Kingdom of Saudi Arabia
Ministry of Higher Education
King Abdul Aziz University
Faculty of Computing and IT
Computer Science Department

MORSE LIGHTHOUSE: AN ARTIFICIAL INTELLIGENCE BASED MORSE CODE LEARNING MOBILE APPLICATION

Submitted by

Mohammed Atiah Alzahrani - 1740166

Omar Abdulaziz Alqurashi - 1742589

Mohammed Saleh Alharbi - 1740373

Hussam Adnan Shawly – 1742403

Supervised by

Dr. Asif I. Khan

Dr. Wajdi H. Aljedaibi

Shaban, 1442 – 2021, April

بسم الله الرحمن الرحيم

قال تعالى: ﴿يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ﴾ [المجادلة: 11]

عن أبي هريرة - رضي الله عنه - قال:

قال رسول الله ﷺ ((من سلك طريقًا يلتمس فيه علمًا سهل الله له به طريقًا إلى الجنة)) أخرجه مسلم

Certification of Research

I certify that I have reviewed this project done by:
Mohammed Atiah Alzahrani-1740166,

Omar Abdulaziz Alqurashi-1742589,

Mohammed Saleh Alharbi-1740373, and

Hussam Adnan Shawly-1742403

during the ***1st and 2nd*** semester of the academic year ***2021***. And I approve the project submission for discussion by the project responsible committee to grade it as the graduation project degree in ***Computer Science.***

Supervisor: Dr. Asif I. Khan

Signature: ……

Co-Supervisor: Dr. Wajdi H. Aljedaibi

Signature: ……

Declaration

We certify that the implementation of this graduation project has been done by our efforts. This includes the preparation and writing of the analysis, design, programming, and documentation. We have got assistance during the implementation period of this project, after Allah, from the references mentioned in this thesis, and Allah is the witnesses of what we say.

Student Name: Mohammed Alzahrani
Student ID: 1740166
E-mail: Malzahrani1150@ stu.kau.edu.sa
Mobile: 0506692884
Signature: *Alzahrani*

Student Name: Omar Alqurashi
Student ID: 1742589
E-mail: oalqurashi0018@stu.kau.edu.sa
Mobile: 0549281172
Signature: *Alqurashi*

Student Name: Mohammed Alharbi
Student ID: 1740373
E-mail: Malluhaybi0008@stu.kau.edu.sa
Mobile: 0550692224
Signature: *Alharbi*

Student Name: Hussam Shawly
Student ID: 1742403
E-mail: hshawly@stu.kau.edu.sa
Mobile: 0598448617
Signature: *Shawly*

Acknowledgements

We would like to express our utmost gratitude and sincerity to King Abdulaziz University and all people who have helped us during this endeavor, only with their guidance, help, and support, we have made it to the end of this project.

We would also like to thank the people at FCIT for their great effort in this project during these troubling times by facilitating our journey.

We would also like to specifically thank our project supervisor Dr. Asif Khan and co-supervisor Dr. Wajdi Aljedaibi for their valuable support and guidance throughout this project.

MORSE LIGHTHOUSE: AN ARTIFICIAL INTELLIGENCE BASED MORSE CODE LEARNING MOBILE APPLICATION

Abstract

Morse code is an old method for encoding letters used for communication. Morse code is still used and has various usages in our current age. However, the methods used for teaching Morse code are outdated and rely on traditional learning techniques. These techniques do not make use of modern technology or adapt to the latest learning techniques, Hence the need to modernize the way Morse code is taught and make use of the newly available technology. There are two main methods, which was used to gather the data for this project, the first method is the questionnaire, which was conducted by Google forms and distributed to people from different educational backgrounds, and the second method is document analysis, which was conducted by reviewing five different academic papers. This project introduces a modern and more practical approach for learning Morse code, by making use of the deep learning method and computer vision for real-time detection of Morse code flashlight signals, and by developing an application that applies learning techniques such as gamification, haptic learning, and collaborative learning, on a smartphone system. In conclusion, the solution enhances Morse code learning and provides a practice tool for its users on a popular platform, which can be easily set up to accept Morse code input from people with limited motor skills.

منارة مورس: تطبيق للأجهزة الذكية مبني على ذكاء صناعي لتعليم شفرة مورس

نبذة مختصرة

شفرة مورس هي طريقة قديمة لتشفير الرسائل للتواصل، هذه الشفرة ما زالت مستخدمة، ولها عدة استخدامات في هذا العصر الحالي. ولكن الطرق المستخدمة حاليا لتعليم شفرة مورس هي في الواقع قديمة وتعتمد على طرق وتقنيات تقليدية. هذه الطرق لا تستفيد من التقنيات الحديثة ولم تتطور وترقى لتستخدم آخر أساليب التعليم الحديثة. لذلك كانت هنالك حاجة لتحديث الطريقة التي يتعلم بها الناس شفرة مورس باستخدام التقنيات المتطورة وأساليب التعليم الحديثة. في هذا المستند سنقدم طريقة حديثة وأكثر فعالية لتعليم شفرة مورس. باستخدام خوارزميات التعليم العميق ورؤية الكومبيوتر لفك تشفير إشارات شفرة مورس. وباستخدام أساليب تعليمية حديثة ومتطورة مثل التلعيب، والتعليم اللمسي والتعليم الجماعي. كل ذلك سيكون في نظام هاتف ذكي. فهكذا وبهذه الطريقة، النظام المقترح بإمكانه ان يزيد القدرة على استيعاب وتحليل وفك شفرة مورس. ويعتبر كأداة تدريب ممتازة. وهو أيضا سيكون متوفرا على منصة ذات وصول سهل. كانت هناك طريقتين جمعنا منها البيانات: أولها استبانة استخدمنا فيه منصة قوقل فورم ووزعناها على أناس من خلفيات تعليمية مختلفة، والطريقة الثانية كانت عن طريق تحليل المستندات والأوراق العلمية. وقد راجعنا فيها خمسة أوراق علمية أكاديمية محكمة. هذا المستند يقترح تصاميم تعبر عن النظام من مستوى عالي، وفي هذا المستند أيضا تصاميم تظهر المستوى المنخفض ويظهر فيها تفاصيل النظام ومكوناته. هذه التصاميم تظهر مبنى النظام، وطريقة عمله، وأدائه، ومجراه المنطقي في أداء الأوامر. هذه التصاميم تجعل النظام التي تصفه أسهل للفهم، وتعطي نظرة أوضح له. مما يسهل على المشروع والقائمين عليه أن ينتقلوا للمرحلة التالية من المشروع حسب ما هو موضح في الخطة.

Table of Contents

List of Tables

Figure

Appendices

INTRODUCTION

We are living in the age of information, where the technologies are quickly evolving, and the learning process is not an exemption, whether the environment is virtual or traditional. Although the learning techniques are evolving, yet Morse Code is still lagging. So, if a person wants to learn Morse Code, he/she needs an instructor that is proficient in it [1], or if he/she learns through a website that teaches Morse Code, however, it still uses the traditional techniques. So, creating a more interactive learning process through a mobile application with the gamification technique is suggested.

