

Fill the Word (Urdu)
An Android Gaming Application



Submitted By: Arfa Batool
Supervised By: Dr. Onaiza Maqbool

Department of Computer Science
Quaid-i-Azam University
Islamabad, Pakistan
Session (2013-2017)

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Author's Declaration

I _____ **Arfa Batool**_____ hereby state that my BSCS final year project report titled **_Fill the word (Urdu) Final Year Project Report**_____ is my own work and has not been submitted previously by me for taking any degree from _____ **Quaid-i-Azam University Islamabad** _____ or anywhere else in the country/world.

At any time if my statement to be found incorrect even after my graduation, the university has the right to withdraw my BSCS degree.

Name of Student: Arfa Batool

Date: 17th July, 2017

ACKNOWLEDGMNET

All praise is due to Allah, the Lord of the Worlds. He bestowed His favors abundantly. I bear witness that there is no deity save Allah, having no associates. All the praises and thanks are due to Him. Thanks to Almighty Allah, who showered his blessings on me and sanctified me with the capabilities that led to the completion of my final year project report.

I would also like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to my final year project manager, Dr. Onaiza Maqbool, whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project especially in writing this report. I would also like to show my gratitude to my professors who encouraged me and shared their pearls of wisdom with me during the course. I specially thank my parents from the core of my heart and pay tribute to them for providing me guidance and encouragement.

Abstract

Fill the word is an android gaming application. It is our everyday observation that children usually want to play instead of studying books. They like to have fun all the time. Therefore, this game is developed in order to enable them to learn and improve their Urdu vocabulary and strengthen their mental abilities so that they would be able to learn quickly with fun. They would not be bored at all.

This game consists of a grid containing random words. The user has to find those words from the grid. This becomes tricky for the people so they enjoy a lot as well as learn. This game is designed for people of all age groups. This is the era of android. Most of the people have smartphone in this modern world. People waste their time by playing games which are not useful for them. So, this game is a good way to utilize their time as well as strengthen their intelligence level.

Table of Contents

<i>ACKNOWLEDGMNET</i>	2
<i>Abstract</i>	3
Table of Contents	4
List of Tables	9
List of Figures	10
1. CHAPTER 1.....	11
Software Project Management Plan.....	11
1.1 INTRODUCTION	12
1.1.1 Project Overview.....	12
1.1.2 Project Deliverables	12
1.2 PROJECT ORGANIZATION	12
1.2.1 Software Process Model.....	12
1.2.2 Roles and Responsibilities	13
1.2.3 Tools and Techniques	13
1.3 PROJECT MANAGEMENT	13
1.3.1 Tasks	13
1. Identify Requirements.....	13
2. Define Use cases	14
3. Develop Analysis Model.....	15
4. Develop SRS	15
5. Analyze SRS	16
6. Develop Design.....	16
7. Develop Algorithms.....	17
8. Implementation	17
9. Testing.....	17
10. Deployment.....	18
11. Maintenance.....	18
1.3.2 Assignments	19
1.3.3 Timetable	19
19	
2. CHAPTER 2.....	21

Software Requirement Specification	21
2.1 INTRODUCTION	22
2.1.1 Purpose.....	22
2.1.2 Scope.....	22
2.1.3 Definitions.....	22
Fill the word.....	22
Grid	22
Word	22
Score	22
Timer.....	22
Hint	22
Acronyms	22
SRS	22
FWG.....	23
2.2 Product Overview	23
2.3 Overall Description.....	23
2.3.1 Product Perspective.....	23
2.4 Product Features.....	24
2.5 User Classes and Characteristics.....	24
2.5.1 Requirements and characteristics of User class User.....	24
2.6 Design and Implementation Constraints	24
2.7 Assumptions and Dependencies.....	25
2.8 External Interface Requirements.....	25
2.8.1 User Interfaces	25
2.8.2 Hardware Interfaces	25
2.8.3 Software Interfaces	25
2.8.4 Communication Interfaces	25
2.9 Software Product Features	25
2.9.1 System Features for Easy Level:.....	25
2.9.2 System Features for Normal Level:	26
2.9.3 System Features for Hard Level:.....	26
2.10 Use case Model.....	26
2.10.1 Use Cases:.....	26
Special Requirements:.....	26

Stakeholders and Interest:	26
Use case Diagram.....	27
Use Cases Description: [1].....	27
2.10.2 Use Case UC1: Start the game	27
2.10.3 Use case UC2: Read the instructions	28
2.10.4 Use case UC3: Mute/unmute the sound	28
2.10.5 Use case UC4: Select the category of dictionary	29
2.10.6 Use case UC5: Select the level	30
2.10.7 Use case UC6: Play the game	30
2.10.8 Use case UC7: Find words from the grid.....	31
2.10.9 Use case UC8: View the score	32
2.10.10 Use case UC9: Check the hint.....	32
2.10.11 Use case UC10: Exit the game.....	33
2.11 Domain Model	34
2.13 Activity Diagram	35
2.14 System Sequence Diagrams	36
SSD1: Start the game	36
SSD2: Read the instructions	36
SSD3: Select the category of dictionary	37
SSD4: Select the Level	37
SSD5: Play the game	38
SSD6: Find words	38
SSD7: Check the Score	39
SSD8: View the Score	39
2.15 System Quality Attributes	39
Security Requirements	39
Availability:	39
Usability:.....	39
Portability.....	40
Reliability.....	40
Performance	40
Maintainability	40
Database requirements	40
3. CHAPTER 3.....	41

Software Design Description.....	41
3.1 INTRODUCTION	42
3.1.1 Design Overview.....	42
3.2 Requirement Traceability Matrix.....	42
3.3 System Architectural Design	42
3.3.1 Chosen System Architecture.....	42
3.3.2 Discussion of Alternative Designs	43
3.3.3 System Interface Description	43
3.4 Detailed Description of Components	43
3.4.1 MainActivity class:	43
3.4.2 Easy.....	44
3.4.3 Normal	45
3.4.4 Hard.....	45
3.4.5 Instructions.....	46
3.4.6 Reward	46
3.5 Sequence Diagram 1	47
3.6 Sequence Diagram 2	48
3.7 Sequence Diagram 3	49
3.8 Sequence Diagram 4	50
3.9 Design Class Diagram.....	51
3.10 User Interface Design.....	51
3.10.1 Description of the User Interface	51
3.10.2 Screen Images (Rough Prototypes).....	52
3.10.3 Objects and Actions	52
4. CHAPTER 4.....	54
Implementation	54
4.1 INTRODUCTION	55
4.2 Framework Selection	55
4.3 Language Selection	55
4.4 Operating System.....	55
4.5 FWG screenshots:	56
5. CHAPTER 5.....	58
Software Test Document	58
5.1 INTRODUCTION	59

5.1.1	Purpose.....	59
5.1.2	System Overview	59
5.1.3	Test Approach.....	59
5.2	TEST PLAN.....	59
5.2.1	Features to be tested.....	59
	Start the game	59
	Read Instructions	59
	Play the game.....	59
	Manage Sound	59
	Design Grid filled with random words	59
	View the Score.....	59
	Mark words.....	59
	Exit the game	59
5.2.2	Features not to be tested.....	59
5.3	Testing Tools and Environment.....	60
5.4	TEST CASES.....	60
5.5	Requirement Traceability Matrix.....	63
5.6	APPENDIX.....	64
	References:.....	64
6.	CHAPTER 6.....	65
	Conclusions & Future Enhancements	65
6.1	Summary	66
6.2	Conclusion & Future Enhancements.....	66
6.2.1	Conclusion.....	66
6.2.2	Future Assessment.....	66

List of Tables

Table 1: Requirement Traceability Matrix.....	42
Table 2: Test Case1.....	60
Table 3: Test Case2.....	60
Table 4: Test Case 3.....	61
Table 5: Test Case 4.....	61
Table 6: Test Case 5.....	61
Table 7: Test Case 6.....	62
Table 8: Test Case 7.....	62
Table 9: Test case 8.....	62
Table 10: Test Case 9.....	63
Table 11: RTM between Test Cases and Requirements	63
Table 12: References.....	64

List of Figures

Figure 1: Project Plan 1	19
Figure 2: Project Plan 2	20
Figure 3: Product Perspective	23
Figure 4: Use case Diagram.....	27
Figure 5: Domain Model	34
Figure 6: Activity Diagram.....	35
Figure 7: SSD1.....	36
Figure 8: SSD2.....	36
Figure 9:SSD3.....	37
Figure 10: SSD4.....	37
Figure 11: SSD5.....	38
Figure 12: SSD6.....	38
Figure 13: SSD7.....	39
Figure 14: SSD8.....	39
Figure 15: Architecture Design.....	43
Figure 16: Sequence Diagram 1	47
Figure 17: Sequence Diagram 2.....	48
Figure 18: Sequence Diagram 3.....	49
Figure 19: Sequence Diagram 4.....	50
Figure 20: Design Class Diagram	51
Figure 21: Prototype 1	52
Figure 22: Prototype 2	52

1. CHAPTER 1

Software Project Management Plan

1.1 INTRODUCTION

1.1.1 Project Overview

This project is android based Fill the word (Urdu Version) game. Many word games are already available on smartphones that serve educational purpose. The problem is that many people have not developed interest to learn Urdu vocabulary. The motivation for this game is to increase Urdu vocabulary of people. This game will fulfill this purpose. This game consists of grid that contains Urdu words placed randomly on it. People will use their mental abilities or intelligence to find those words and learn their spellings quickly. They will develop interest in the people to learn new Urdu words.

The project will be completed within the defined time parameters. It is expected that after the completion of this project, it will be prove ~~every~~ beneficial for the people who want to increase their Urdu vocabulary.

1.1.2 Project Deliverables

Deliverable is a tangible output of human effort provided by a developer to a customer. These deliverables are delivered to the project supervisor. The deliverables for the project “Fill the Word Game” are as follows:

- Software Project Management Plan (SPMP)
- Software Requirement Specification (SRS)
- Software Design Document (SDD)
- Implementation Documentation
- Software Test Documentation (STD)
- Final Product

1.2 PROJECT ORGANIZATION

1.2.1 Software Process Model

I have chosen “Waterfall Process Model”. It is a sequential and non-iterative process model. We go steadily downward through the phases of conception, initiation, analysis, design, construction, production/implementation, testing and maintenance to achieve progress. These phases do not overlap. This model is chosen because this model is simple and easy to understand. Each phase has specific deliverables and a review process. These phases are completed once and there is no chance of repeating and going back to the previous phase. In this project, it is asked to complete each step within the defined time. Therefore it is chosen.

1.2.2 Roles and Responsibilities

This project involves Project Manager named Arfa Batool who is responsible for developing the Project Plan and responsible for communication, status reporting, developing the software product, and, in general, making sure the project is delivered in budget, on schedule, and within scope.

The Project supervisor Dr. Onaiza Maqbool provides support for the Project Manager; assists with major issues, problems, and policy conflicts and removes obstacles.

Stakeholders:

In this project, the user, project manager and project supervisor are the stakeholders. User is responsible to play the game.

