BSCS FINAL PROJECT

Turkishya

Term: Fall 2021

Project Advisor

PROF. AYESHA SIDDIQA

Presented by:
Group ID: UCP-GRTBSCSF18-001

Student Reg#	Student Name
G3F18BSCS0025	Mariam Ikram
G3F18BSCS0021	Fardous Munawar
G3F18BSCS0029	Sawera Rasheed

Faculty of Information Technology

University of Central Punjab

University of Central Punjab

Software Design Specification

SDP Phase-II

Turkishya

Advisor: Prof. Ayesha Siddiqa

Group <UCP-GRTBSCSF18-001>

Member Name	Primary Responsibility
Mariam Ikram	Development / Design
Fardous Munawar	Design / Documentation
Sawera Rasheed	Documentation

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
SRS-1	24-01-22	Format, Gantt Chart was not included; use case was needed to be modified.	1.0

1. Introduction and Background

1.1 Product

Turkishya is all about creating a well-organized Turkish language course from scratch that can help anyone who wants to/has to (students, tourists, business people, international workers, etc.) to learn this language. It is a mobile application that is capable of teaching Turkish to an absolute beginner starting from alphabets through listening, reading, writing and speaking exercises with necessary grammar logics provided before the beginning of a lesson (where required). It is coded in React Native and Firebase is used for back end.

1.2 Background

Turkishya means “towards Turkish” in Turkish language. We all know that media has a great influence on people these days. As, the major religion in Turkey is Islam, our culture is also a lot similar and Turkish media industry has produced and is still producing a lot of phenomenal series that are based on Islamic culture and people are eager to watch them. Watching such series on YouTube and Netflix creates an interest in people to learn this language. Also, tourist rate of Turkey has also increased in recent years. One of the prime reasons for this increase in rate is social media trend. Influencers and common people want to visit different places in Turkey in order to jump on the bandwagon of trend and for the sake of their social media profiles. Also, there are a lot of cultural and religious sites and monuments that are actually worth visiting such as Cappadocia, Blue Mosque, Hagia Sophia Mosque etc. Visitors have the biggest problem of language barrier in Turkey as majority does not know English there (only $1/5^{th}$ of the population knows English). They can be business people, international workers or students. Such people do not have enough time in their working schedule to attend physical language learning centers in order to communicate properly. Hence, the sole purpose of our project is to create an application that teaches people Turkish language without any hurdle and with all the ease from scratch at their own pace on their personal mobile phones.

1.3 Scope

Turkishya is a mobile application that will enable beginners who want to / have to learn Turkish from scratch with all the basic listening, speaking, reading and writing exercises.

Turkishya is an ideal application if you want to learn Turkish from the very basic level (alphabets). It differs from other Turkish learning applications in a way that all other applications provide unstructured and abstract course and they may be, mere a boring and lazy translation of same material. But that is not the case with Turkishya.

Turkishya will enable the user to share his result report with friends and family to motivate others as well as to remain motivated, consistent and enthusiastic towards his own course.

1.4 Objectives

Our mobile application will be capable of:

- Teaching Turkish language starting from alphabets
- Covering all the reading,
- listening,
- writing and
- speaking exercises
- with given basic grammar rules at every lesson (where required).

1.5 Challenges

We will face problem while building logic because Turkish is not out first language and it has a completely different sentence structure than English.

We will also face problem while using React Native and Firebase as we did not use them before.

1.6 Learning Outcomes

In this Project, we will learn

- React Native
- JavaScript
- Firebase database
- VS Code
- Canva

1.7 Nature of End Product

At the end, Turkishya will provide a Turkish learning course for absolute beginners starting from alphabets with a skill sheet that will be divided into chapters. Chapters will have different skills and skills will have different number of lessons at every level for learning, practicing and taking a final test. It will enable the user to share his result report with friends and family.

1.8 Completeness Criteria

The application will be completed when the user:

- Can register and then login
- Can view skill sheet with some locked content (skills)
- Can start learning Turkish from scratch
- Can practice all skills through speaking, listening, reading and writing and
- Can unlock the content after practicing lessons one by one until he reaches the end
- Can share the result report with friends and family
- Can download result report

1.9 Business Goals

Our mobile application will:

- Enable the user to read and understand any written Turkish material.
- Enable the user to listen to and understand the native Turkish speakers.
- Enable the user to communicate (speaking/writing) with native Turkish speakers.

1.10 Related Work

Duolingo teaches everyday Turkish very well but it does not teach basics and sentence structure, which is a major turn off for the absolute beginners. It will not teach grammar lessons, which can be fun if one likes to figure out the grammar of the language through a puzzle, or frustrating if one just can't figure out how something works.
Babbel has an appealing UI but at the beginning, there is no information about how the course is structured (how many questions & tasks are there?). Some apps are set up like a game, while Babbel is set up more like traditional instruction and can become a little repetitive after a while.
Busuu is a free course but it is nothing more than boring, lazy translation of same material. Some exercises don't even have translations and there is no entertainment at all along with learning.

1.11 Document Conventions

SDS: Software Design Specification

2. Overall Description

2.1 Product Features

- Turkishya is a well-organized Turkish language course from scratch that can help anyone to learn Turkish.
- Turkishya has a skill sheet divided into different skills (basic phrases, nations, food, sentence making etc.).
- Turkishya will enable the user to learn Turkish from alphabets without any hurdle and with all the ease at his own pace of comfort on his personal mobile phone.
- Turkishya will cover all the reading, listening, writing and speaking exercises.
- Turkishya will enable the user to share his result report with friends and family to motivate others as well as to remain motivated, consistent and enthusiastic towards his own course.

2.2 User Classes and Characteristics

User:

- registers
- logins
- learns alphabets
- practices a chapter
- starts a skill
- learns new vocabulary at level 1
- practices at level 2 and 3
- takes test at level 4
- shares result report
- downloads result report

2.3 Operating Environment

For using our application, the user must have to login. Other hardware or software specifications are as follow:

- Smart phone
- Wi-Fi or mobile network connection
- Speaker works properly
- Microphone works properly
- Touchpad works properly for writing exercises

2.4 Design and Implementation Constraints

User interface will be composed using React Native, Visual Studio Code and Canva.