Morse code, which was created by Samuel Morse in the early 1830s, are literal codes used to send messages by a sequence of pulses of two types: long pulse and short pulse. These impulses compose of lights or some other media [2].

This project, Morse Lighthouse, is going to be an intelligent Morse code learning application for smartphones, which uses machine learning algorithms, gamification strategy, and collaborative learning to enhance the learning process for a more interactive experience.

This final report will be a combination of the final report for (CPCS-498) and (CPCS-499), with the help of the given guidelines for making it. The following sections of this report are:

1. Project Proposal
2. Domain Analysis
3. Data collection and Analysis Techniques
4. Requirements Specification
5. Analysis Stage
6. Design Stage
7. Implementation Stage
8. Testing Stage
9. Conclusion and Recommendation

CHAPTER 1

PROJECT PROPOSAL

CHAPTER 1: PROJECT PROPOSAL

1.1 Problem Definition

The problem definition is about developing an application for people who are interested in learning Morse code, like radio amateur groups and boy scouts, to help the user learn it through the practice individually or together, thus increase the effectiveness and engagement of learning for the users. Still, no app has these features yet.

This application includes the following:

- Teaching Morse code using modern techniques (i.e., gamification)
- From Natural language to Morse code translation
- From Morse code to Natural language translation
- Uses a device's flashlight for sending Morse Code.
- Receiving Morse code signals using a camera (Computer Vision)

1.2 Suggested Solution

The suggested solution is developing a smartphone application because of its high accessibility, and it is more user-friendly. Thus, providing a simple and easy way to learn Morse code.

This application is divided into two main modes: Single-User Mode and Multi-User Mode. In Single-User and Multi-User mode, they have two parts: training and exam. The Single-User mode is for one user, and the Multi-User mode allows one or more users to receive messages from a sender through Computer Vision.

The abovementioned points should be implemented using Java for android smartphones.

1.3 Project Objectives and Goals

The goals and objectives for Single-User and Multi-User mode are in the following table: (**Table 1**)

Table 1: Objectives & Goals in the Proposal

Mode	Objectives	Goals
Single-User	• Applying the gamification technique. • Using the flashlight during the learning/testing process.	• Improve the user's knowledge of Morse code effectively.
Multi-User	• Applying gamification technique. • Enabling the collaborative learning technique. • Sender: use the flashlight to send Morse code signals. • Receiver: use the camera (Computer Vision) to receive Morse code signals.	• Improve the knowledge of multiple users in Morse code effective way. • Improve group communication.

CHAPTER 2

DOMAIN ANALYSIS

CHAPTER 2: DOMAIN ANALYSIS

2.1 Scope

The scope of this project includes the following:

- Smartphones with Android platform only due to time limits in learning and development for other platforms like iOS.
- Computer Vision and Deep Learning for recognizing the flashlight signals.

2.2 Schedule for the Project

This schedule section presents the tasks of our Morse code project using Work Breakdown Structure (WBS). Also, it represents the timeline of this project through the Gantt Charts and the Network Diagram.

2.2.1 Work Breakdown Structure (WBS)

In (**Figure 1**), the WBS presents the tasks that will be performed during this and the next semester.

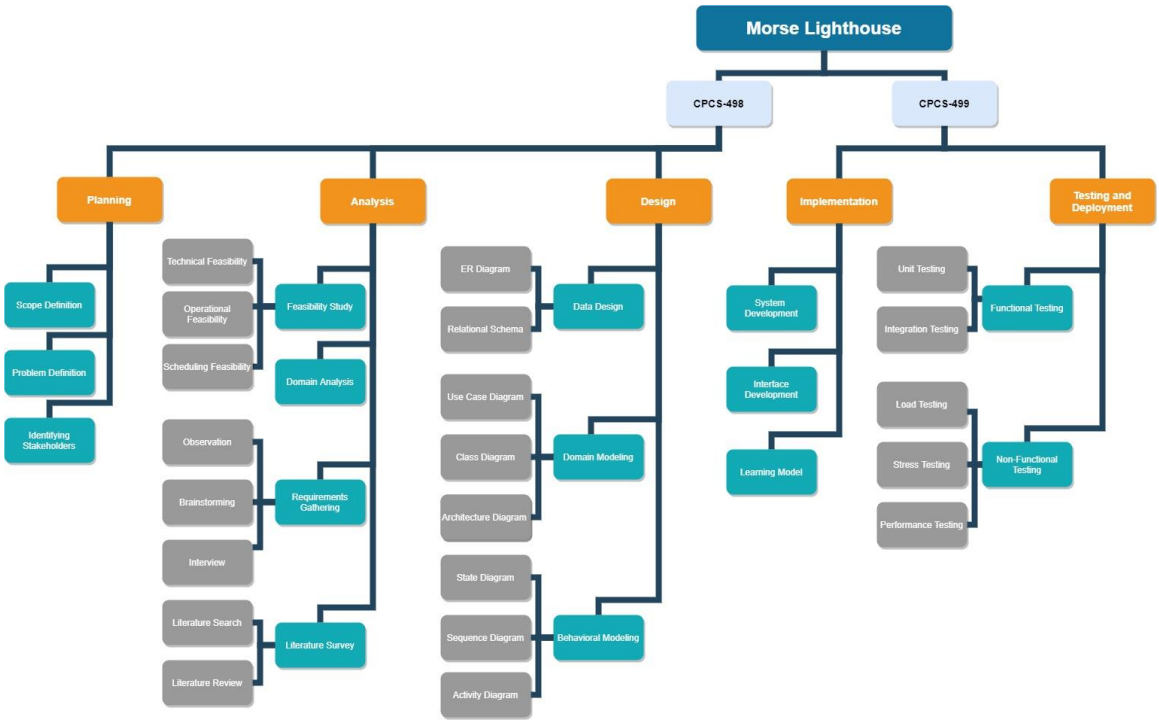


Figure 1: The WBS of Morse Lighthouse

2.2.2 Gantt Chart Diagrams

The Gantt Chart figures for the project's schedule are the following:

1. Full project schedule (**Figure 2,3**)
2. Planning phase (**Figure 4,5**)
3. Analysis phase (**Figure 6,7**)
4. Design phase (**Figure 8,9**)
5. Project's final schedule (**Figure 10,11**)

These figures are on the following pages. (Note that The Gantt Chart of each schedule and phase is divided into two parts, that is, two figures, and each of these two parts is within a separate single page, so it can fit each in this report without any problem)

ID	Task Name	Duration	Start	Finish	Predecessors
1	**1 CPCS-498**	**79 days**	**Sun 30/08/20**	**Wed 16/12/20**	
2	**1.1 Planing**	**16 days**	**Tue 01/09/20**	**Tue 22/09/20**	
6	1.2 <Initial Presentation>	0 days	Tue 22/09/20	Tue 22/09/20	٢
7	**1.3 Analysis**	**20 days**	**Wed 23/09/20**	**Tue 20/10/20**	٦
19	1.4 <Report & presentation 1>	0 days	Tue 20/10/20	Tue 20/10/20	٧
20	**1.5 Design**	**15 days**	**Wed 21/10/20**	**Tue 10/11/20**	١٩
32	1.6 <Report & presentation 2>	0 days	Wed 11/11/20	Wed 11/11/20	٢٠
33	1.7 <Poster Session>	0 days	Fri 13/11/20	Fri 13/11/20	٣٢
34	1.8 <Final report submission>	0 days	Mon 23/11/20	Mon 23/11/20	٣٣
35	**2 CPCS-499**	**67 days**	**Sun 17/01/21**	**Sun 18/04/21**	١
36	**2.1 Implementation**	**40 days**	**Sun 17/01/21**	**Thu 11/03/21**	
39	**2.2 Testing and Maintanance**	**27 days**	**Thu 11/03/21**	**Fri 16/04/21**	٣٦