1.2.3 Tools and Techniques

In this project “ProjectLibre” is used as a tool for software project management plan. A technique is used that divides activities and tasks into smaller ones so that they will be completed easily within the defined time. Division of larger tasks into smaller ones is similar to work breakdown structure. “Gantt Chart” is used which includes activities and tasks to be completed within the defined time parameters. Date and time is assigned for each task to be completed. Graphs are also used for making the project plan. That’s why one can easily understand visual display of the scheduled time of a task or an activity. The steps to develop project plan are listed below:

- List all activities or tasks in the plan
- For each task, show the earliest start date, estimated length of time it will take
- Plot tasks onto graph paper

1.3 PROJECT MANAGEMENT

1.3.1 Tasks

1. Identify requirements
2. Define use cases
3. Develop Analysis Model
4. Develop SRS
5. Analyze SRS
6. Develop Design
7. Develop algorithms
8. Implementation
9. Testing of software
10. Deployment
11. Maintenance

1. Identify Requirements

Description

To develop this project, the initial step is to identify the requirements of the project. The requirements include functional and non-functional requirements.

Deliverables and Milestone

Milestone	Date
Identify Requirements	4 th Oct, 2016

Resources Needed

1. People
2. Software
3. Hardware

People: Arfa Batool, supervisor

Software: MS Word

Hardware: PC

Dependencies and Constraints

-

Risk and Contingencies

-

2. Define Use cases**Description**

After identifying requirements, next task is to write use cases and draw use case diagram. Use cases are the functions that user has to perform on a system. The system then responds accordingly.

Deliverables and Milestones

Tasks	Date
Write use cases	10/14/16
Draw use case diagram	10/18/16
Review requirements and use cases	10/19/16

Resources needed

People: Project manager, supervisor

Software: Argo UML, MS Word

Hardware: PC

Dependencies and constraints

-

Risks and Contingencies

None

3. Develop Analysis Model

Description

Analysis model includes Domain Model and system sequence diagrams. These diagrams help understand the system in detail

Deliverables and Milestones

Tasks	Date
Develop Domain Model	10/18/16
Develop System Sequence Diagram	10//18/16
Review diagrams	10/19/16

Resources needed

People: Project manager, supervisor

Software: MS Word, Argo UML

Hardware: PC

Dependencies and constraints

-

Risks and Contingencies

None

4. Develop SRS

Description

System Requirement Specification includes the functional and non-functional requirements

Deliverables and Milestones

Tasks	Date
Define Functional and Non-Functional Requirements	10/14/16
Review Requirements	10/18/16
Finalize SRS	10/18/16

Resources needed

People: Project manager, supervisor

Software: MS Word

Hardware: PC

Dependencies and constraints

-

Risks and Contingencies

None

5. Analyze SRS**Description**

System Requirement Specification includes the functional and non-functional requirements

Deliverables and Milestones

Tasks	Date
Meet Stakeholders	10/28/16

Resources needed

People: Project manager, supervisor

Software: MS Word

Hardware: PC

Dependencies and constraints

-

Risks and Contingencies

None

6. Develop Design**Description**

In this step, algorithms will be written for different functions that system has to perform.

Deliverables and Milestones

Tasks	Date
Develop Architecture design	11/3/16
Develop Sequence Diagrams	11/9/16
Develop Design Class Diagram	11/9/16

Resources needed

People: Project manager, supervisor

Software: ArgoUML

Hardware: PC

Dependencies and constraints

-

Risks and Contingencies

None

7. Develop Algorithms

Description

The algorithms will be written for different functions that system has to perform.

Deliverables and Milestones

Tasks	Date
Write pseudo code	12/15/16

Resources needed

People: Project manager, supervisor

Software: MS Word

Hardware: PC

Dependencies and constraints

-

Risks and Contingencies

None

8. Implementation

Description

This is a very important phase in the project development.

Deliverables and Milestones

Tasks	Date
Start coding	5/26/17
Validate code	6/12/17

Resources needed

People: Project manager, supervisor

Software: Android Studio

Hardware: PC

Dependencies and constraints

-

Risks and Contingencies

None

9. Testing

Description

The product is tested in this stage.

Deliverables and Milestones

Software Testing

06/12/17

Resources needed

People: Project manager, supervisor

Software: Android Studio, android smart phone

Hardware: PC

Dependencies and constraints

-

Risks and Contingencies

None

10. Deployment

The product is deployed in this stage

Deliverables and Milestones

Tasks

Date

Install

07/08/17

Update

07/11/17

Resources needed

People: Project manager, supervisor

Software: Android Studio, android smart phone

Hardware: PC

Dependencies and constraints

-

Risks and Contingencies

None

11. Maintenance**Description**

The code is updated and maintained at this step.

Deliverables and Milestones

Maintenance

07/12/17

Resources needed

People: Project manager, supervisor

Software: Android Studio, android smart phone

Hardware: PC

Dependencies and constraints

-

Risks and Contingencies

None

1.3.2 Assignments

- Software Requirement Specification
- Software Project Management Plan
- Software Design Description
- Software Test Documentation
- Final Product

1.3.3 Timetable Analysis and Design:

	🕒	Name	Duration	Start	Finish	Predecessors	Resource Names
1	👤	Analysis of case-study	22 days	10/4/16 8:00 AM	11/2/16 5:00 AM		People;Supervisor;Software;ms word;Hardware;pc;Arfa Batool
2	👤	Identify requirements	22 days	10/4/16 8:00 AM	11/2/16 5:00 AM		
3	👤	review case-study	5 days	10/5/16 8:00 AM	10/11/16 5:00 AM		Arfa Batool;;Supervisor
4	👤	meet stakeholder	2 days	10/6/16 8:00 AM	10/7/16 5:00 AM		Hardware;pc
5	👤	define usecase	11 days	10/4/16 8:00 AM	10/18/16 5:00 AM		
6	👤	write usecase	5 days	10/7/16 8:00 AM	10/13/16 5:00 AM		Software;ms word;pc;Hardware
7	👤	Draw usecase diagram	3 days	10/14/16 8:00 AM	10/18/16 5:00 AM	6	Hardware;pc;Software;usecase tool (ArgoUML)
8	👤	review requirement and use case	2 days	10/4/16 8:00 AM	10/5/16 5:00 AM		People;Arfa Batool;
9	👤	Develop analysis Model	4 days	10/18/16 8:00 AM	10/21/16 5:00 AM		People;Arfa Batool;
10	👤	develop domain model	4 days	10/18/16 8:00 AM	10/21/16 5:00 AM		
11	👤	develop system sequence diagram	3 days	10/18/16 8:00 AM	10/20/16 5:00 AM		
12	👤	Develop SRS	8 days	10/17/16 8:00 AM	10/26/16 5:00 AM		People;Arfa Batool;
13	👤	Identify functional and non-functional requirer	7 days	10/18/16 8:00 AM	10/26/16 5:00 AM		
14	👤	refine SRS	4 days	10/17/16 8:00 AM	10/20/16 5:00 AM		
15	👤	Analyze SRS	4 days	10/28/16 8:00 AM	11/2/16 5:00 AM		People;Arfa Batool;
16	👤	Meet stakeholder	4 days	10/28/16 8:00 AM	11/2/16 5:00 AM		Hardware;pc
17	👤	Analysis done	1 day	11/2/16 8:00 AM	11/2/16 5:00 AM		
18	👤	Design software	33 days	11/3/16 8:00 AM	12/19/16 5:00 AM		
19	👤	Develop design	9 days	11/3/16 8:00 AM	11/15/16 5:00 AM		People;Arfa Batool;;Software;ms word
20	👤	Develop architectural design	3 days	11/3/16 8:00 AM	11/7/16 5:00 AM		
21	👤	Develop Sequence Diagrams	3 days	11/9/16 8:00 AM	11/11/16 5:00 AM		
22	👤	Develop Design Class Diagrams	3 days	11/11/16 8:00 AM	11/15/16 5:00 AM		
23	👤	Develop Algorithms	5 days	12/13/16 8:00 AM	12/19/16 5:00 AM		
24	👤	draw flowchart	3 days	12/13/16 8:00 AM	12/15/16 5:00 AM		
25	👤	write pseudo code	3 days	12/15/16 8:00 AM	12/19/16 5:00 AM		Software;programming tool(Android Studio);Hardware;pc

Figure 1: Project Plan 1

Implementation:

		Name	Duration	Start	Finish	Predecessors	Resource Names
25		write pseudo code	3 days	12/15/16 8:00 AM	12/19/16 5:0...		Software;programming tool(Android Studio);Hardware;pc
26		Implementation	73 days	3/1/17 8:00 AM	6/9/17 5:00...		
27		preparation of coding	72 days	3/1/17 8:00 AM	6/8/17 5:00...		
28		select language	5 days	3/1/17 8:00 AM	3/7/17 5:00 PM		People;Arfa Batool;
29		create a set of unit test	20 days	3/7/17 8:00 AM	4/3/17 5:00 PM		
30		start coding	40 days	4/3/17 8:00 AM	5/26/17 5:0...		
31		write code	40 days	4/3/17 8:00 AM	5/26/17 5:00 ...		Software;programming tool(Android Studio
32		validate code	10 days	5/26/17 8:00 AM	6/8/17 5:00...		People;Arfa Batool;
33		review code	10 days	5/26/17 8:00 AM	6/8/17 5:00 PM		People;Arfa Batool;
34		conduct a unit test	5 days	5/29/17 8:00 AM	6/2/17 5:00 PM		
35		coding completed	2 days	6/8/17 8:00 AM	6/9/17 5:00 PM		
36		Testing of software	4 days	6/12/17 8:00 AM	6/15/17 5:00 ...	26	People;Arfa Batool;
37		Deployment	7 days?	7/1/17 8:00 AM	7/11/17 5:0...	36	Arfa Batool;People
38		Release	1 day?	7/1/17 8:00 AM	7/3/17 5:00 PM		
39		Install	1 day?	7/8/17 8:00 AM	7/10/17 5:00 ...		
40		Update	1 day?	7/11/17 8:00 AM	7/11/17 5:00 ...		
41		Maintenance	1 day?	7/12/17 8:00 AM	7/12/17 5:00 ...		

Figure 2: Project Plan 2

2. CHAPTER 2

Software Requirement Specification

2.1 INTRODUCTION

2.1.1 Purpose

This document describes all the requirements for Fill the word (Urdu Version) game. It gives the complete description of the functions that the system and the user have to perform. It includes all the functional as well as non-functional requirements. It will help the user to determine whether this product meets his/her needs or how the software must be modified according to the user needs. This document is intended for developers, project managers, testers, and documentation writers.

2.1.2 Scope

This document specifies the requirements for Fill the Word (Urdu Version) game software. This game will help the children and adults to increase their Urdu vocabulary. They will learn new words as well as the spellings of those words quickly.

2.1.3 Definitions

Fill the word

This is the game that is to be developed. It comprises of a grid containing random words. User has to find the words from the grid. In this way, it will help the people to learn new Urdu words.