Hardware Requirements:

- PC
- Processor core i3 or above
- Hard Disk space 20 GB

Software Requirements:

- Firebase database
- React Native
- Visual Studio Code
- Canva

2.5 Assumptions and Dependencies

- We assume that the user has an internet connection.
- We assume that the user can understand English.
- We assume that user has properly working speaker, microphone and touchpad / keypad.
- The end user of this software is assumed to have basic level of app knowledge.
- We assume that the user must be login before using the application.

3. Functional Requirements

User:

- registers
- logins
- learns alphabets
- practices a chapter
- starts a skill
- learns new vocabulary at level 1
- practices at level 2 and 3
- takes test at level 4
- shares result report
- downloads result report

Use-Cases

3.1 Use Case: Register

Identifier	UC-1	
Purpose	Create an account	
Priority	High	
Pre-conditions	Install the application	
Post-conditions	Login & use application	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Click register option	Open registration form
2	Fill in the registration form	Store the data in database
3		Registered successfully
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Click register option	Open registration form
2	Fill in the registration form	Data not filled properly
3		Ask to re-enter the data

Table 1: Register

3.2 Use Case: Login

Identifier	UC-2	
Purpose	Login to the application	
Priority	High	
Pre-conditions	Must be registered	
Post-conditions	Skill sheet is visible	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Click login option	Open the login form
2	Fill in the login form	Verify the data and grant access to skill sheet
3		Logged in successfully
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Click login option	Open login form
2	Fill in the login form	Data do not match
3		Ask to re-enter the data

Table 2: Login

3.3 Use Case: Learn Alphabets

Identifier	UC-3	
Purpose	Learn alphabets	
Priority	Medium	
Pre-conditions	Login to application	
Post-conditions	1st skill gets unlocked	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Click on alphabets option	Open alphabet sheet
2	Click on alphabet	Show alphabet, related vocabulary and pronounce the alphabet
3	Click on skip/ finish all alphabets	Open the test sheet
4	Take test	Validate the user input and unlock the 1st skill
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Click on alphabets option	Open alphabet sheet
2	Click on alphabet	Show alphabet, related vocabulary and pronounce the alphabet
3	Click on skip/ finish all alphabets	Open the test sheet
4	Take test	Incorrect user input
		Redirect to test sheet (retest)

Table 3: Learn Alphabets

3.4 Use Case: Practice a Chapter

Identifier	UC-4	
Purpose	Learn, practice and test	
Priority	High	
Pre-conditions	Pass the alphabet test / Pass previous chapter's test	
Post-conditions	Access to subsequent chapter / course completed	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Click on skill	Open skill having levels
2	Finish/practice through levels	Open corresponding lessons and validate data
3	Take "End of the Chapter" test	Validate user input and unlock next chapter/congratulate about course completion
4	Share result report	Open share via
5	Click on exit	Hide the result report
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Click on skill	Open skill having levels
2	Finish/practice through levels	Open corresponding lessons and validate data
3	Take "End of the Chapter" test	Incorrect User Input
		Redirect to test (retest)

Table 4: Practice a Chapter

3.5 Use Case: Start a Skill

Identifier	UC-5	
Purpose	Learn a skill	
Priority	High	
Pre-conditions	Start a chapter / Pass test of previous skill	
Post-conditions	Practice through the skill	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Click on skill	Open skill levels
2	Click levels	Show learn, practice and test lessons in 4 levels

Table 5: Start a Skill

3.6 Use Case: Level 1 – Learn New Vocabulary/Other Stuff (sentences, grammar etc.)

Identifier	UC-6	
Purpose	Learn new vocabulary/stuff	
Priority	High	
Pre-conditions	Start a skill	
Post-conditions	Access to next level	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Click on level 1	Show new vocabulary
2	Click on word	Show related grammar rules (where given and required) and pronunciation

Table 6: Level 1 - Learn New Vocabulary/Other Stuff

3.7 Use Case: Level 2 and 3 - Practice

Identifier	UC-7	
Purpose	Practice the learnt vocabulary	
Priority	Medium	
Pre-conditions	Finish level 1	
Post-conditions	Able to take test in level 4	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Click on levels	Open practice lesson
2	Click on skip/ complete practice	Validate input
		Level completed successfully
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Click on levels	Open practice lesson
2	Click on complete practice	Incorrect user Input
		Redirect to lesson

Table 7: Level 2 and 3 - Practice

3.8 Use Case: Level 4 – Take Test

Identifier	UC-8	
Purpose	To take test of learnt skill	
Priority	High	
Pre-conditions	Learn and/or practice new vocabulary/stuff	
Post-conditions	Share result report	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Click on test lessons	Open test lessons
2	Solve given question	Validate user input and unlock next skill / give access to "End of chapter" test
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Click on test lessons	Open test lessons
2	Solve given question	Incorrect user input
		Redirect to the test (retest)

Table 8: Level 4 - Take Test

3.9 Use Case: Share Result Report

Identifier	UC-9	
Purpose	Share result report	
Priority	Low	
Pre-conditions	Complete a skill / chapter	
Post-conditions	Start a new skill or chapter / exit	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Click on share report	Open share result via
2	Click on exit	Hide the result report