Figure 2: The Gantt Chart of the full project schedule (part 1)

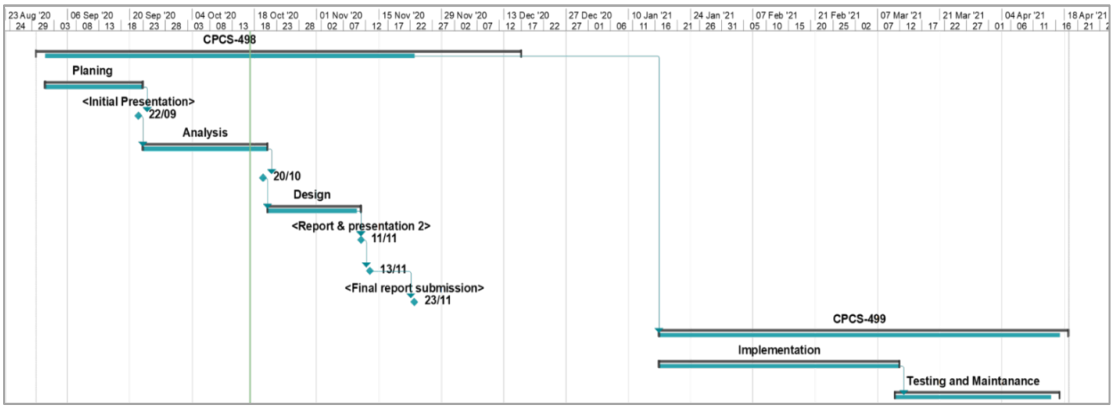


Figure 3: The Gantt Chart of the full project schedule (part 2)

ID	Task Name	Duration	Start	Finish	Predecessors
1	**1 CPCS-498**	**79 days**	**Sun 30/08/20**	**Wed 16/12/20**	
2	**1.1 Planing**	**16 days**	**Tue 01/09/20**	**Tue 22/09/20**	
3	1.1.1 Problem Definition	6 days	Tue 01/09/20	Tue 08/09/20	
4	1.1.2 Scope Definition	6 days	Tue 08/09/20	Tue 15/09/20	٣
5	1.1.3 Identifying Stakeholders	6 days	Tue 15/09/20	Tue 22/09/20	٤
6	1.2 <Initial Presentation>	0 days	Tue 22/09/20	Tue 22/09/20	٢

Figure 4: The Gantt Chart of the planning phase (part 1)

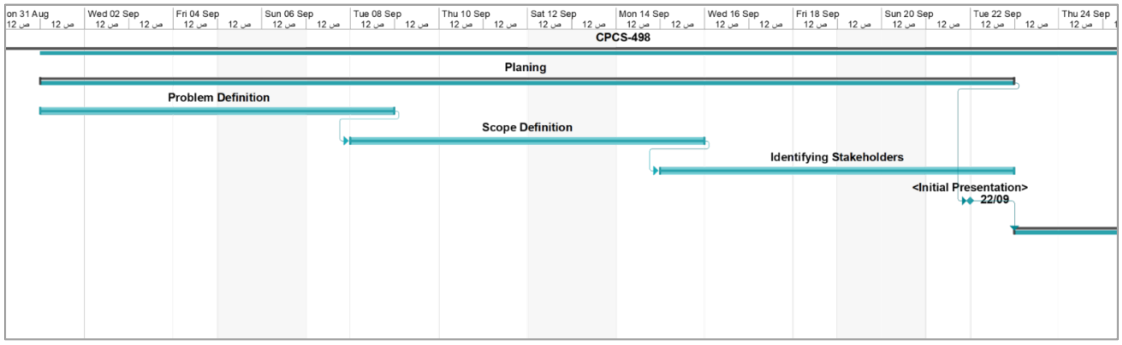


Figure 5: The Gantt Chart of the planning phase (part 2)

7			**1.3 Analysis**	**20 days**	**Wed 23/09/20**	**Tue 20/10/20**	٦
8	✓		**1.3.1 Feasibility Study**	**4 days**	**Wed 23/09/20**	**Mon 28/09/20**	
9	✓		1.3.1.1 Technical Feasibility	4 days	Wed 23/09/20	Mon 28/09/20	
10	✓		1.3.1.2 Operational Feasibility	4 days	Wed 23/09/20	Mon 28/09/20	٩SS
11	✓		1.3.1.3 Scheduling Feasibility	4 days	Wed 23/09/20	Mon 28/09/20	١٠SS
12	✓		**1.3.2 Requirements**	**4 days**	**Tue 29/09/20**	**Fri 02/10/20**	٨
13	✓		1.3.2.1 Observation	2 days	Wed 30/09/20	Thu 01/10/20	
14	✓		1.3.2.2 Brainstorming	0.5 days	Wed 30/09/20	Wed 30/09/20	١٣SS
15	✓		1.3.2.3 Interview	3 days	Wed 30/09/20	Fri 02/10/20	١٤SS
16			**1.3.3 Literature Survey**	**13 days**	**Sun 04/10/20**	**Tue 20/10/20**	١٢
17	✓		1.3.3.1 Literature Search	6 days	Sun 04/10/20	Fri 09/10/20	
18			1.3.3.2 Literature Review	7 days	Fri 09/10/20	Tue 20/10/20	١٧
19	✓		1.4 <Report & presentation 1>	0 days	Tue 20/10/20	Tue 20/10/20	٧

Figure 6: The Gantt Chart of the analysis phase (part 1)