Grid

Grid is a board on which the game will be played by the user. A grid consists of rows and columns whose size will be different for different levels. It will contain random words placed on it randomly.

Word

A word consists of Urdu characters.

Score

Score is the status of user's performance.

Timer

It is the time taken by the user to complete level of the game.

Hint

It is the option that the user may want to use when he/she finds any difficulty in finding words from the grid.

Acronyms

SRS

System Requirement Specification

FWG

Fill the Word Game

2.2 Product Overview

Today, many word games are available on smartphone. These are educational games that help a lot in learning vocabulary of a language. These games include word search, word puzzles and fill the word games. People want to get an easy way to learn spellings of difficult words of respective language quickly. They want to sharpen their language skills. Therefore, word games fulfill their purpose. Especially children enjoy playing these games. It is very beneficial to play such word games because these games do not waste our time rather give us enjoyment and serve their educational purpose as well.

This document is fully organized, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type. Rest of the chapter contains Section 2 - Section 5. Section 2 presents an overall description. Section 3 presents requirements for user interface, hardware interface, software requirements. Section 4 includes system features. Section 5 includes the explicit specification of non-functional requirements.

2.3 Overall Description

2.3.1 Product Perspective

This is a new self-contained product. This game has a novelty factor. It's not been built yet in Urdu version. This game consists of two interfaces; one for the game menu and the other for playing the game i.e. grid interface. File storage stores dictionary in it. The system calculates scores and time. This system stores Urdu dictionary and searches or matches words from it whenever the user finds a word.

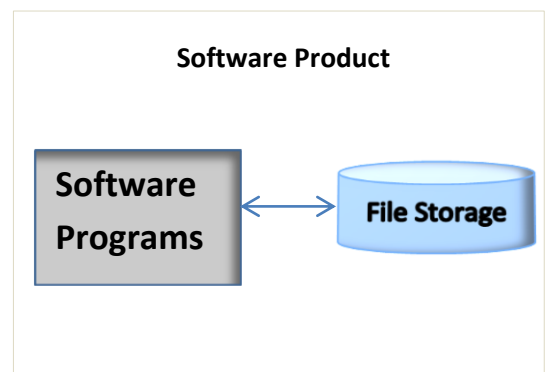


Figure 3: Product Perspective

2.4 Product Features

Summary of major functions that the system has to perform are listed below:

- **Load the game**
This function will load the game when user will start the game
- **Design the grid**
This function will design the grid of specified length and width for the respective level.
- **Select random words:**
This function will select random words from the dictionary
- **Place words on grid:**
This function will place the random words on the grid randomly
- **Store the dictionary:**
This function will store the Urdu dictionary in data structure
- **Search words:**
This function will check words from the dictionary indicated by the user
- **Count time:**
This function will count the time taken for finding all words from the grid
- **Mark words:**
This function will mark the words found by the user

2.5 User Classes and Characteristics

For FWG, there is only one user class that is the user or player who plays the game. No other user class is involved in this game because only the user/player is interacting with the game.

2.5.1 Requirements and characteristics of User class User

- User must have introductory knowledge about Urdu language
- He must know about Urdu Characters
- He must have technical expertise to use smartphone and play game

2.6 Design and Implementation Constraints

The constraints are mentioned below:

- This is Urdu version game in which only Urdu language is supported
- This game will be designed only for android operating system
- This game will open only the default level for all the users. The user who has completed level 1 next time the default level will be level 2 and so on.

2.7 Assumptions and Dependencies

- It is assumed that the game will be single player only
- It is assumed that no internet access will be provided for playing game.

2.8 External Interface Requirements

2.8.1 User Interfaces

This game has two user interfaces. At runtime, the system ensures the best possible screen resolution.

- Fits on small and large screens (so users can actually use your application)
- Is optimized for both landscape and portrait orientations

The first screen after launching the game is the main menu. User has multiple options here to play the game to mute/unmute sound effects, to view instructions, to exit the game. This is a space where humans and machine interact with each other.

2.8.2 Hardware Interfaces

Hardware interfaces must include the smartphone or tablet having touch screen UI. As this game is android based therefore all the android devices must meet memory requirements.

2.8.3 Software Interfaces

FWG will be designed in android studio. This platform includes Java and XML languages that will be used to design the game.

2.8.4 Communication Interfaces

In the assumptions section, it is assumed that at the initial level FWG will not include the features that require internet access such as comparing your score with other users or playing the game with other players etc.

2.9 Software Product Features

1. The system shall load the menu of game
2. The system shall design grid
3. The system shall generate random words from the dictionary
4. The system shall place words on the grid randomly
5. The system shall search the words from the dictionary found by the user
6. The system shall mark the words
7. The system shall calculate score and count time taken to find words.

2.9.1 System Features for Easy Level:

1. The system shall design a grid of 3 x 3
2. The system shall not select words of length greater than 9 from the dictionary

3. If the user finds other words from the letter of words placed by system, then system shall display a message “word is true but find the word selected by system”
4. System shall mark only those words that are selected by system

2.9.2 System Features for Normal Level:

1. Design a grid of 4 x 4
2. If the system selects words of length 2 to 16 then system shall place the words on the grid randomly
3. the system shall not select words of length greater than 16
4. If the user finds other words from the letter of words placed by system, then system shall display a message “word is true but find the word selected by system”
5. system shall mark only those words that are selected by system

2.9.3 System Features for Hard Level:

1. Design a grid of 6 x 6
2. The system shall not select words of length greater than 36
3. if the user finds other words from the letter of words placed by system, then system shall display a message “word is true but find the word selected by system”
4. system shall mark only those words that are selected by system

2.10 Use case Model

2.10.1 Use Cases:

1. Start the game
2. Read the instructions
3. Mute/unmute the sound
4. Select the category of dictionary
5. Select the level
6. Play the game
7. Find words from the grid
8. View the score
9. Check the hints
10. Exit the game

Special Requirements:

1. Android based smartphone
2. Touch Screen user interface
3. Text must be visible to the user

Stakeholders and Interest:

User: User wants to learn spellings of Urdu words and increase his/her vocabulary. He wants to play the game. He wants error free and interesting game that fulfills his demand.

Use case Diagram



Figure 4: Use case Diagram

Use Cases Description: [1]

2.10.2 Use Case UC1: Start the game

Primary Actor:

User

Pre-conditions:

-

Post-conditions:

User has started the game. System has displayed the menu of game to the user.

Main Success Scenarios:

1. User starts the game
2. System displays menu/interface of the game that includes play option, sound, read instructions and exit option.
3. User sees game interface displayed by the system

Alternative flows:

- a. At any time the system fails to respond

User restarts the game. The system recovers the previous state where the system failed to respond.

1. System fails to display menu on the screen
System displays a message to restart the game

Frequency:

Any time the user wants to start the game

2.10.3 Use case UC2: Read the instructions**Primary Actor:**

User

Pre-conditions:

1. User has started the game
2. User has seen the menu of game

Post-conditions:

User has clicked “read instructions” option. He has read the instructions to play the game.

Main Success Scenarios:

1. User clicks read instructions option
2. System displays instructions on the screen.
3. User reads the instructions

Alternative flows:

1. User does not want to read instructions. He clicks any other option

System responds the user depending upon which option he selects.

2. User clicks back option and does not read instructions

System displays menu of the game

Frequency:

Whenever the user wants to read instructions of the game

2.10.4 Use case UC3: Mute/unmute the sound**Primary Actor:**

User

Pre-conditions:

1. User has started the game
2. System has displayed menu of the game

Post-conditions:

System has muted/unmuted the sound

Main Success Scenarios:

1. User clicks sound option
2. System allows user to mute/unmute the sound

Alternative flows:

1. User clicks any other option
System responds the user according to the option selected by user

Frequency:

Whenever the user wants to mute/unmute the sound of the game

2.10.5 Use case UC4: Select the category of dictionary**Primary Actor:**

User

Pre-conditions:

User has started the game

Post-conditions:

User has selected the category of dictionary.

Main Success Scenarios:

1. User clicks category option
2. System displays many categories of dictionary.
3. User selects one of the category of dictionary

Alternative flows:

2. System fails to display the categories of dictionary
 - 2.1 System displays a message to the user to restart the game
 - 2.2 User re-clicks category option
 - 2.3 System recovers from errors and displays the categories of dictionary

Frequency:

Whenever the user wants to play the game

2.10.6 Use case UC5: Select the level

Primary Actor:

User

Pre-conditions:

User has started the game

Post-conditions:

User has selected the level of his/her choice

Main Success Scenarios:

1. User clicks Level option
2. System displays the type of levels i.e. easy, normal and hard
3. User selects one type of level
4. System saves the selected type of level

Alternative flows:

1. System fails to display the type of levels
 - 2.1 System displays a message to the user to restart the game
 - 2.2 User re-clicks category option
 - 2.3 System recovers from errors and displays the type of levels

Frequency:

Whenever the user wants to play the game

2.10.7 Use case UC6: Play the game

Primary Actor:

User

Pre-conditions:

User has started the game

Post-conditions:

System has displayed the grid containing random words on the screen

Main Success Scenarios:

1. User selects play option
2. System displays a grid containing random words on the screen

3. User starts playing the game
4. System starts counting time

Alternative flows:

1. System fails to generate grid of random words
 - 1.1 System displays a message to the user to go back and re-click the play option
 - 1.2 User re-clicks play option
 - 1.3 System recovers from errors and repeats steps 2-5

Frequency:

Whenever the user wants to play the game

2.10.8 Use case UC7: Find words from the grid

Primary Actor:

User

Pre-conditions:

1. User has clicked play option (UC6)
2. System has displayed grid containing random words

Post-conditions:

User has found the words from the grid

Main Success Scenarios:

1. User finds a words from the grid
2. System searches the word from the dictionary
3. System marks the word
4. System repeats step 1-3
5. System stops counting time when user finds all words from the grid

Alternative flows:

1. User faces difficulty in finding word
 - 1.1 User clicks the hint option or tries to find a word on his own
 - 1.2 User finds the word
2. System finds no such word from dictionary found by the user
System displays a message “word is incorrect”

Frequency:

Whenever the user wants to find words from the grid

2.10.9 Use case UC8: View the score

Primary Actor:

User

Pre-conditions:

1. User has clicked play option
2. System has displayed grid containing random words

Post-conditions:

User has viewed the score of game

Main Success Scenarios:

1. User finds words from the grid
2. System starts calculating score
3. System displays score when user clicks score option
4. User views the score

Alternative flows:

- a. At any time system fails

User restarts the game

System recovers the previous state where the system failed to respond

Frequency:

Whenever the user wants to play the game

2.10.10 Use case UC9: Check the hint

Primary Actor:

User

Pre-conditions:

1. User has clicked play option
2. System has displayed grid containing random words

Post-conditions:

User has viewed the hints of the game

Main Success Scenarios:

1. User selects Hint option
2. System allows user to see different hints of the words placed on the grid
3. User has viewed the hints of game

Alternative flows:

1. If the user does not need hints, he/she will not click the hint option
2. He will find the words himself

Frequency:

Whenever the user wants to check the hints

2.10.11 Use case UC10: Exit the game**Primary Actor:**

User

Pre-conditions:

1. User has started the game
2. System has displayed the menu of game

Post-conditions:

User has exited the game

Main Success Scenarios:

1. User selects exit option
2. System allows user to exit the game

Alternative flows:

-

Frequency:

Whenever the user wants to exit the game

2.11 Domain Model

Domain model describes the real world entities and the relationships between them.