Table 9: Share Result Report

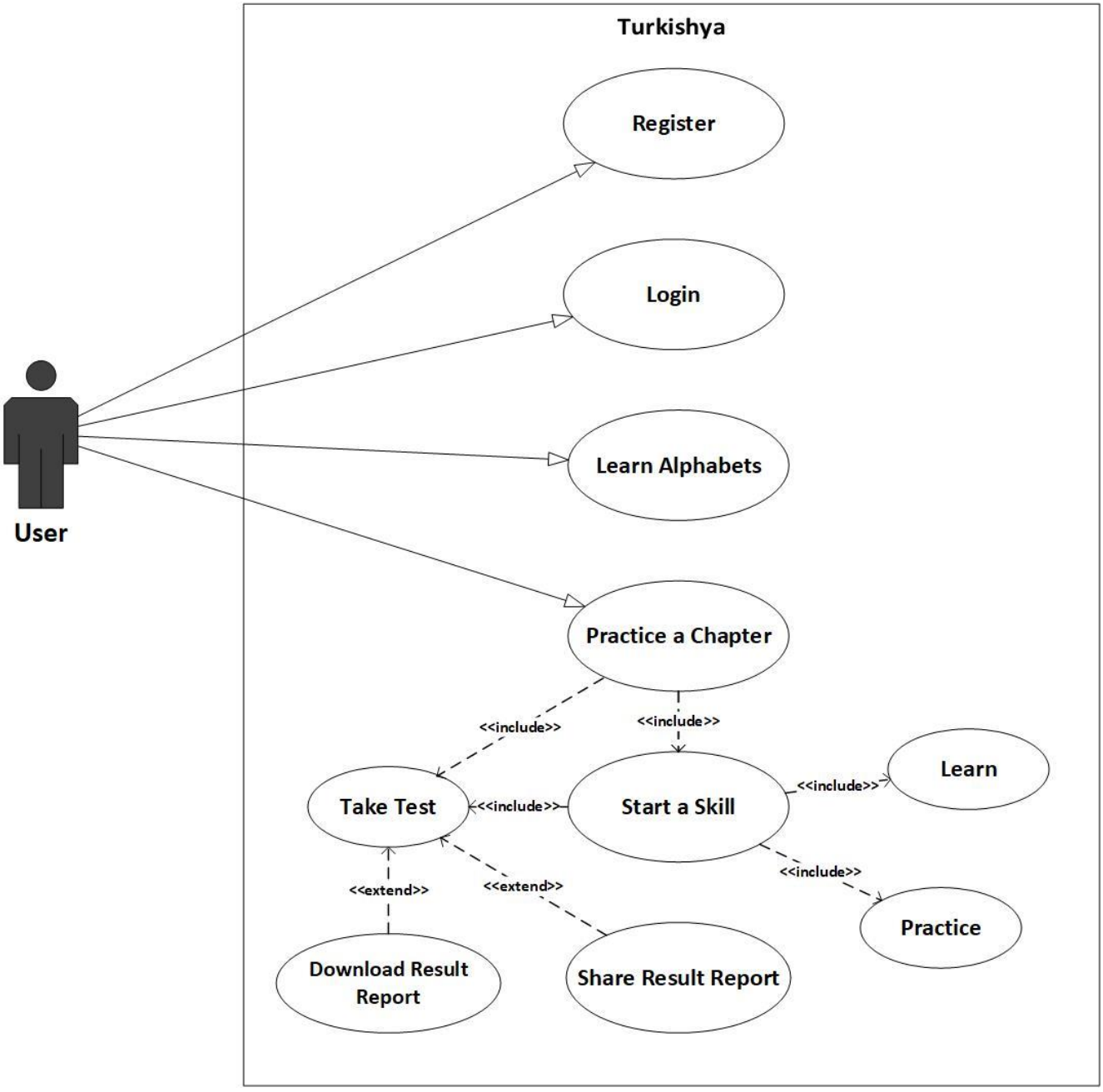
3.10 Use Case: Download Result Report

Identifier	UC-10	
Purpose	Download result report	
Priority	Low	
Pre-conditions	Complete a skill/ chapter	
Post-conditions	Start a new skill or chapter / exit	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Click on download report	Download the report
2	Click on exit	Hide the result report

