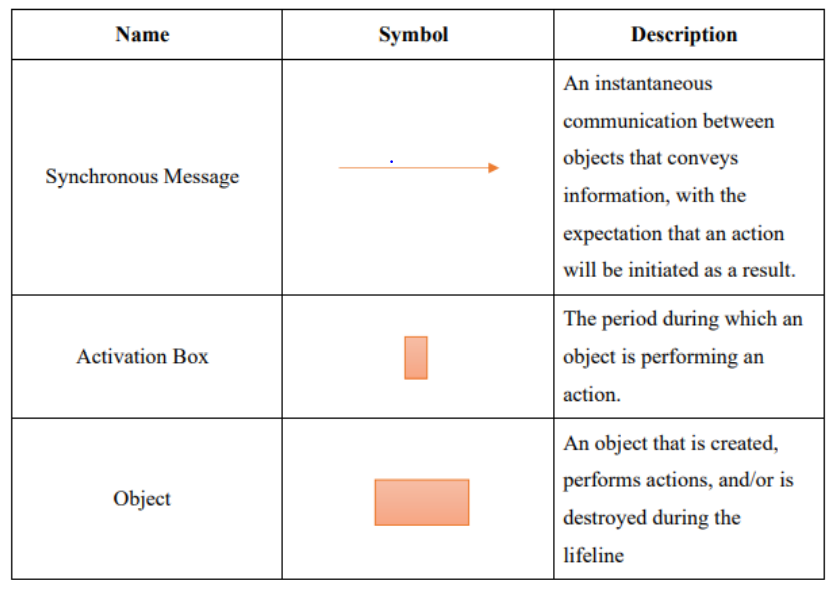
**Notation for Activity Diagram:**



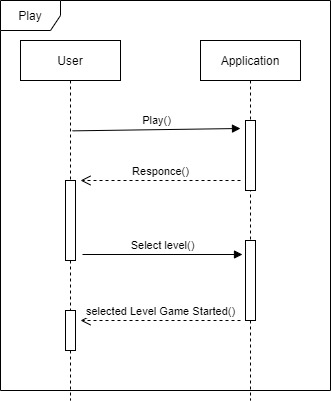
Activity diagram:

**4.3 Sequence Diagram:**

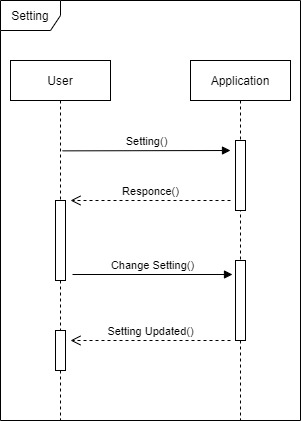
A sequence diagram in a Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams typically are associated with use case realizations in the Logical View of the system under development.

**Notation for Sequence Diagram:** 

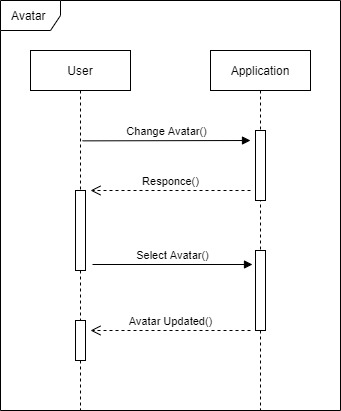
**1. Play:**



**2 Setting:**



**3 Avatar:**

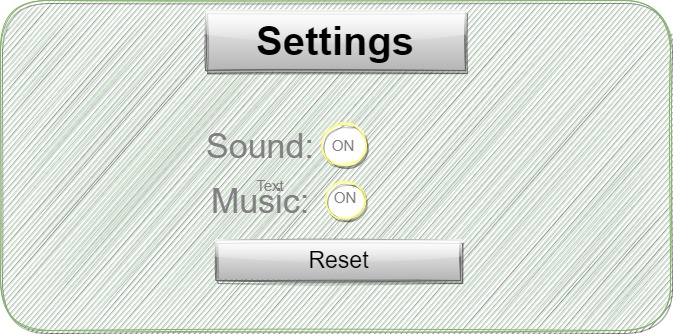


**4.4 User Interface Design:**

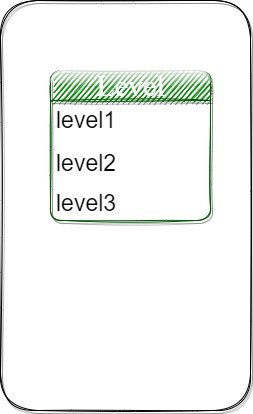
**1 Start UI:**



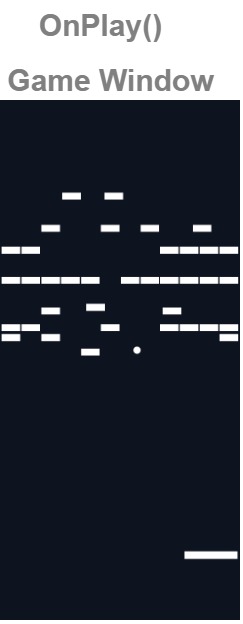
**2 Setting:**



1. **Levels:**



**4 On Play:**



**4.5 Test Cases**

**1 User UI:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case** | **Expected**  **Output** | **Actual Output** | **Pass/Fail** | **Remark** |
| **1** | On Click Play | Level UI Open |  |  |  |
| **2** | On Click Skin | Avatar selection list for slider and ball open |  |  |  |
| **3** | On Click Setting | Setting menu open |  |  |  |

1. **Setting Menu UI:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case** | **Expected**  **Output** | **Actual Output** | **Pass/Fail** | **Remark** |
| **1** | On Click Music | If music is on then it turns off  Or  If music is off then it turns on |  |  |  |
| **2** | On Click Sound | If sound is on then it turns off  Or  If sound is off then it turns on |  |  |  |

1. **Skin UI:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case** | **Expected**  **Output** | **Actual Output** | **Pass/Fail** | **Remark** |
| **1** | List of Avatar | List of unlock avatar appears |  |  |  |
| **2** | Select Avatar | Selected avatar changes user’s avatar (ball or slider) |  |  |  |

1. **Game Play UI:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case** | **Expected**  **Output** | **Actual Output** | **Pass/Fail** | **Remark** |
| **1** | Slider Touch Control | Slider move according to players touch movement | Not following touch of user | **Fail** |  |
| **2** | Ball Bounce | Ball bounce on every single object | Bounce properly | **Pass** |  |
| **3** | Braking of bricks | Brick gets destroy on hitting with wall | not beak the brick | **Fail** |  |
| **4** | Live Count | Live count decrement by 1 when condition is trigger  [Condition 1: ball fell out of slider  Condition 2: ball destroy] | Live count changes according to ball fall  And life power up | **Pass** |  |
| **5** | Display Live Count | Current live count display on screen | Display properly | **Pass** |  |
| **6** | Score Count | Score count incremented by 1 or 2 on collision with points brick | Changes according to brick point | **Pass** |  |
| **7** | Display Score | Current score count display on screen | Display  properly | **Pass** |  |

**Bibliography & References**

* Software Engineering, “Ian Somerville”, 8th Edition, Pearson Education.
* Software Testing Principles, Techniques and Tools | M. G. Limaye | TMH •

**Web References:**

* W3 School: https://www.w3schools.in/ (3 days)
* https://lucid.app/lucidchart/ (8 day)
* YouTube: https://www.youtube.com/ (16 days)
* Draw io: https://app.diagrams.net/ (3 days)