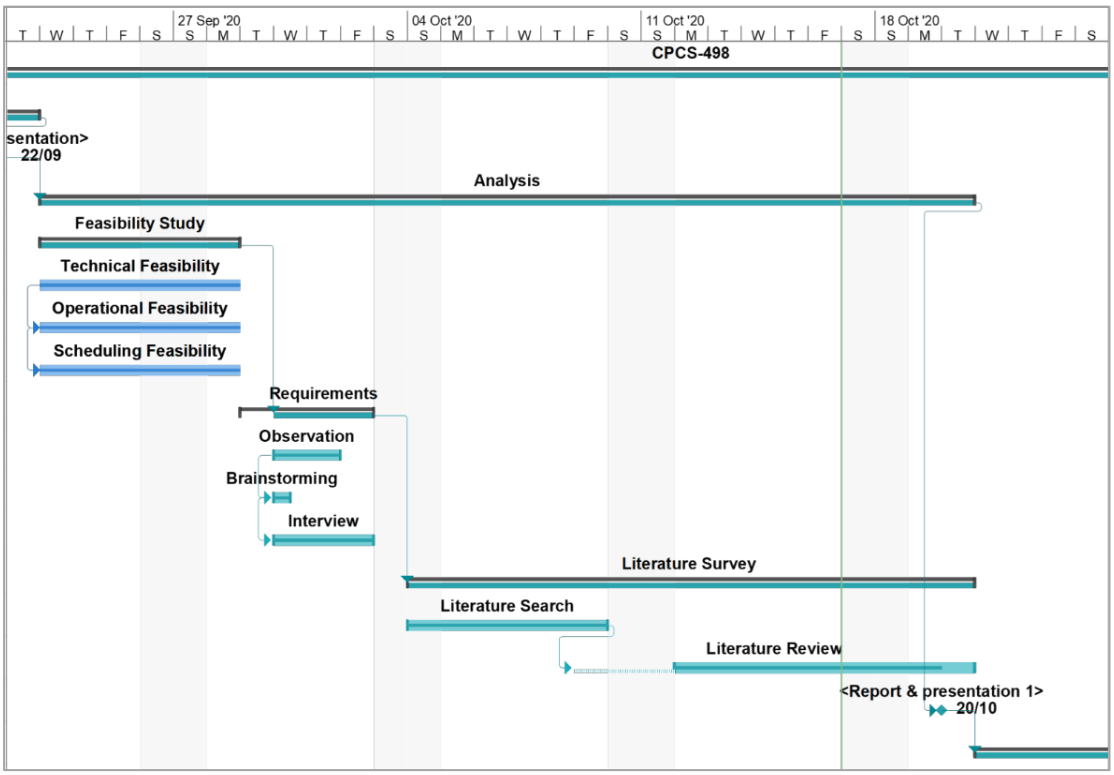


Figure 7: The Gantt Chart of the analysis phase (part 2)

20		**1.5 Design**	**15 days**	**Wed 21/10/20**	**Tue 10/11/20**	١٩
21		**1.5.1 Data Design**	**4 days**	**Wed 21/10/20**	**Mon 26/10/20**	
22		1.5.1.1 ER Diagram	2 days	Wed 21/10/20	Thu 22/10/20	
23		1.5.1.2 Relational Schema	2 days	Fri 23/10/20	Mon 26/10/20	٢٢
24		**1.5.2 Domain Modeling**	**5 days**	**Tue 27/10/20**	**Mon 02/11/20**	٢١
25		1.5.2.1 Use Case Diagram	2 days	Tue 27/10/20	Wed 28/10/20	
26		1.5.2.2 Class Diagram	2 days	Thu 29/10/20	Fri 30/10/20	٢٥
27		1.5.2.3 Architecture Diagram	1 day	Mon 02/11/20	Mon 02/11/20	٢٦
28		**1.5.3 Behavioral Modeling**	**5 days**	**Tue 03/11/20**	**Mon 09/11/20**	٢٤
29		1.5.3.1 State Diagram	1 day	Tue 03/11/20	Tue 03/11/20	
30		1.5.3.2 Sequence Diagram	2 days	Wed 04/11/20	Thu 05/11/20	٢٩
31		1.5.3.3 Activity Diagram	2 days	Fri 06/11/20	Mon 09/11/20	٣٠
32		1.6 <Report & presentation 2>	0 days	Wed 11/11/20	Wed 11/11/20	٢٠
33		1.7 <Poster Session>	0 days	Fri 13/11/20	Fri 13/11/20	٣٢
34		1.8 <Final report submission>	0 days	Mon 23/11/20	Mon 23/11/20	٣٣

Figure 8: The Gantt Chart of the design phase (part 1)

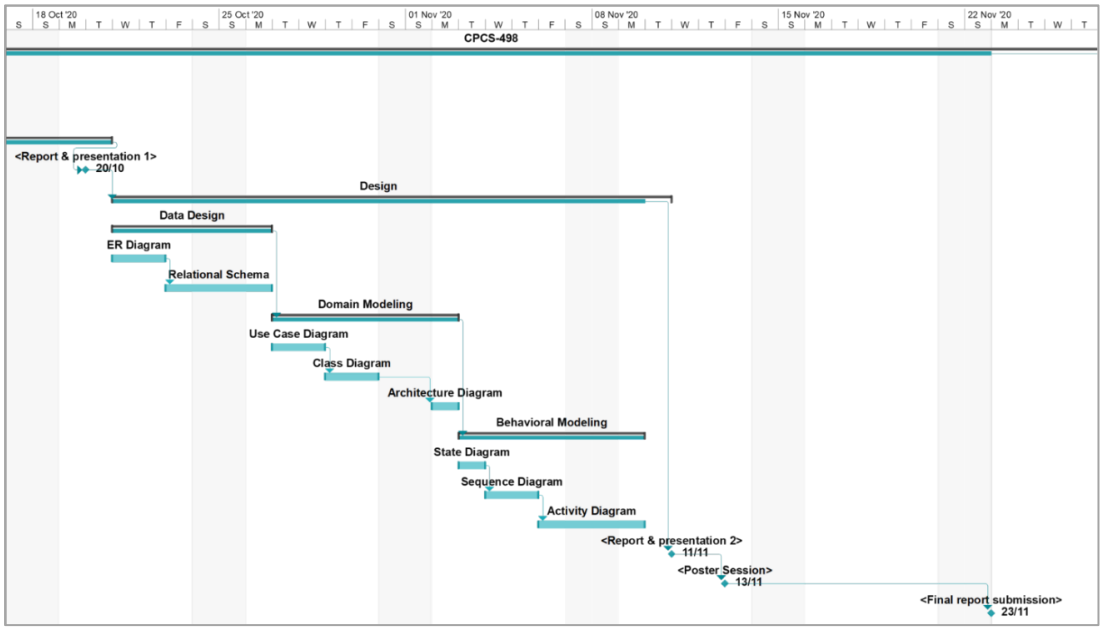


Figure 9: The Gantt Chart of the design phase (part 2)

35	**2 CPCS-499**	**67 days**	**Sun 17/01/21**	**Sun 18/04/21**	١
36	**2.1 Implementation**	**40 days**	**Sun 17/01/21**	**Thu 11/03/21**	
37	2.1.1 Android System	40 days	Sun 17/01/21	Thu 11/03/21	
38	2.1.2 Learning Model	39 days	Mon 18/01/21	Thu 11/03/21	٣٧SS
39	**2.2 Testing and Maintanance**	**27 days**	**Thu 11/03/21**	**Fri 16/04/21**	٣٦
40	**2.2.1 Functional Testing**	**12 days**	**Thu 11/03/21**	**Fri 26/03/21**	
41	2.2.1.1 Unit Testing	6 days	Thu 11/03/21	Thu 18/03/21	
42	2.2.1.2 Integration Testing	6 days	Fri 19/03/21	Fri 26/03/21	٤١
43	**2.2.2 Non-Functional Testing**	**11 days**	**Fri 26/03/21**	**Fri 09/04/21**	٤٠
44	2.2.2.1 Load Testing	4 days	Fri 26/03/21	Wed 31/03/21	
45	2.2.2.2 Stress Testing	3 days	Thu 01/04/21	Mon 05/04/21	٤٤
46	2.2.2.3 Performance Testing	3 days	Tue 06/04/21	Thu 08/04/21	٤٥
47	**2.2.3 Maintanance**	**5 days**	**Sat 10/04/21**	**Thu 15/04/21**	٤٣
48	2.2.3.1 Proactive Maintenance	3 days	Sat 10/04/21	Tue 13/04/21	
49	2.2.3.2 Reactive Maintenance	2 days	Wed 14/04/21	Thu 15/04/21	٤٨