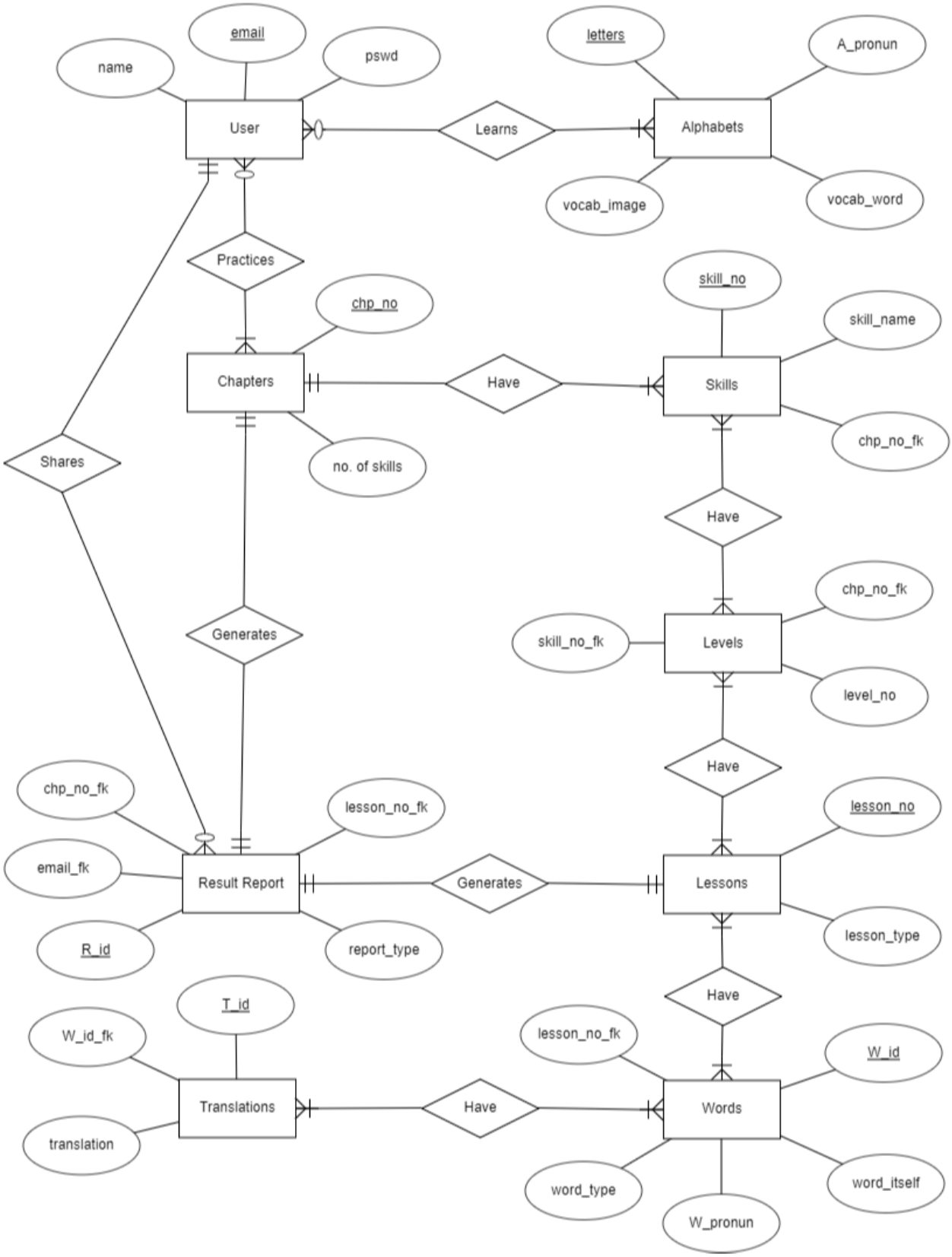
Table 10: Download Result Report

4. Requirements Analysis and Modeling

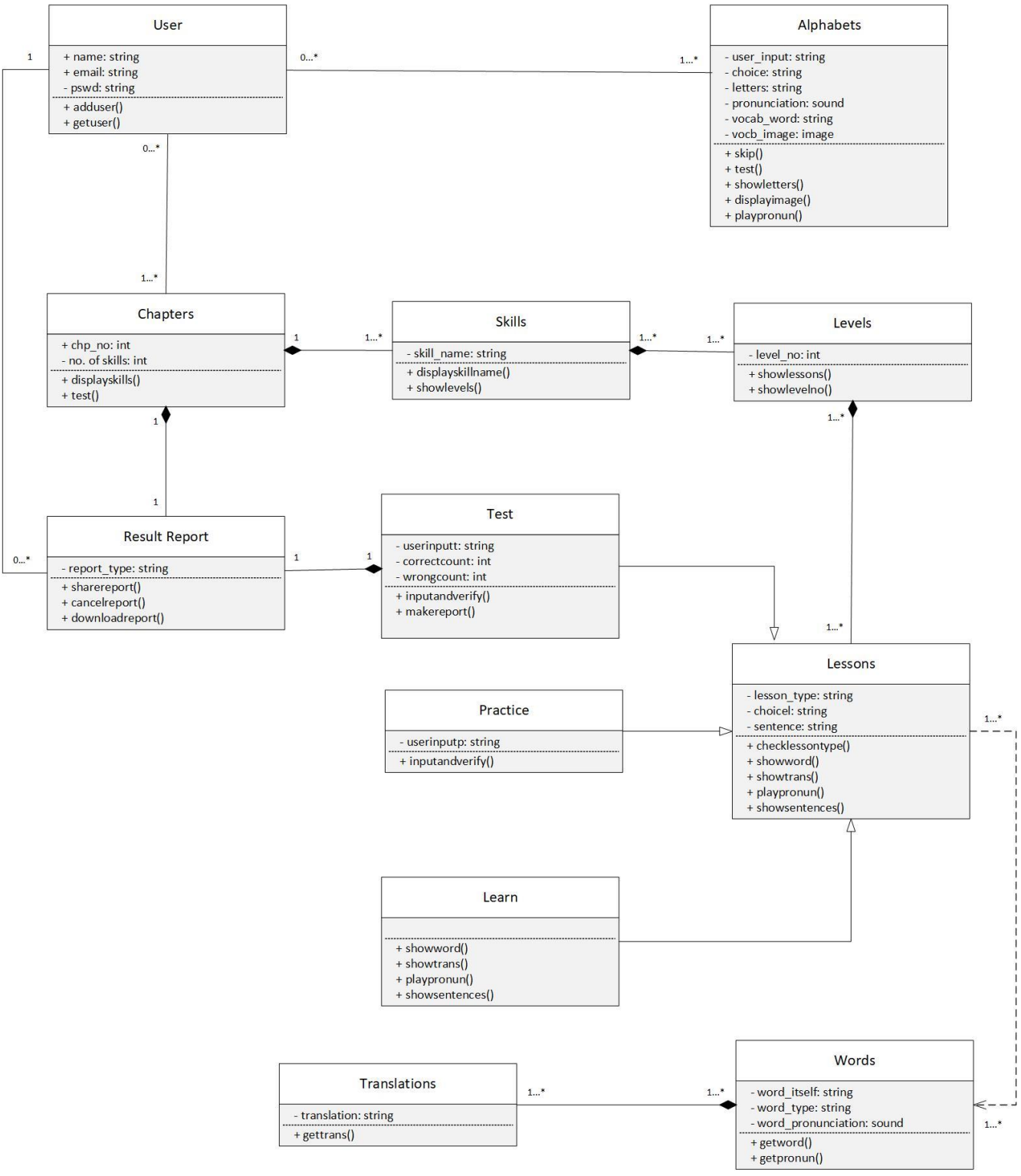
4.1 Use Case Diagram