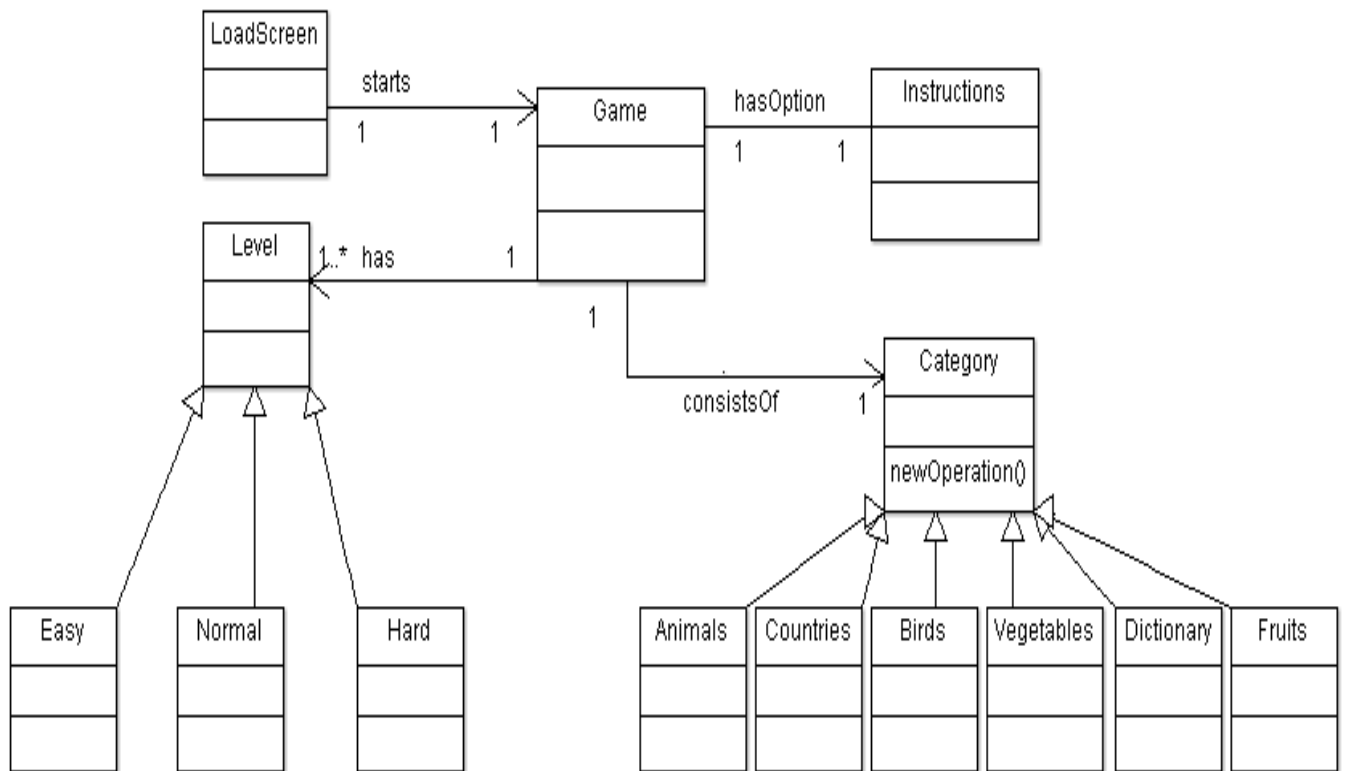


Figure 5: Domain Model

2.13 Activity Diagram

An activity diagram is used to represent a series of actions or flow of control in a system similar to a flowchart or a diagram. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent. [6]

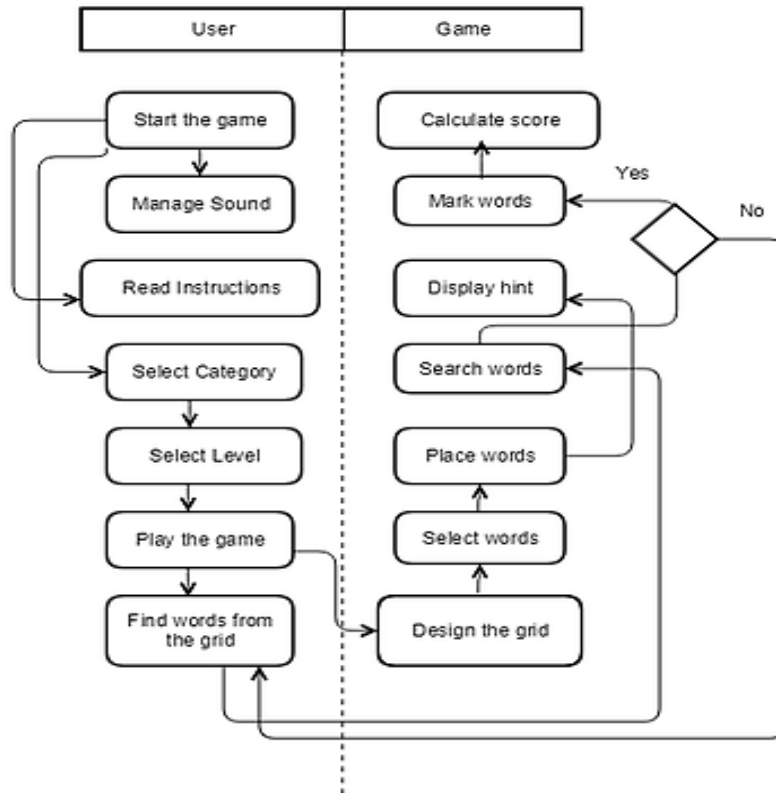


Figure 6: Activity Diagram

2.14 System Sequence Diagrams

A system sequence diagram (SSD) is a sequence diagram that shows, for a particular scenario of a use case, the events that external actors generate their order, and possible inter-system events. [5]