Figure 10: The Gantt Chart of the CPCS-499 initial schedule (part 1)

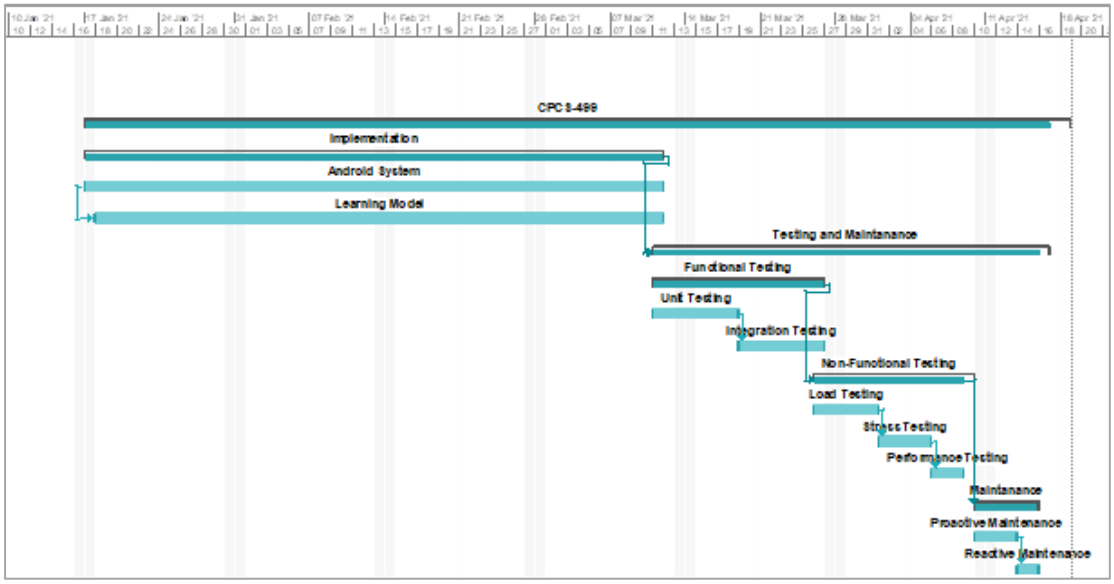


Figure 11: The Gantt Chart of the Project's final schedule (part 2)

2.2.3 Network Diagram

The following network diagram, **Figure 12**, represents how the tasks of this project are going to be conducted in sequence. (Some information has been omitted from this diagram)

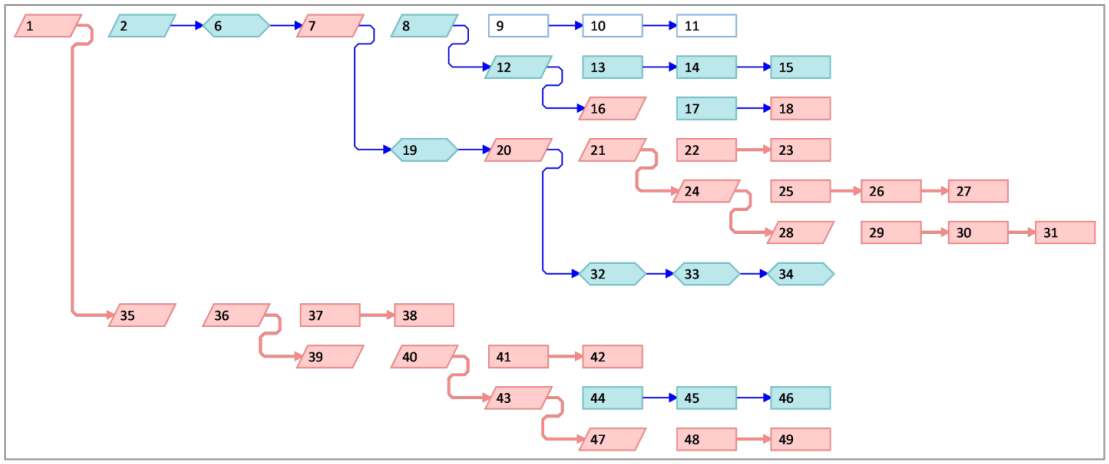


Figure 12: The Network Diagram of the project schedule

2.3 System Analysis

Many existing applications have been analyzed that perform similar activities, and the good ones from Google Play Store are "Morse Code Agent" [3] and "Morse Code: Learn and Play" [4]. “Morse Code Agent" uses a camera or sound recognition to decode Morse code signals and sends Morse code through flashlight signals. The other one, “Morse Code: Learn and Play" uses the gamification technique for learning Morse code. The following table (**Table 2**) shows the observations.

Table 2: Domain Analysis for "Morse Code Agent" and "Morse Code: Learn and Play"

Facet	Morse Code Agent	Morse Code: Learn and Play
Type of Business	• Learning uses	• Communication uses
Parts of the business	• Communication Service. • Translation service.	• Learning service. • Translation service.
How does it operate	1. Morse Code Interpreter: a. Receive Morse Code from light. b. Translate Morse Code into a string. 2. Morse Code Sender: a. Type in the specified string b. Convert string into Morse Code c. Sends the Code using light signals	1. Input using the (.) and (-) of the Morse code input by a long and short touch of the screen, respectively. 2. Guessing the natural language letter or Morse code.
Business's System environment	Android Smartphone Operating System	
Problems with the current system	• Does not provide any learning techniques	• Does not provide collaborative learning (more than one user)

The idea of merging these two apps into one app with the addition of a collaborative learning feature is going to be in Morse Lighthouse project.

CHAPTER 3

DATA COLLECTION AND ANALYSIS TECHNIQUES

CHAPTER 3: DATA COLLECTION AND ANALYSIS TECHNIQUES

3.1 Data Collection Techniques

The following table illustrates the techniques of the used information gathering (Questionnaire and Document Analysis) and their strengths and weaknesses. (**Table 3**)

Table 3: The Used Information Gathering Techniques

Technique	Weaknesses	Strengths
Questionnaire	• Participants can be dishonest in their answers. • Some questions may be difficult to analyze. • The Scope of the question is too broad and does not provide detailed information.	• Large numbers of people. • Provide both information and opinions. • No time limits. • Cost-effective. • Easy to create and deliver.
Document Analysis	• Document author can provide us with biased information. • Documents often have information that is not relevant to our project. • Can be time-consuming to look through it to find information.	• Cost efficient. • Document's data is easily accessible and reliable as well. • Offers functional information. • Very useful if we do not have access to the stakeholders.

3.2 The Needed Skills

Achieving the objectives and goals of the project needs some required skills, and these are the following:

- Morse code
- Deep Learning
- Computer Vision
- Learning programming in Android Studio
- Develop an AI learning model.