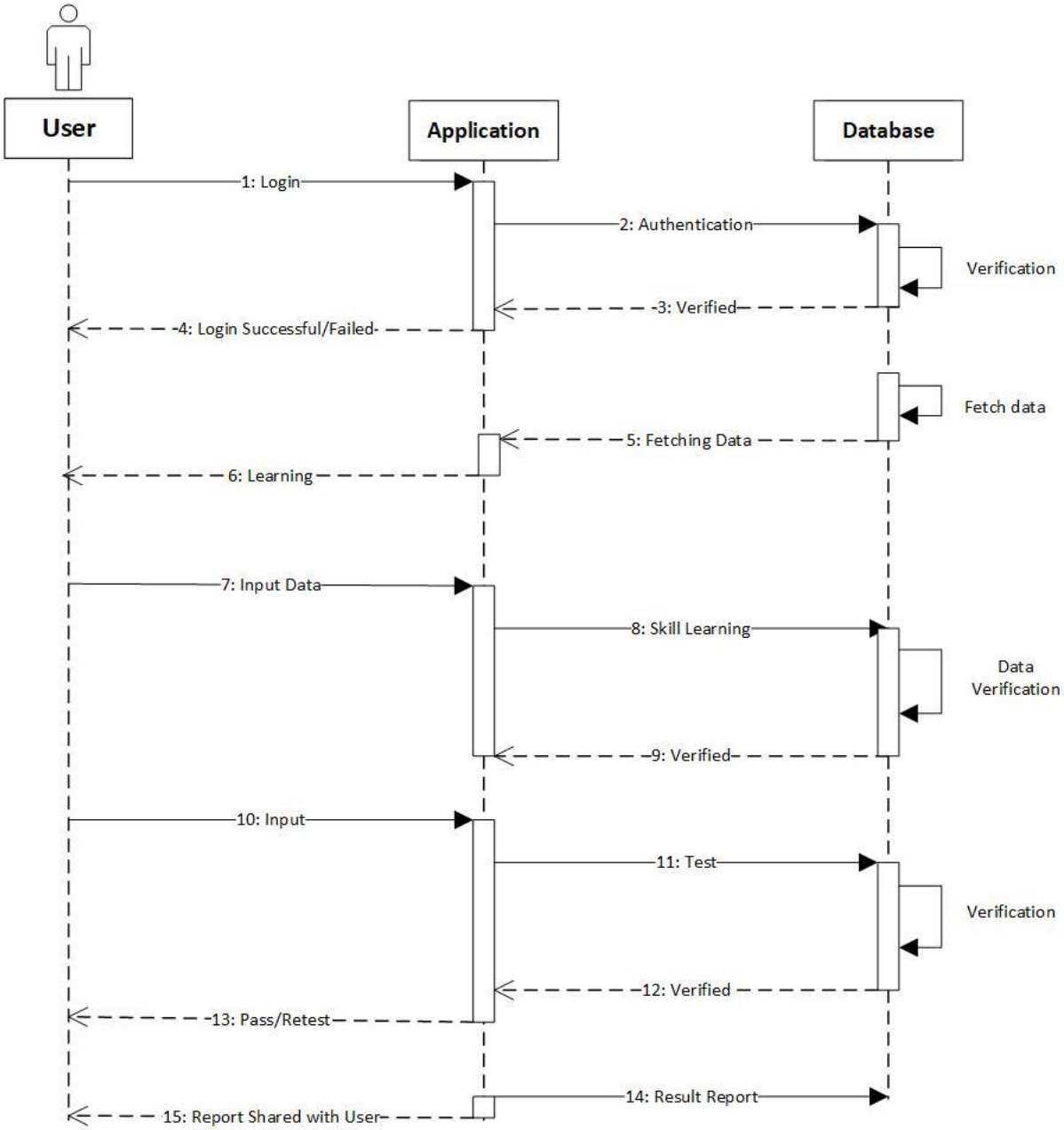
4.2 ER Diagram



4.3 Class Diagram

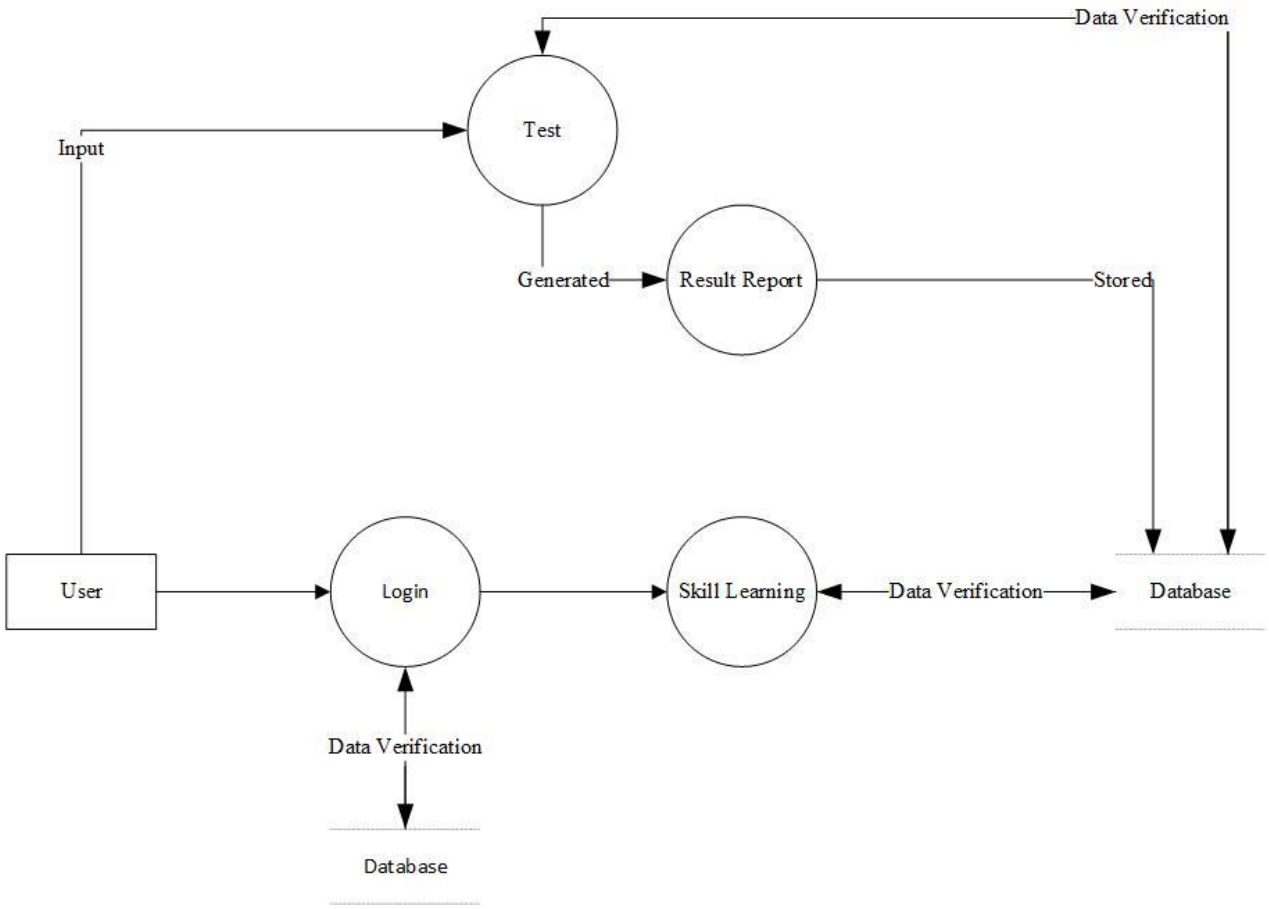


4.4 Sequence Diagram

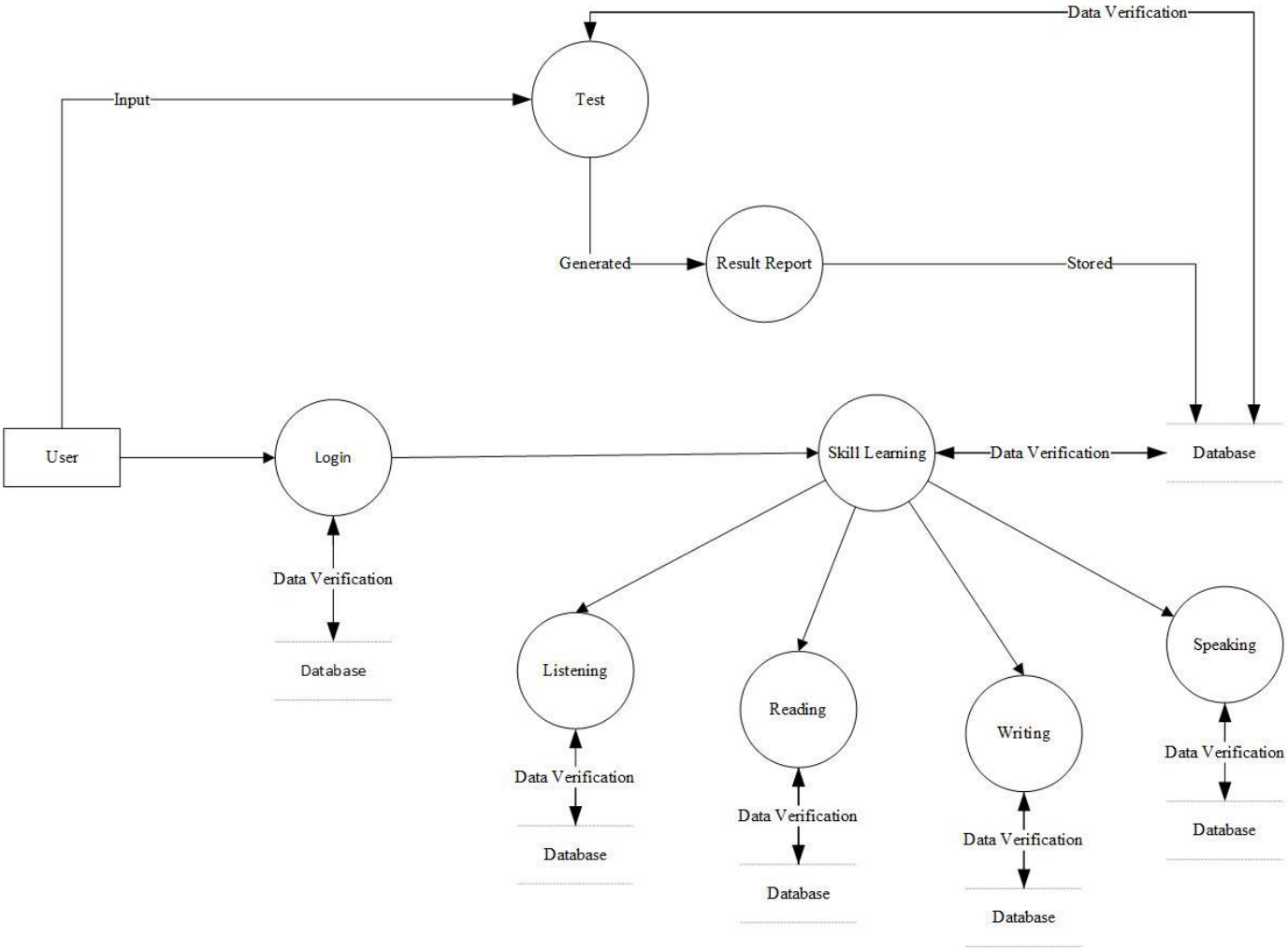


4.5 Data Flow Diagram