SSD1: Start the game

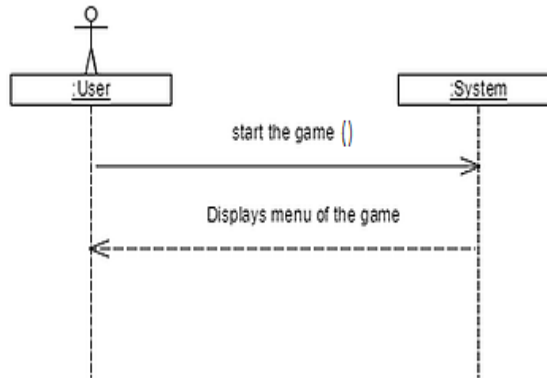


Figure 7: SSD1

SSD2: Read the instructions

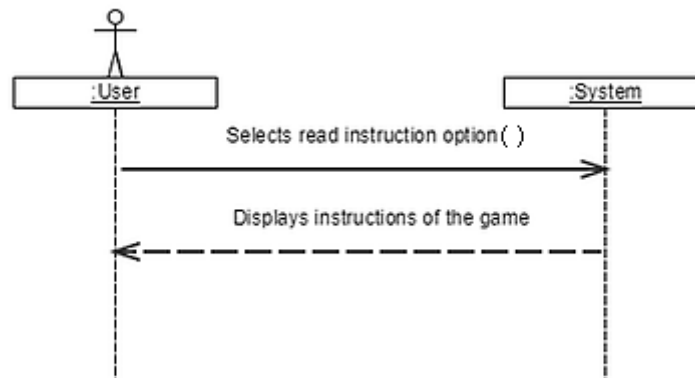


Figure 8: SSD2

SSD3: Select the category of dictionary

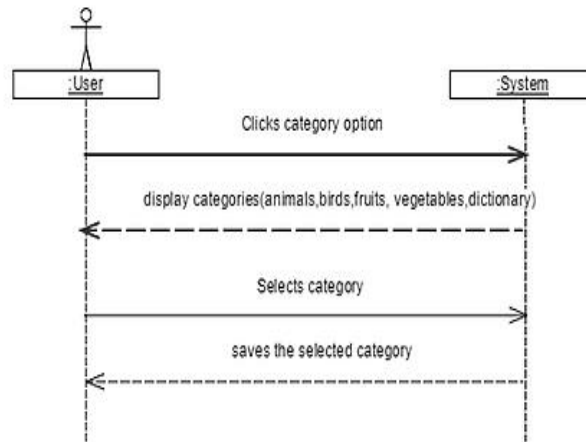


Figure 9:SSD3

SSD4: Select the Level

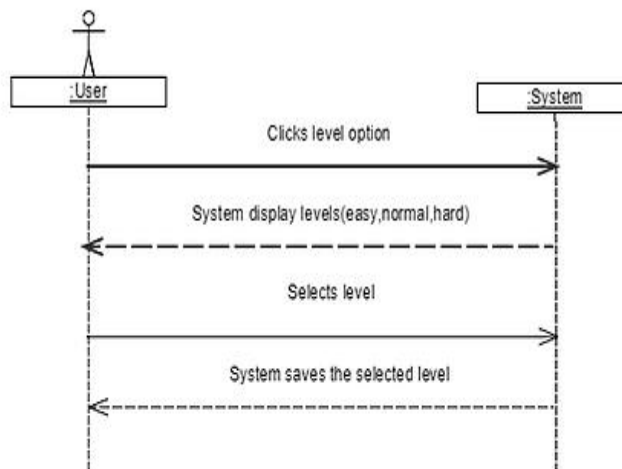


Figure 10: SSD4

SSD5: Play the game

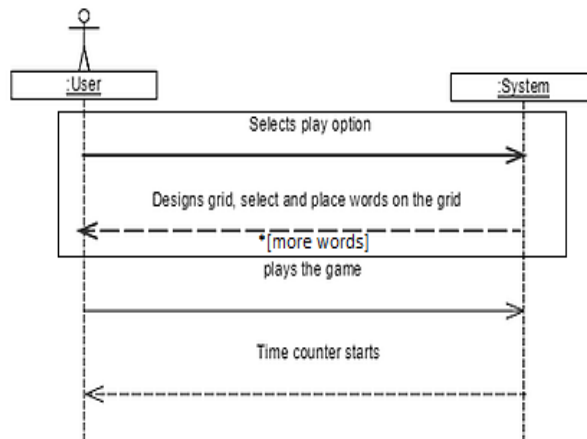


Figure 11: SSD5

SSD6: Find words

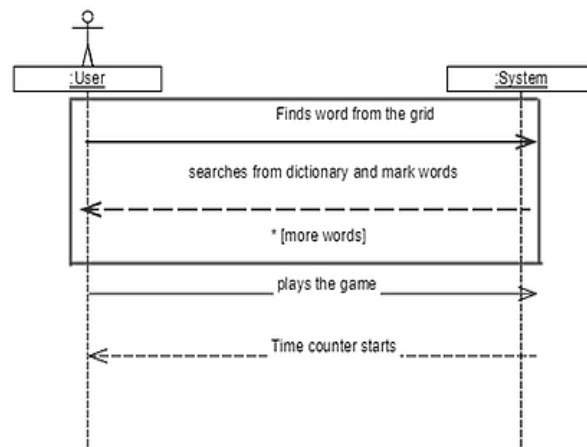


Figure 12: SSD6

SSD7: Check the Score

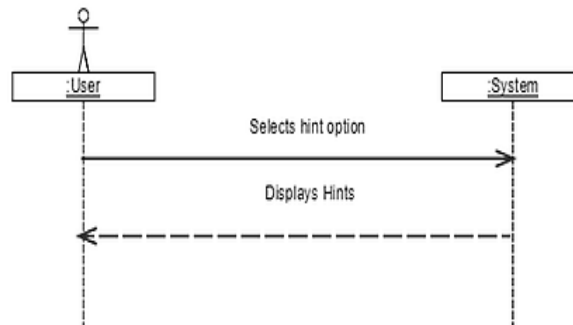


Figure 13: SSD7

SSD8: View the Score

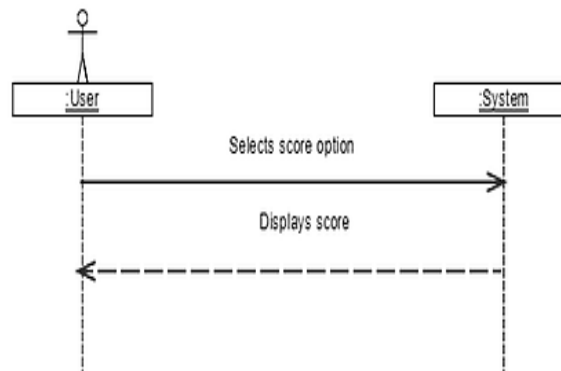


Figure 14: SSD8

2.15 System Quality Attributes

Security Requirements

1. The program will not access any user data files or programs.
2. The program will not alter or replace any system files.

Availability:

The game will be available on the smartphone 24/7 unless the user uninstalls.

Usability:

- 1 A new user should have introductory knowledge about Urdu language so that he would be able to play the game

- 2 A user who is familiar with the rules of this game will be able to correctly operate the program without any written documentation.

Portability

The program will run on android system. It should be portable to any other android to android system.

Reliability

As the program involves no user data, reliability is of low importance. Only one thing is important that the system should not crash.

Performance

- **Loading time:** the time for loading the menu of the game will be less than 5 seconds.
- 95% of the functions shall be performed by the system in less than 5 seconds
- **Executable size:** the size of executable file will be less than 100 MB

Maintainability

- The program will use modular approach so that it would be easier to update or change code when needed
- Updates in the software will be done in a separate file

Database requirements

File storage is needed that will store the Urdu dictionary.



3. CHAPTER 3

Software Design Description

3.1 INTRODUCTION

3.1.1 Design Overview

This document describes a design model with architectural, interface, component level representations.

This chapter is divided into subsections to facilitate the reader. Those are: Section 2 contains requirement traceability matrix. Section 3 contains the system architectural design. Section 4 includes the detailed description of components Section 5 contains the user interface design and some screenshots. This document also includes class diagram and sequence diagrams of components. [1]

3.2 Requirement Traceability Matrix

Requirements	Domain Model	System Sequence Diagram	Sequence Diagram	Class Diagram
Use Cases				
UC1	Fig 6	Fig 7	Fig 17	Fig 21
UC2	Fig 6	Fig 8	Fig 17	Fig 21
UC3	Fig 6	-	Fig 17	Fig 21
UC4	Fig 6	Fig 9	Fig 18	Fig 21
UC5	Fig 6	Fig 10	Fig 18, 19, 20	Fig 21
UC6	Fig 6	Fig 11	Fig 18, 19, 20	Fig 21
UC7	Fig 6	Fig 12	Fig 18, 19, 20	Fig 21
UC8	Fig 6	Fig 13	Fig 18, 19, 20	Fig 21
UC9	Fig 6	Fig 14	Fig 18, 19, 20	Fig 21
UC10	Fig 6	-	-	Fig 21

Table 1: Requirement Traceability Matrix

3.3 System Architectural Design

3.3.1 Chosen System Architecture

Logical architecture diagram is divided into layers. Basically it has three layers i.e. presentation, domain and database layers. But this game has 2- tier architecture because FWG has no database class [1]. It will store the words and retrieve from a file. The diagram is shown here.

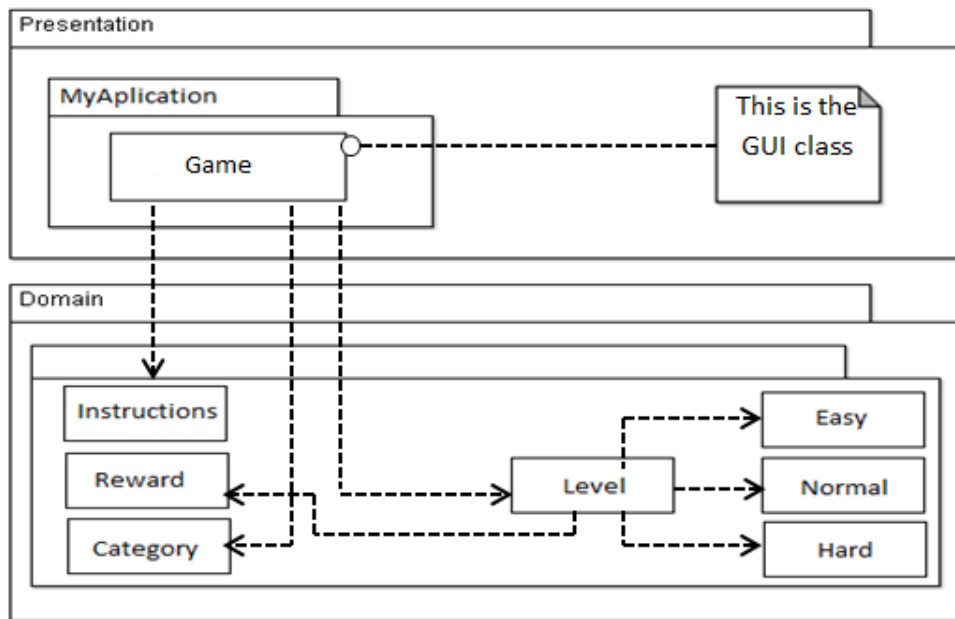


Figure 15: Architecture Design

3.3.2 Discussion of Alternative Designs

At the early stages, I thought to choose a 3-tier architecture design because I was considering my file storage in the database. I thought to put the words in the database table but later on, I removed database from architecture diagram. Because i thought that reading data from the file takes less memory and there will be no need of database connectivity, that's why I preferred 2-tier architecture. I preferred object oriented approach because in Object Oriented Programming, Programs are divided into small objects and supports **inheritance, encapsulation, abstraction, polymorphism**, etc. OOP methods make code more maintainable. It becomes easier to identify the source of errors. As far as user interface is concerned, initially I did not include score option in the game but later I included this option because it tells the user his/her performance in the game.

3.3.3 System Interface Description

System Interface consists of user interface, communication interface, hardware interface as well as software interfaces. In this game, there is no connection with internet therefore this game has not any communication interface. User interface is described in detail with screen images in section 4. Hardware interfaces must include the smartphone or tablet having touch screen UI. FWG will be designed in android studio. These platforms include Java language that will be used to design the game.

3.4 Detailed Description of Components

FWG consists of so many classes. Here is a detailed description of each component of FWG. Each class has its own methods.

3.4.1 MainActivity class:

FWG consists of this class. It is the base class of game. It is the GUI class in which the user can click the play button to play the game, can mute/unmute sound of game. He can exit the game, he can read instructions; can select the level type of his choice (i.e. easy, normal and hard). He can also select the category of dictionary.

Functions

onClickPlayListener()

This method will take the user to the class in which he can play game.

onClickHelpListener()

This method will take the user to the ReadInstructions class. So that he can read instructions to play the game

onClickSoundListener()

This method will allow the user to mute/unmute the sound of game

onClickExitListener()

This method will allow the user to exit the game

3.4.2 Easy

If the user selects level type easy from the MainActivity class then the control will go to that class and user will be able to play that level. Once the grid is designed, system will place the words randomly on the grid. Each cell of the grid will be filled with one letter of the word. The user will find the words from the grid. The system will search and mark the words. System will calculate score and take the user to the reward class on completion of level.

Attributes

cols: This is an attribute of data type integer. It has number of columns in the grid.

rows: This is an attribute of data type integer. It has number of rows in the grid.

size: this is the attribute of data type integer. It will multiply rows and columns to get the size of grid.

Word: this is the attribute of data type String. This takes a word from the text file which is to be placed on the grid.

Functions:

selectWord(): this method will select the word from Urdu vocabulary file.

searchWord(): this method is for the system which will search the word found by the user from the file whether it exists there or not. If exists, it will mark the words and take the user to reward screen upon completion of level

placeWord(): this method will place the words on the grid.

MakeEmpty(): this will make the grid empty

calculateScore(): this method will calculate the score.

getHint(): this method will provide hint to the user

3.4.3 Normal

If the user selects level type normal from the MainActivity class then the control will go to that class and user will be able to play that level. Once the grid is designed, system will place the words randomly on the grid. Each cell of the grid will be filled with one letter of the word. The user will find the words from the grid. The system will search and mark the words. System will calculate score and take the user to the reward class on completion of level.

Attributes

cols: This is an attribute of data type integer. It has number of columns in the grid.

rows: This is an attribute of data type integer. It has number of rows in the grid.

size: this is the attribute of data type integer. It will multiply rows and columns to get the size of grid.

Word: this is the attribute of data type String. This takes a word from the text file which is to be placed on the grid.

Functions:

selectWord(): this method will select the word from Urdu vocabulary file.

searchWord(): this method is for the system which will search the word found by the user from the file whether it exists there or not. If exists, it will mark the words and take the user to reward screen upon completion of level

placeWord(): this method will place the words on the grid.

MakeEmpty(): this will make the grid empty

calculateScore(): this method will calculate the score.

getHint(): this method will provide hint to the user

3.4.4 Hard

If the user selects level type hard from the MainActivity class then the control will go to that class and user will be able to play that level. Once the grid is designed, system will place the words randomly on the grid. Each cell of the grid will be filled with one letter of the word. The user will find the words from the grid. The system will search and mark the words. System will calculate score and take the user to the reward class on completion of level.

Attributes

cols: This is an attribute of data type integer. It has number of columns in the grid.

rows: This is an attribute of data type integer. It has number of rows in the grid.

size: this is the attribute of data type integer. It will multiply rows and columns to get the size of grid.