The following table explains the reason and the procedures for each mentioned skill. (**Table 4**)

Table 4: The Needed Skills for the Project

Skill	Reason	Procedures
Morse code	Morse code is the main topic in this project	• Websites for Morse code learning • Morse code learning apps
Deep Learning	To ensure that we obtain high accuracy for real-time Morse code detection	• CPCS-433 • Online courses for Deep Learning
Computer Vision	To get a visual-type signal of Morse code	• Online courses for Computer Vision • YouTube
Learning programming in Android Studio	We are planning to develop this application on a smartphone Android system	• YouTube • Official Documentation
Data Science	To have the ability to analyze Morse code signals	• YouTube • Online courses

3.3 Literature Review

The following tables (**Table 5, 6, 7, 8, 9**) summarized the selected five papers that may contribute to the Morse Lighthouse project. The contributions are the following:

- The learning process techniques
- The application of Machine Learning for this project

Table 5: "Using Gamification and Serious Games for English Language Learning"

Summary	
Publishing Year	2019
Publisher	IEEE
Title of the paper	"Using Gamification and Serious Games for English Language Learning"
Problem	How to understand in a better way and research further about factors that influence the optimal utilization of gamification and serious games to learn the English language.
Hypothesis or Research Questions	The research question is on what might be inhibiting or enabling gamification and serious games as learning tools for the English language.
AIM of the study	The aim of the author in this research is to classify the prominent opportunities and challenges that come with using gamification and games for English language learning.
Methods Used	The method that the author uses is a SWOT analysis from a brainstorming session as a preliminary exploratory study with students at the College of Computer and Information Science.

Continue…

…Continuation	
Results	The result is that there is a need for an exciting and motivating learning method consequently, this will diminish stress and help to increase students' responsiveness. This kind of learning ambiance improves their concentration and stimulate the instinct for achievement, according to the participants.
Limitations	1- Students highlighted that they are not very comfortable with ludic environments. 2-Students may feel bored, or distracted, or frustrated if they misunderstand the game. 3- Could produce discipline issues and provoke lack of control 4- Could be perceived as childish, immature, unnecessary, or even a waste of time. 5- Could miss the main purpose of the game.
Future Work	The author suggested that future work is conducting a quantitative study examining the relationship between the success key factors of the use of ludic learning materials and students' performance.
Contribution for the Project	For this project, the paper will be used as a reference in implementing gamification techniques to teach Morse code.

Source: [4]

Table 6: "Efficient Algorithm for Blinking LED Detection Dedicated to Embedded Systems Equipped with High Performance Cameras"

Summary	
Publishing Year	2015
Publisher	IEEE
Title of the paper	"Efficient Algorithm for Blinking LED Detection Dedicated to Embedded Systems Equipped with High Performance Cameras"
Problem	Implementing blinking LED or similar sources detecting algorithm that is efficient in terms of speed.
Hypothesis or Research Questions	How to detect LED blinking light with high performance in embedded systems.
AIM of the study	This paper aims to show the concept of an efficient algorithm for the detection of blinking LED or similar signal sources, and its implementation in embedded devices with high-performance cameras.
Methods Used	The authors use the quantitative method to collect data (through AVT algorithm) using a camera.
Results	It has been shown that the proposed algorithm reduces the computational power necessary for blinking LED detection especially when the amount of data to be processed is high.
Limitations	The reduction of computational time depends on the number of changes in the scene, but compared to the referenced algorithm, the reduction is significant.
Future Work	Not Present
Contribution for the Project	We can use this paper as a reference in implementing the algorithm for the Multi-User mode of the application.

Source: [5]

Table 7: "Towards Haptic Learning on a Smartwatch"

Summary	
Publishing Year	2018
Publisher	Association for Computing Machinery
Title of the paper	"Towards Haptic Learning on a Smartwatch"
Problem	It is not clear if haptic learning can be used with the smartwatch actuators.
Hypothesis or Research Questions	Can a smartwatch teach a new skill with its haptic elements?
AIM of the study	Using a smartwatch for passive tactile learning.
Methods Used	Six participants participated in a between-subjects study about learning from tactile stimulation on a smartwatch.
Results	The users seemed to be better at the learning process of Morse code through the instructional stimuli.
Limitations	Learning using this technique may not have long-lasting effects
Future Work	Explore the longevity of this learning method
Contribution for the Project	The technique can be utilized in the Single-User mode of the application.

Source: [6]

Table 8: "Research on Automatic Decoding of Morse Code Based on Deep Learning"

Summary	
Publishing Year	2019
Publisher	IEEE
Title of the paper	"Research on Automatic Decoding of Morse Code Based on Deep Learning"
Problem	The current automatic decoding method of the Morse telegram has limited accuracy, and cannot adapt to signal distortion and code length deviation of the manual telegram
Hypothesis or Research Questions	Experimental results show that the decoding system has strong adaptability to manual deviation and frequency drift and is robust in a noisy environment.
AIM of the study	This paper introduces the deep learning method and constructs an automatic decoding model, which integrates feature extraction, sequence modeling, and transcription into an end-to-end training neural network. The time-frequency diagrams signals are used for training and testing.
Methods Used	The Convolutional Recurrent Neural Network (CRNN) combining Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) is proposed to realize image sequence-based character recognition. Inspired by this, we used this structure and combined it with the Connectionist Temporal Classification (CTC) method to solve the alignment between labels and outputs to identify the time-frequency diagram of the Morse code.

Continue…

…Continuation	
Results	The result is that the neural network model proposed in this paper has a significantly better recognition effect. Especially in the case of low SNR, our algorithm can still maintain good recognition accuracy, while others decline seriously.
Limitations	The algorithm is only tested for audio datasets and not visual - light signals.
Future Work	Not Present
Contribution for the Project	Using the CRNN described in the paper to improve the accuracy of Morse code detection.

Source: [7]

Table 9: "An Automatic Decoding Method for Morse Signal based on Clustering Algorithm"

Summary	
Publishing Year	2016
Publisher	Springer
Title of the paper	"An Automatic Decoding Method for Morse Signal based on Clustering Algorithm"
Problem	Low accuracy of automatic decoding for Morse signals.
Hypothesis or Research Questions	What are the technical problems of shortwave Morse telegraph automatic decoding, especially the impact SNR and unstable typing of Morse code, and how to solve it? The K-means clustering method-based decoding algorithm is proposed, which is based on the existing Morse code decoding method.
AIM of the study	Achieving automatic identification and decoding of mechanical and manual Morse code using Machine learning methods, which restore the Morse code message in a meaningful way. (numbers, letters, or symbols)
Methods Used	k-means clustering
Results	The algorithm can adapt to poor noise environments. For the SNR of the signal higher than -10dB, it could get the right result. In the simulation experiments, it is easy to know that the decoding's effect has a downward trend when the SNR decreases.
Limitations	Not Present
Future Work	Know the way of optimizing the clustering algorithm to improve the robustness of clustering.
Contribution for the Project	This algorithm in this paper can be used in the project

Source: [8]

The following tables (**Table 10, 11, 12, 13, 14**) is the PROMPT Criteria for the abovementioned selected documents.