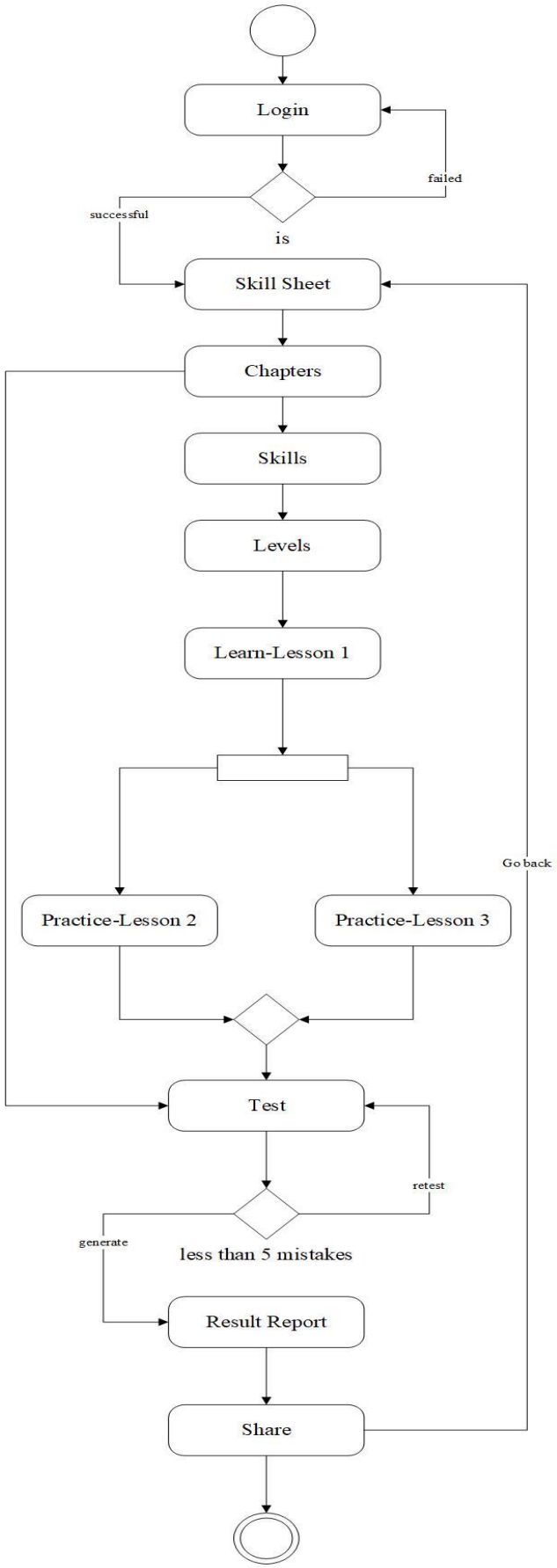
Level - 0



Level – 1

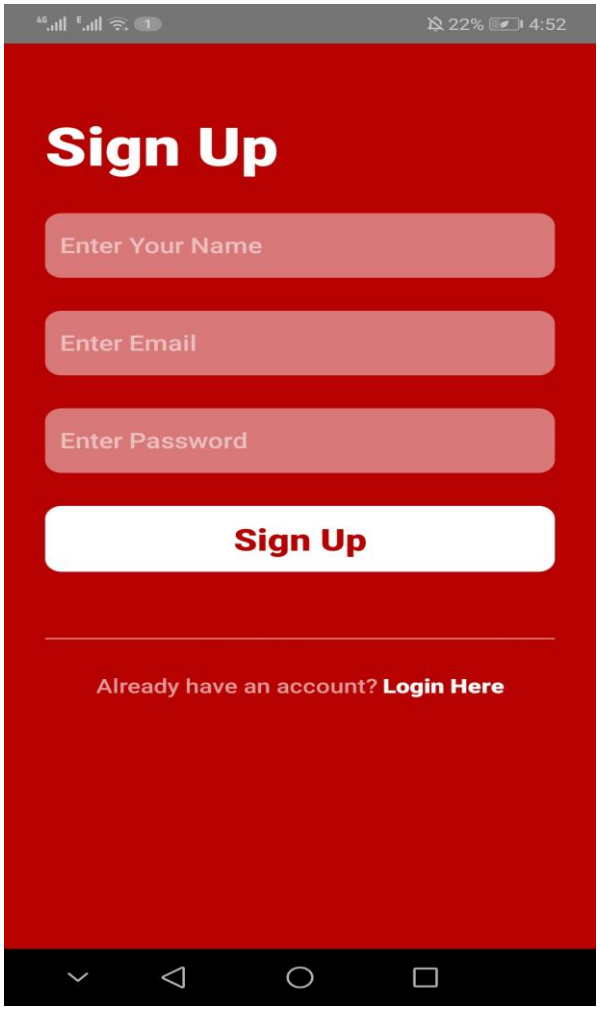

Data Verification
Input
Test
Generated
Result Report
Stored
User
Login
Skill Learning
Data Verification
Database
Data Verification
Database
Listening
Reading
Writing
Speaking
Data Verification
Database
Data Verification
Database
Data Verification
Database
Data Verification
Database