Word: this is the attribute of data type String. This takes a word from the text file which is to be placed on the grid.

Functions:

selectWord(): this method will select the word from Urdu vocabulary file.

searchWord(): this method is for the system which will search the word found by the user from the file whether it exists there or not. If exists, it will mark the words and take the user to reward screen upon completion of level

placeWord(): this method will place the words on the grid.

MakeEmpty(): this will make the grid empty

calculateScore(): this method will calculate the score.

getHint(): this method will provide hint to the user

3.4.5 Instructions

This class consists of game rules. It will help the user to read instructions for playing the game.

Function:

onCreate (): this function will be used to provide instructions of the game. All the functionality for reading the instructions of the game will be implemented in this function.

nextSuggestion(): this method will take the user to Suggestion class and that class will display how to play the game. It shows a complete guide to play the level.

3.4.6 Reward

This class displays the reward for completing the level. If the user has completed the level in zero moves, then three stars are given to him as a reward. If he has completed the level in one move then two stars are given. Similarly, if he has completed the level with two or greater than two moves, then he will get one star.

Functions:

nextLevel(): this function will open the next level when the user has finished the previous level and got reward.

3.5 Sequence Diagram 1

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. [1]

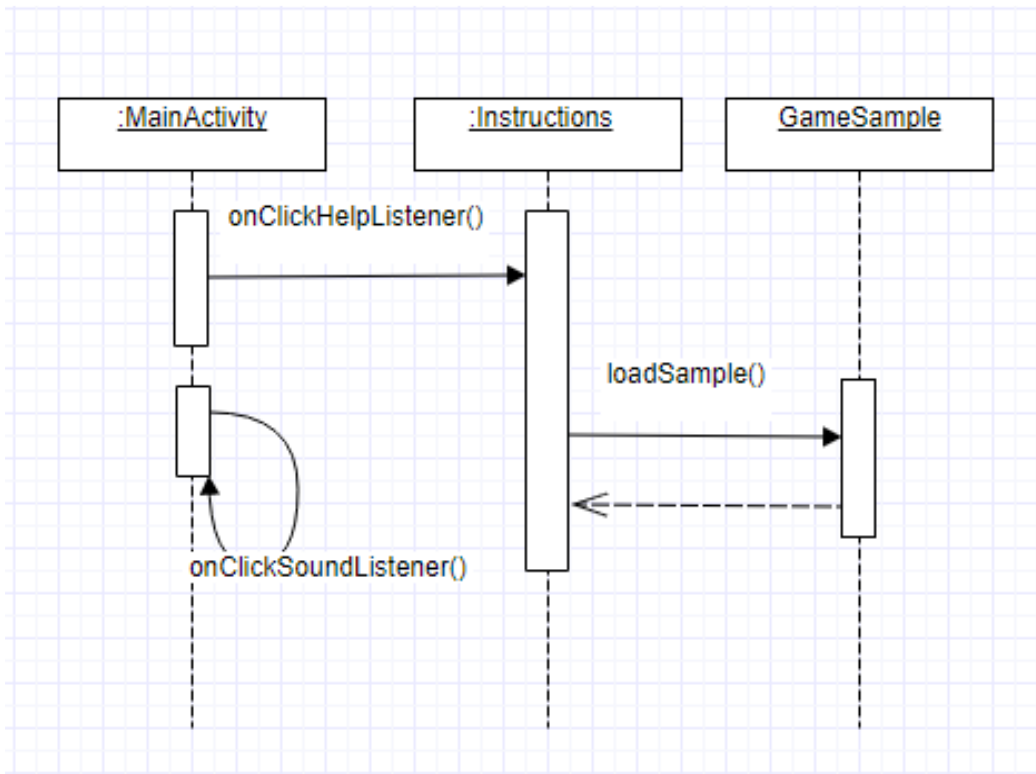


Figure 16: Sequence Diagram 1

3.6 Sequence Diagram 2

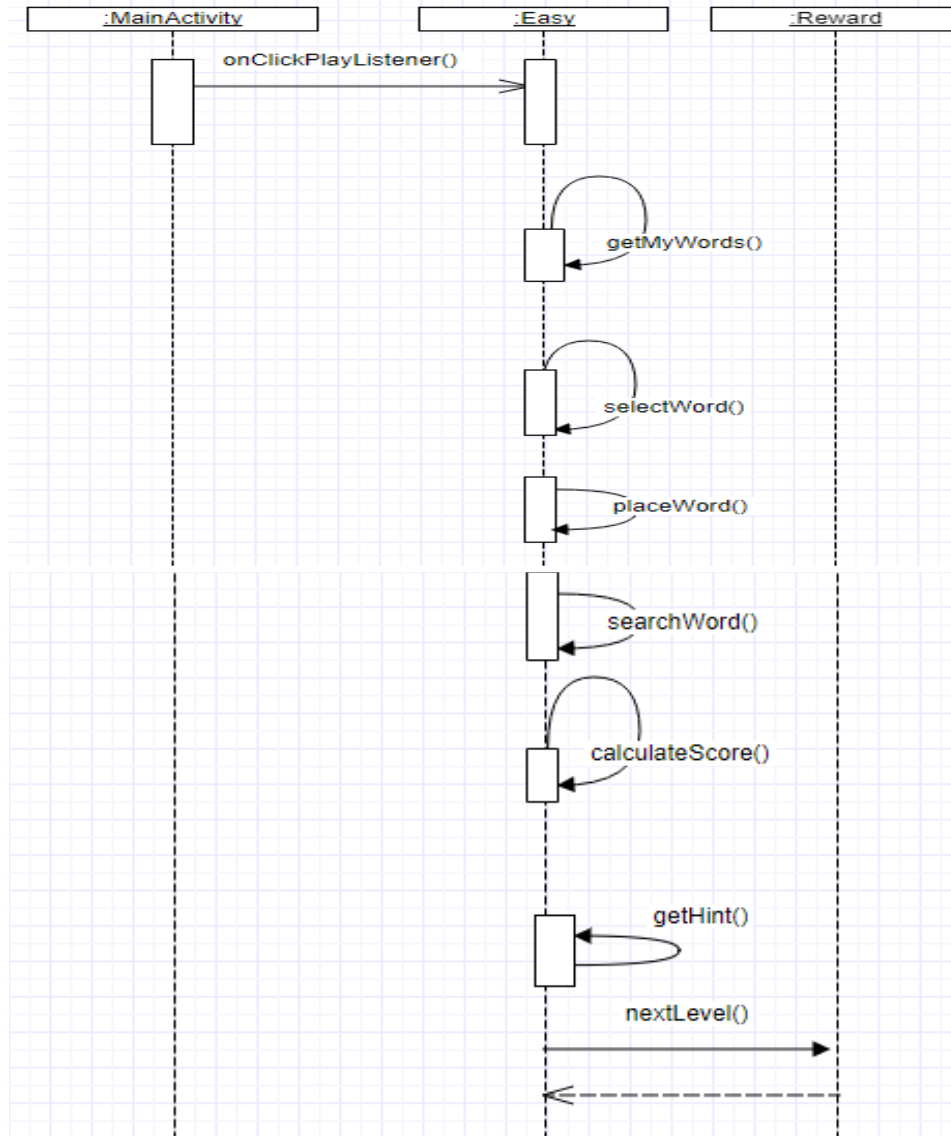


Figure 17: Sequence Diagram 2

3.7 Sequence Diagram 3

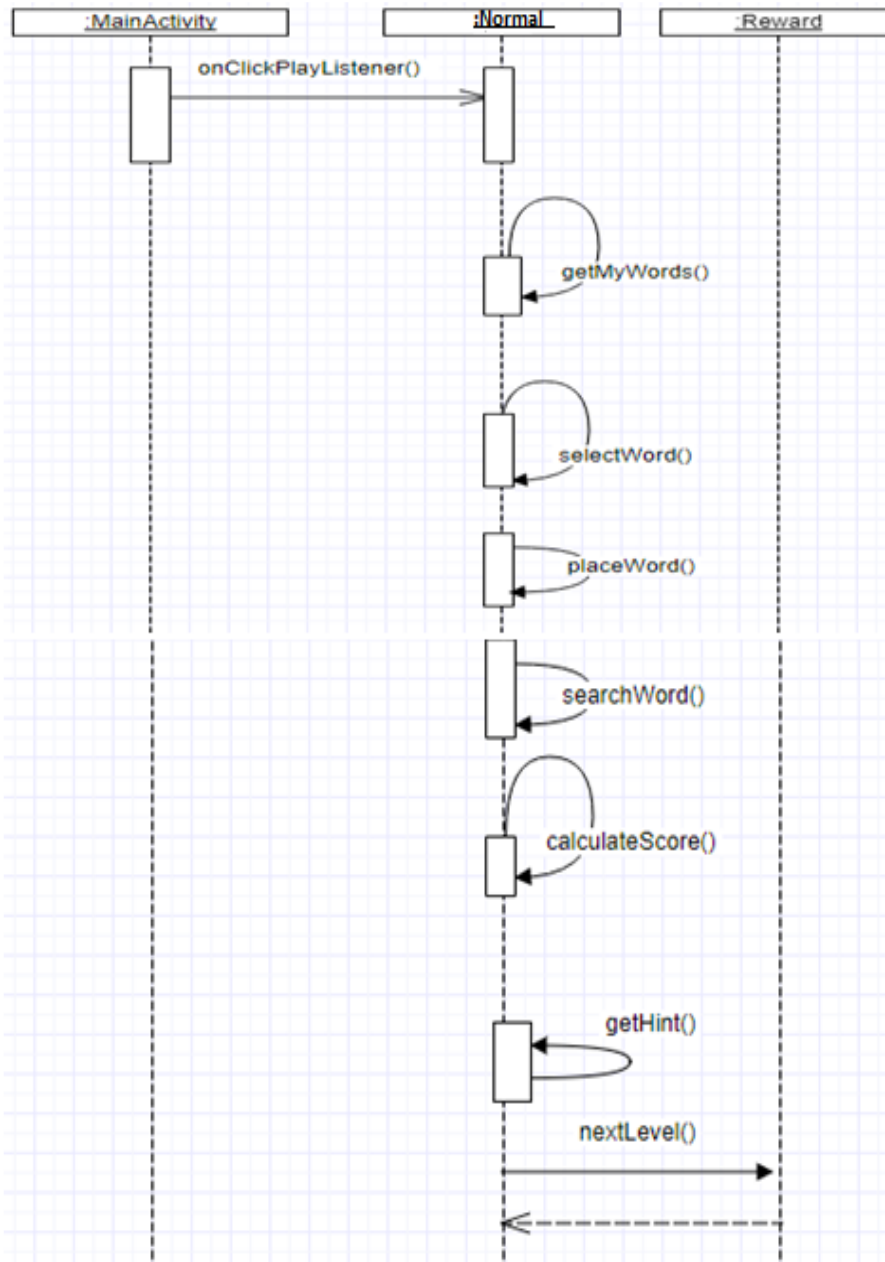


Figure 18: Sequence Diagram 3

3.8 Sequence Diagram 4

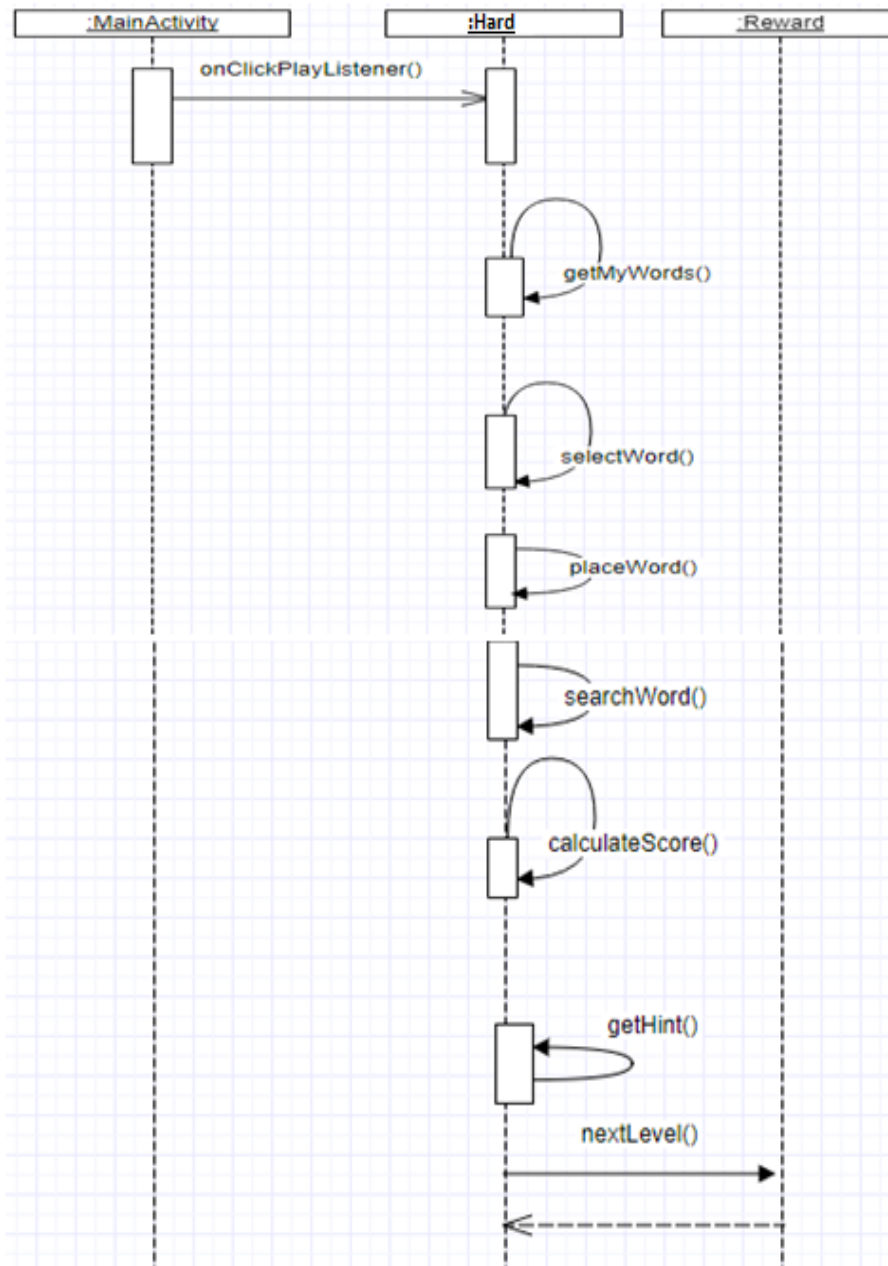


Figure 19: Sequence Diagram 4

3.9 Design Class Diagram

Design Class Diagram (DCD) show software class definitions. They are based on the sequence diagram. Classes are shown with their simple attributes and methods listed.

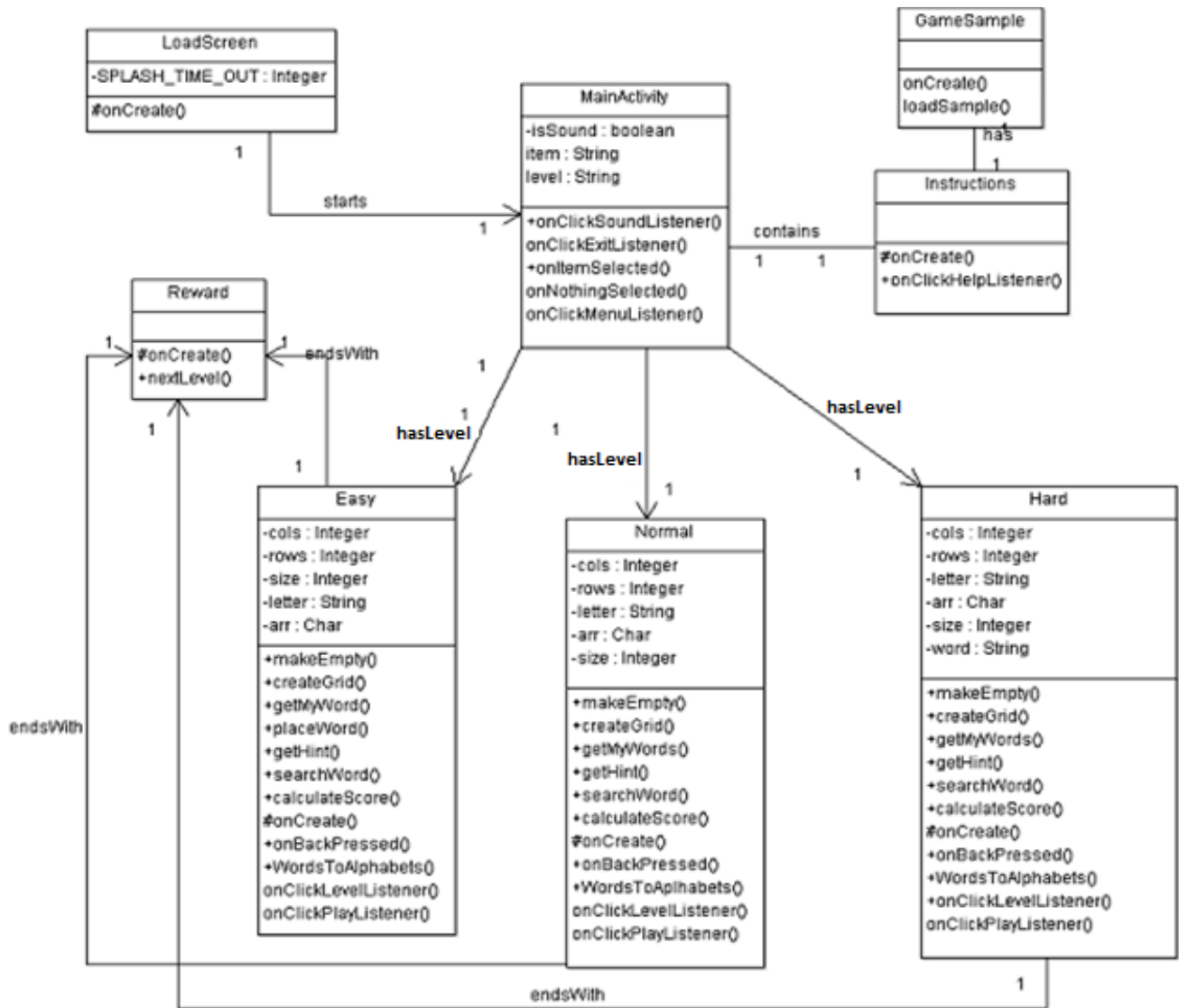


Figure 20: Design Class Diagram

3.10 User Interface Design

A description of the user interface design of the software is presented here:

3.10.1 Description of the User Interface