Table 10: PROMPT Criteria for "Using Gamification and Serious Games for English Language Learning"

Document	PROMPT Criteria
"Using Gamification and Serious Games for English Language Learning"	• **Presentation:** The information is clearly communicated, and it is not difficult to read. It was written for academic audiences. You do not need to additional knowledge to understand it. • **Relevance:** Yes, it is relevance to our needs. It discusses the methodology of the gamification learning technique. This technique can be adapted in the project. • **Objectivity:** The writer was objective about this issue in the research. • **Method:** SWOT analysis and Delphi method • **Provenance:** Authors: Nacim YANES, Ikram BOUOUD. Organization: ResearchGate • **Timeliness:** It is up to date, as it is published in 2019.

Table 11: PROMPT Criteria for "Efficient Algorithm for Blinking LED Detection Dedicated to Embedded Systems Equipped with High Performance Cameras"

Document	PROMPT Criteria
"Efficient Algorithm for Blinking LED Detection Dedicated to Embedded Systems Equipped with High Performance Cameras"	• **Presentation:** The information is sufficiently clear in its words, the structure, the figures, and the layout, and the way the author wrote the algorithms so it can be read by any people, so the readability is good. It was written for academic audiences who looking to improve healthcare, safety systems, and assets management. • **Relevance:** It is relevant to the computer vision part of our project. The beginning of the paper was about making a more efficient LED blinking detection algorithm in terms of performance for embedded devices, and we need that for making a more efficient Morse code light signals detection algorithm, and this paper is a good resource for the project. • **Objectivity:** It was objective in the research's issue. The writers wrote the theories, suggested the solution, and tested and wrote his findings from the result. • **Method:** The authors use the quantitative method to collect data in a scene through a camera using the AVT algorithm, which is the right way to receive this kind of data. • **Provenance:** Three Authors: Michal Tarkowski, Lukasz Kulas, and Przemyslaw Woznica. Organization: ResearchGate. It was published in: https://www.researchgate.net/publication/308846394 • **Timeliness:** It is per to our requirements; it was published in 2015

Table 12: PROMPT Criteria for "Towards Haptic Learning on a Smartwatch"

Document	PROMPT Criteria
"Towards Haptic Learning on a Smartwatch"	• **Presentation:** The information was clear, and it was not difficult to read. While this document was written for academic purposes, it was not difficult to follow and did not require prior knowledge on the topic. • **Relevance:** The information that this document provides is excellent, and it is helpful in our project, as it discusses the benefits of haptic learning on smartwatches, this technique can be adapted in the project. • **Objectivity:** The author stays objective of the whole document, and he does not give any opinions and provides only his findings. • **Method:** They recruited 6 participants to participate in a between-subjects study about learning from tactile stimulation on a smartwatch. • **Provenance:** Authors: Caitlyn Seim, Rodrigo Pontes, Sanjana Kadiveti, Zaeem Adamjee, Annette Cochran, Timothy Aveni, Peter Presti, Thad Starner. Publisher: Association for Computing Machinery. • **Timeliness:** The document is up to date, as it was published in October 2018.

Table 13: PROMPT Criteria for "Research on Automatic Decoding of Morse Code Based on Deep Learning"

Document	PROMPT Criteria
"Research on Automatic Decoding of Morse Code Based on Deep Learning"	• **Presentation:** The information is communicated, organized and ease to read, the readers must have pre-knowledge of the Machine Learning, Neural Network and it is written for academic audience. • **Relevance:** Yes, it is relevance to the project's needs. It discusses an efficient method to decode Morse Code using deep learning approach and this learning model is required in the Multi-User mode of the app. • **Objectivity:** The author interested in enhancing the accuracy of automatic recognition of Morse code using a novel approach, the state of the author position on the issue is that traditional methods have high requirements on signal quality and are difficult to adapt to the actual environment, and there is no vested interest. • **Method:** Yes, the author used universal dataset for Morse code recognition, the method was appropriate, and data can be trusted. • **Provenance:** It was clear where the information came from, there are two authors, “Weihao L from Information Engineering University, Zhengzhou, China” and “Keren Wang from National Key Laboratory of Science and Technology on Blind Signal Processing, Chengdu, China”, it was published by IEEE which is a reputable source. • **Timeliness:** The information is communicated, organized and ease to read, the readers must have pre-knowledge of the Machine Learning, Neural Network and it is written for academic audience.

Table 14: PROMPT Criteria for "An Automatic Decoding Method for Morse Signal based on Clustering Algorithm"

Document	PROMPT Criteria
"An Automatic Decoding Method for Morse Signal based on Clustering Algorithm"	• **Presentation:** The language used in the paper is clear, and the tables and figures made sense, no additional knowledge was needed to understand the paper. The audiences are academics. • **Relevance:** The information in the paper matched the project's needs because the algorithm can decode Morse code, and that what the paper is about. • **Objectivity:** The author is based on a university so there is no apparent bias in their view. Furthermore, their findings are about an algorithm that is easier to peer review. • **Method:** The method used was a K-means clustering so there is little doubt in the method they used. • **Provenance:** The authors are from the Beijing Laboratory of Advanced Information Networks, Beijing, China, College of Electronic Information and Control Engineering, Beijing University of Technology, Beijing, and China. • **Timeliness:** It is up to date since it is from 2016

3.4 Questionnaire Analysis

This section discusses the reasons for asking the questions (see **APPENDIX A**) and what to learn from them. The questionnaire methodology was used to collect information from the public, which is related to the Morse Lighthouse Project. The questionnaire was through the Google Forms platform. The number of people who have participated in this questionnaire is (286). The results and reasons behind each question are in the following table: (**Table 15**)

Table 15: The Reasons and Results of Each Question

Question Number	Reasons	Outcomes
1	To know the participant's demographic. The popularity of Morse code Between Males and Females.	• 78.85% Male • 21.15% Female
2	To know the participant's educational background. The popularity of Morse code between the people with different educational background.	• 5.2% Higher Education • 80.35% Bachelor's Degree • 10.8% High School • 3.65% Middle School
3	To know the participant's opinion about traditional learning methodology. The level of satisfaction about traditional learning methodology.	Most of the participants are neutral about traditional learning technique
4	To know the participant's experience with smartphone learning applications. The familiarity of users with learning application.	More than half of the participants use smartphone learning applications
5	To know what kind of application the participants have used. To get inspired by ideas from other application.	• Blackboard • Microsoft Team • Madrasati • Duolingo • Kahoot!

Continue…

		…Continuation
6	To know the participant's demand for smartphones educational applications. If the smartphone is a suitable environment for our application.	Most of participants are satisfied about smartphone learning applications
7	To know if the participants are aware of Morse code. the popularity of Morse code in general.	More than half of the participants are not aware of Morse code
8	To know about the level of knowledge of Morse among the participants. helps in identifying suitable level of proficiency for our application.	Most of participants do not have a practical knowledge in Morse code
9	To know if the participants are interested in Morse code. facilitate the learning of Morse code for those who are interested.	Almost half of the participants are interested in Morse code
10	To know if the participant is more likely to use Morse Lighthouse App	More than half of the participants are candidate to use Morse code

CHAPTER 4

REQUIREMENTS SPECIFICATION

CHAPTER 4: REQUIREMENTS SPECIFICATION

This section lists the requirements for each type, including:

- Functional Requirements
- Non-Functional Requirements
- Data Requirements
- Hardware Requirements

In addition to these core requirements, the extended requirements, which are identified by the (*) symbol, are included.