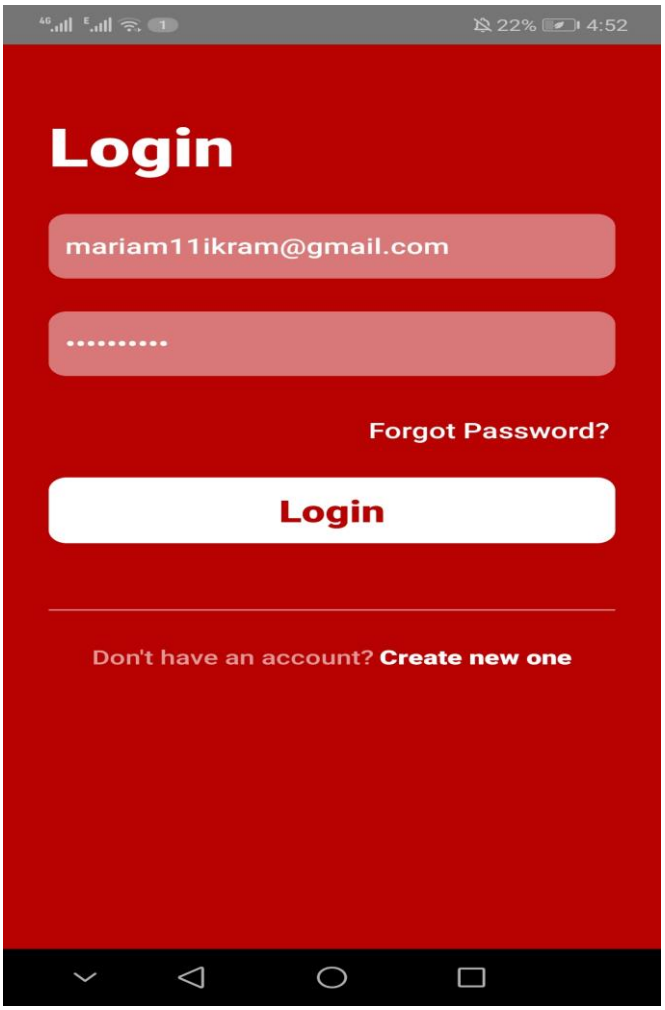
4.6 Activity Diagram

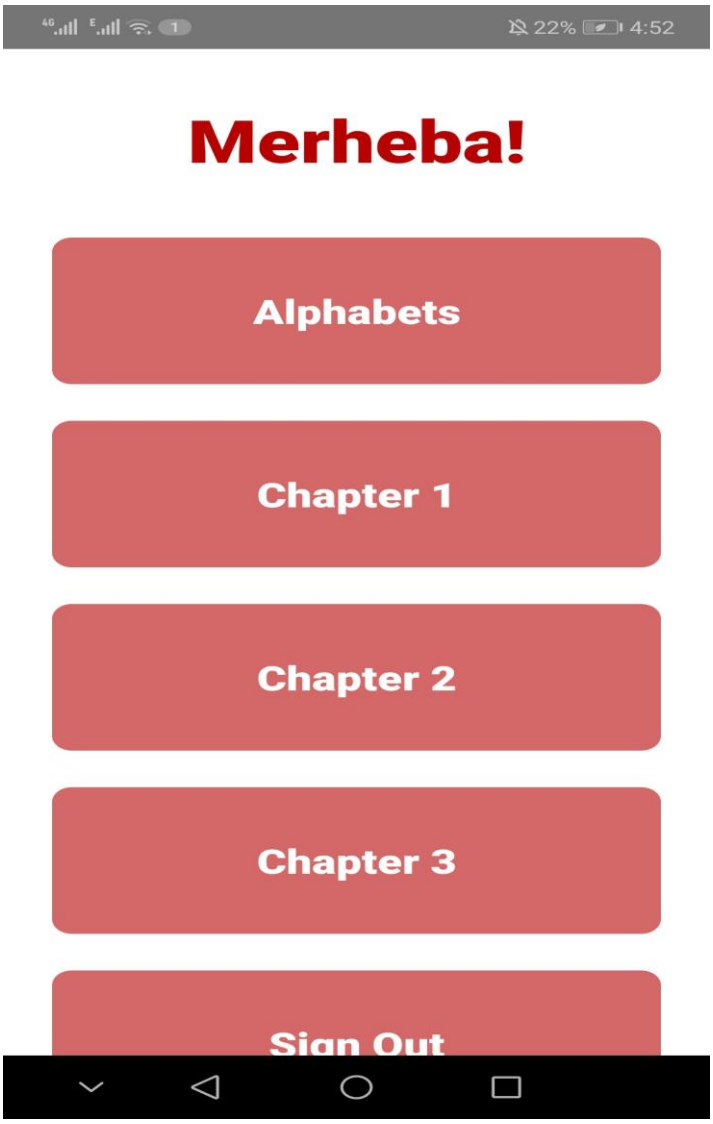


4.7 User Interface

Screenshots


22% 4:52
Sign Up
Enter Your Name
Enter Email
Enter Password
Sign Up
Already have an account? Login Here


22% 4:52
Login
mariam11ikram@gmail.com
Forgot Password?
Login
Don't have an account? Create new one


22% 4:52
Merheba!
Alphabets
Chapter 1
Chapter 2
Chapter 3
Sign Out


22% 4:52
Alphabets
Learning Sheet
Skip to Test


22% 4:52
Cheat Sheet
A / a
Letter Pronunciation
Word: At
Meaning: Horse
Word Pronunciation
Next
Back
Give Test

5. Nonfunctional Requirements

5.1 Performance Requirements

Application should:

- Open and load within 3-5 seconds
- Reload the skill sheet immediately and accurately after user interaction
- Validate the data accurately at the time of tests

5.2 Safety Requirements

All data about user's progress should be kept safe and should not be lost.

5.3 Security Requirements

At the time of login, there should be a strict authentication of user. Unauthorized users should not be allowed to use the application.

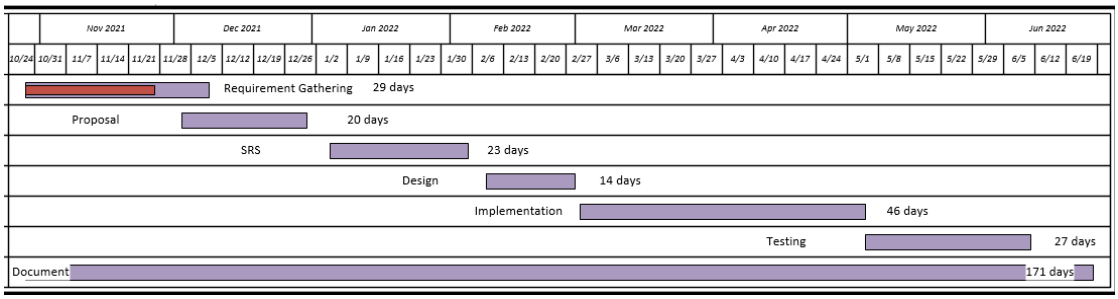
5.4 Software Quality Attributes

System should be:

- Consistent in performance
- Safe and secure
- User friendly
- Efficient
- Easily available

6. Revised Project Plan

	Task Name	Resource Names	Start	Finish	Duration	% Complete
1		Requirement Gathering	10/29/2021	12/8/2021	29d	70%
2		Proposal	12/3/2021	12/30/2021	20d	0%
3		SRS	1/5/2022	2/4/2022	23d	0%
4		Design	2/9/2022	2/28/2022	14d	0%
5		Impementation	3/2/2022	5/4/2022	46d	0%
6		Testing	5/5/2022	6/10/2022	27d	0%
7		Document	10/29/2021	6/24/2022	171d	0%



7. References

https://www.business2community.com/tech-gadgets/top-5-databases-for-react-native-app-development-02383205
https://templatelab.com/gantt-chart-templates/
https://doublespeakdojo.com/how-common-is-spoken-english-in-turkey/
https://www.planetware.com/tourist-attractions/turkey-tr.htm
http://blog.cambridgecoaching.com/learning-a-foreign-language-a-review-of-duolingo
https://specialistlanguagecourses.com/review-language-learning-app-babbel/
https://ling-app.com/tips/busuu-review/
https://erdplus.com/
https://creately.com/diagram-type/template/gstx57ng1/er-diagram
https://www.lucidchart.com/pages/data-flow-diagram
https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/
https://www.tutorialspoint.com/uml/uml_activity_diagram.htm
https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/
https://creately.com/blog/diagrams/sequence-diagram-tutorial/
https://www.smartdraw.com/sequence-diagram/

The name was chosen by us keeping in view the nature of our application

8. Appendix A: Glossary

SRS: Software Requirements Specification

SDS: Software Design Specification

UML: Unified Modelling Language

UC: Use Case

ERD: Entity Relationship Diagram

DFD: Data Flow Diagram

Appendix B: IV & V Report

(Independent verification & validation)

IV & V Resource

Name Signature

S#	Defect Description	Origin Stage	Status	Fix Time	
				Hours	Minutes
1					
2					
3					
…					

Table 2: List of non-trivial defects