The user interface (UI) is android-based and provides a visual front-end to the user to interact with the game. The user interface consists of many features so that the user will be able to interact with the system. It will provide the “Read Instruction” button, sound button. In case of any ambiguity, the user will read the instruction to play the game. Screen resolution will be kept in such a way that fits to

every android device. Sound icon is chosen instead of button, because icons look more suitable for the games. Game buttons are placed on the center of screen because their position matters according to HCI rules.

3.10.2 Screen Images (Rough Prototypes)



Figure 21: Prototype 1



Figure 22: Prototype 2

3.10.3 Objects and Actions

The interface consists of the play button, help button, sound button and exit button.

Play button/icon:

Play button will take the user to the next game screen which consists of a grid containing random words which the user has to find.

Help button/icon:

On pressing the Help button, user will be able to read out the instructions to play the game.

Sound button/icon:

On pressing the Sound button, user will be able to mute or unmute the sound.

Exit button/icon:

On pressing the exit button, it will be asked the user whether he want to exit the game or not. If the user chooses exit option, then the game will allow him/her to exit from game. On the second game screen which consists of score button/icon.

Score icon/button:

It will allow the user to check out the score during the game.



4. CHAPTER 4

Implementation

4.1 INTRODUCTION

At implementation stage, algorithms and the technical specifications are changed to the program, software component or any other computer system through programming and deployment. This stage involves framework selection, language selection, operating system and the application screenshots.

4.2 Framework Selection

Android Studio:

The FWG is developed using Android Studio 2.3 and Android Software Development Kit (SDK) with Android Development Tools (ADT) plugins. Android studio is the best framework to design different applications because it provides Gradle-based build support and android-specific refactoring and quick fixes. It also provides Template-based wizards to create common Android designs and components. It contains rich layout editor that allows you to drag-and-drop UI components, preview layouts on multiple screen configurations, and much more.

4.3 Language Selection

XML:

Extensible Markup Language is used to design the interfaces in android studio. Therefore the interfaces of FWG are designed through XML

Java:

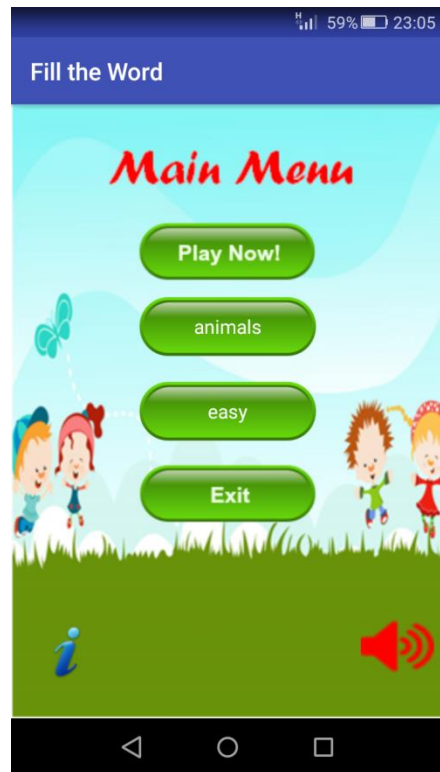
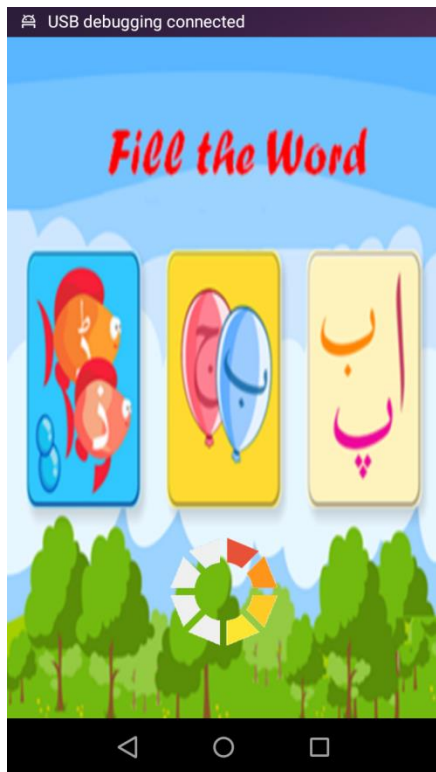
Java is the object oriented computer programming language. Java is a general-purpose computer programming language that is class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. So, java is used for implementing the algorithms and technical specifications of FWG. JDK consists of runtime environment that is used to run the java programs.

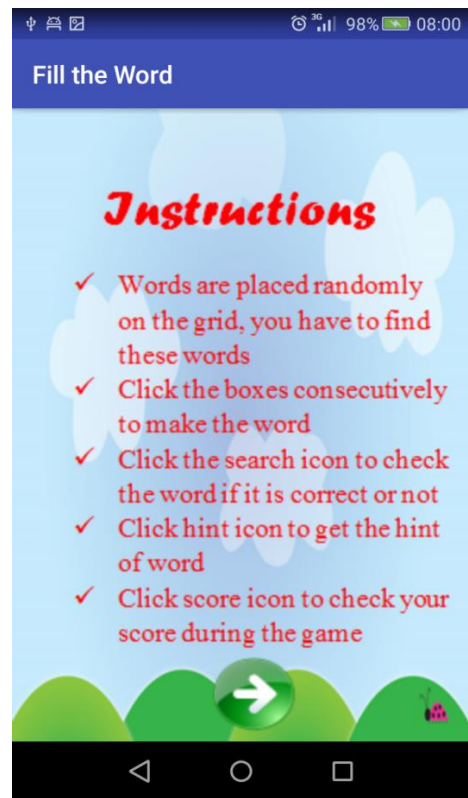
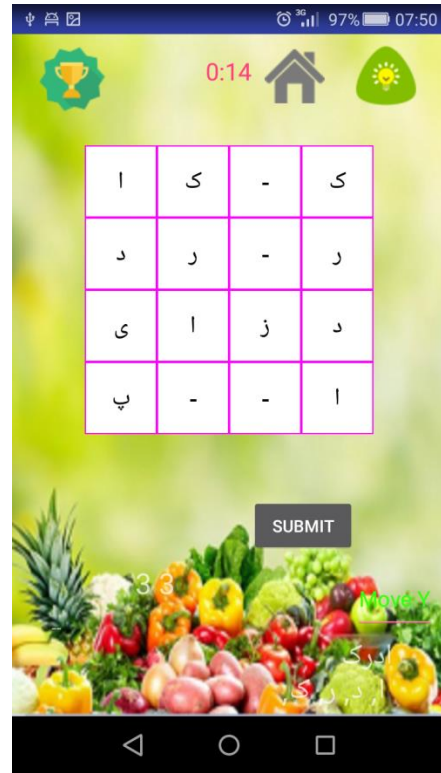
4.4 Operating System

Android

FWG is an android based game. It is the most widely used operating system in the modern world. Many updates have been released in the original version of android. Android 2.3 supports all the versions therefore FWG is designed and implemented on android 2.3.

4.5 FWG screenshots:





5. CHAPTER 5

Software Test Document

5.1 INTRODUCTION

5.1.1 Purpose

This document is the software test description of the FWG software development project. It contains the description of test cases. The tests are written to validate use cases, requirements (both functional and non-functional), system architecture, and object design.

5.1.2 System Overview

This section describes the structural aspects of testing and provides an overview of the system in terms of the components that are tested during each test. All the components and their dependencies are defined in this section.

5.1.3 Test Approach

Testing technique used for the FWG project is black box testing for the time being. It is acceptance testing in which I am testing each requirement with the user point of view.[4]

5.2 TEST PLAN

5.2.1 Features to be tested

Start the game

This feature is to be tested, because when the user clicks the game icon, the system must respond and game menu should be displayed.

Read Instructions

It is to be tested that the functionality of “read instruction button” instruction must be performed by the system.

Play the game

It is to be tested that the functionality of “play button” should be performed successfully.

Manage Sound

It is to be tested that the functionality of “sound” button should be performed correctly.

Design Grid filled with random words

It is to be tested that when the play button is clicked, the grid filled with random words must be displayed.

View the Score

It is to be tested that when the score button is clicked, the score must be displayed by the system

Mark words

It is to be tested that when the word is found then it should be marked.

Exit the game

It is to be tested that when the exit button is pushed, the game should be exited.

5.2.2 Features not to be tested

All features of the game will be tested.

5.3 Testing Tools and Environment

Beta-testing is performed temporarily therefore at this stage no testing tool is used. Later on after implementing the game, testing tools and environment will be used. [4]

5.4 TEST CASES

Test Case ID	Test Case Name	Test Case Description	Test Step Number	Test Case Inputs	Expected Outputs	Observed Output	Verdict
TC01	Start the game test	This is to test that when the game icon is pressed, it must start.	01	Click the game icon	Game has started	Game has started	Pass

Table 2: Test Case1

Test Case ID	Test Case Name	Test Case Description	Test Step Number	Test Case Inputs	Expected Outputs	Observed Outputs	Verdict
TC02	Manage Sound test	This is to test that when the background sound is off, the sound off button should be displayed and vice versa	01	Select the sound off button	Background sound is off and sound off button is displayed	Background sound is off and sound off button is displayed	Pass
			02	Select the sound on button	Background sound is on and sound on button is displayed	Background sound is on and sound on button is displayed	Pass

Table 3: Test Case2

Test Case ID	Test Case Name	Test Case Description	Test Step Number	Test Case Inputs	Expected Outputs	Observed Outputs	Verdict
TC03	Play the game test	This is to test that when the play button is pressed, the grid screen should be displayed and timer should be started.	01	Click the game icon	The grid screen is displayed and timer has started.	The grid screen is displayed and timer has started	Pass

Table 4: Test Case 3

Test Case ID	Test Case Name	Test Case Description	Test Step Number	Test Case Inputs	Expected Outputs	Observed Outputs	Verdict
TC04	Select the word test	This is to test that when the play button is pressed, System should select words from the file successfully	01	Click the game icon	Words are selected by the system successfully	Words are selected by the system successfully	Pass

Table 5: Test Case 4

Test Case ID	Test Case Name	Test Case Description	Test Step Number	Test Case Inputs	Expected Outputs	Observed Outputs	Verdict
TC05	Place words test	This is to test that when the play button is pressed, the grid filled with random words is displayed	01	Select the play button	Grid is designed and words are placed on grid randomly	Errors occurred during the placement of words on the grid Words placed on the grid successfully	Fail Pass

Table 6: Test Case 5

Test Case ID	Test Case Name	Test Case Description	Test Step Number	Test Case Inputs	Expected Outputs	Observed Outputs	Verdict
TC06	Search words test	This is to test that when the user finds words correctly, the words should be marked	01	Find the word from grid	Word is present in the dictionary and it is marked by the system	Word search function executed successfully	Pass

Table 7: Test Case 6

Test Case ID	Test Case Name	Test Case Description	Test Step Number	Test Case Inputs	Expected Outputs	Observed Outputs	Verdict
TC07	View the score test	This is to test that when the user presses score button, his score should be displayed	01	Click the score button	Score is displayed on the screen	Score is displayed as "0" each time Score is displayed on the screen	Fail Pass

Table 8: Test Case 7

Test Case ID	Test Case Name	Test Case Description	Test Step Number	Test Case Inputs	Expected Outputs	Observed Outputs	Verdict
TC08	Read instructions test	This is to test that when the read instruction button is pressed, the instructions should be displayed.	01	Select the read instruction button	The instructions are displayed on the screen	The instructions are displayed on the screen successfully	Pass

Table 9: Test case 8

Test Case ID	Test Case Name	Test Case Description	Test Step Number	Test Case Inputs	Expected Outputs	Observed Outputs	Verdict
TC09	View the hints test	This is to test that when the hint icon is pressed, the hints should be displayed	01	Press the hint icon	Hint icon displayed the hints	Hint icon displayed the hints	Pass

Table 10: Test Case 9

5.5 Requirement Traceability Matrix

Requirements	UC01	UC02	UC03	UC04	UC05	UC06	UC07	UC08	UC09	UC10
Test Cases										
TC01	x									
TC02			x							
TC03				X						
TC04										
TC05										
TC06										
TC07								x		
TC08		x								
TC09									x	

Table 11: RTM between Test Cases and Requirements

5.6 APPENDIX

References:

Books	<ol style="list-style-type: none">1. C. Larman, APPLYING UML AND PATTERNS An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd ed., Massachusetts: Pearson Education, 20052. Roger S. Pressman, Software Engineering - A Practitioner's Approach, McGraw Hill, 7th Edition, 2010
Websites	<p>Visited on 29-01-2017</p> <ol style="list-style-type: none">3. http://www.cs.utah.edu/~jamesj/ayb2005/docs/SDS_v2.htm <p>Visited on 29-01-2017</p> <ol style="list-style-type: none">4. https://www.tutorialspoint.com/software_testing_dictionary/black_box_testing.html <p>Visited on 2-06-2017</p> <ol style="list-style-type: none">5. https://en.wikipedia.org/wiki/System_sequence_diagram <p>Visited on 2-06-2017</p> <ol style="list-style-type: none">6. https://www.smartdraw.com/activity-diagram/
Instructor	Dr. Onaiza Maqbool

Table 12: References

6. CHAPTER 6

Conclusions & Future Enhancements

6.1 Summary

This is an android gaming application which helps the people to learn and improve their Urdu vocabulary. This game serves the educational purpose. This game consists of grid that contains Urdu words placed randomly on it. Children are interested in playing games instead of studying. So, by playing this game they would be able to learn Urdu words with fun.

6.2 Conclusion & Future Enhancements

6.2.1 Conclusion

This is a very useful gaming application for the people of all age groups. Instead of wasting time on playing such games that do not increase your mental abilities, one should play this game because this will serve two purposes i.e. fun as well as education.

6.2.2 Future Assessment

In future, this gaming application can be enhanced by

1. Including features that will enable the people to learn more about Urdu language such as Urdu grammar.
2. Including a feature of two players. So that two people would play the game in order to compare their knowledge of Urdu Language