To gather these requirements, the Use-Case Diagram will be needed, which describes the interaction between users and the system. The included users are:

- User in general (**Figure 13**)
- Trainee (**Figure 14**)
- Spectator (**Figure 15**)
- Participant (**Figure 16**)
- Sender (**Figure 17**)
- Receiver (**Figure 18**)

The Use Case tables are in **APPENDIX B**.

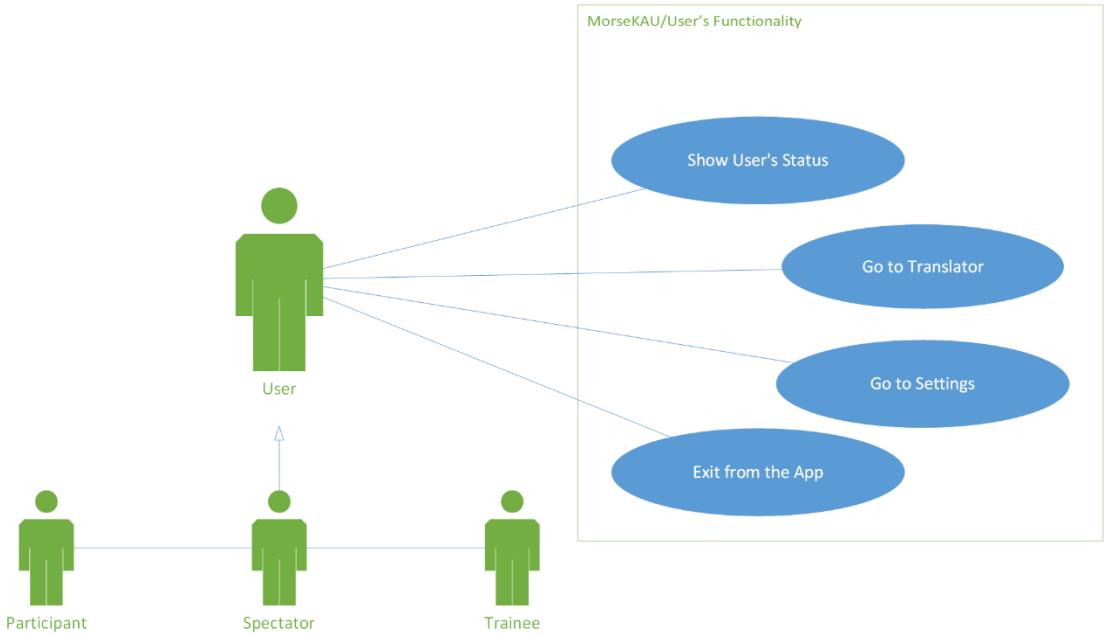


Figure 13: Use-Case Diagram for User's Functionality

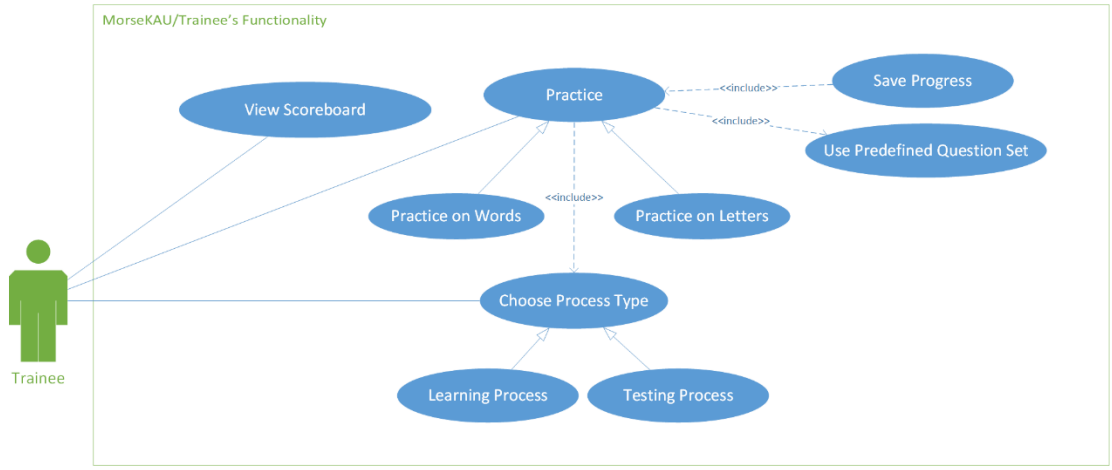


Figure 14: Use-Case Diagram for Trainee's Functionality

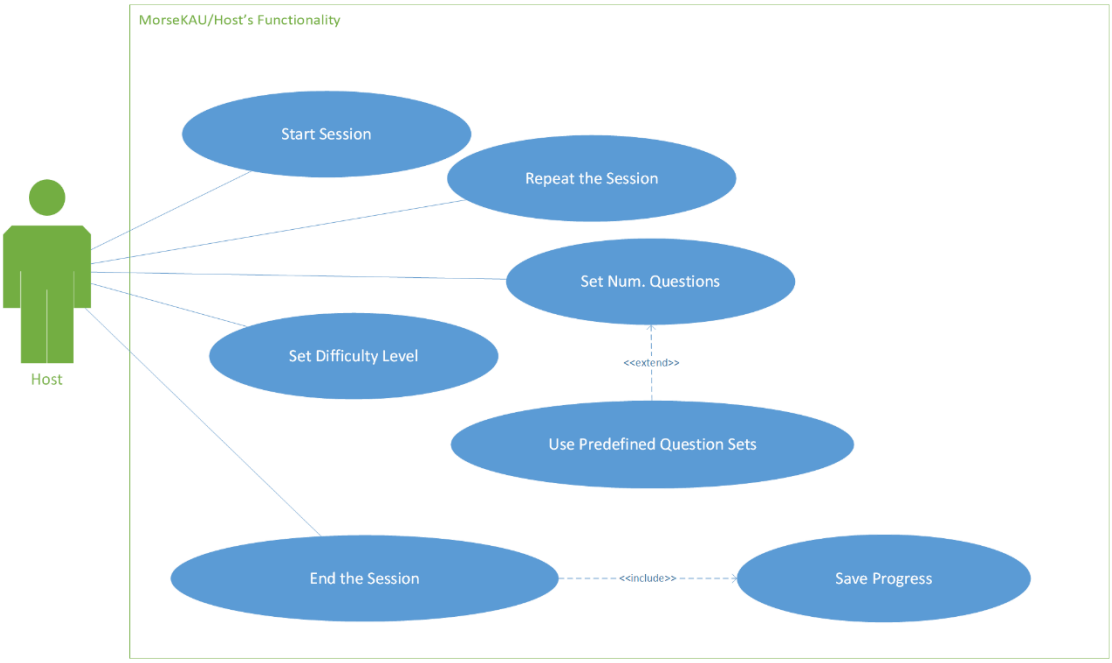


Figure 15: Use-Case Diagram for Host's Functionality

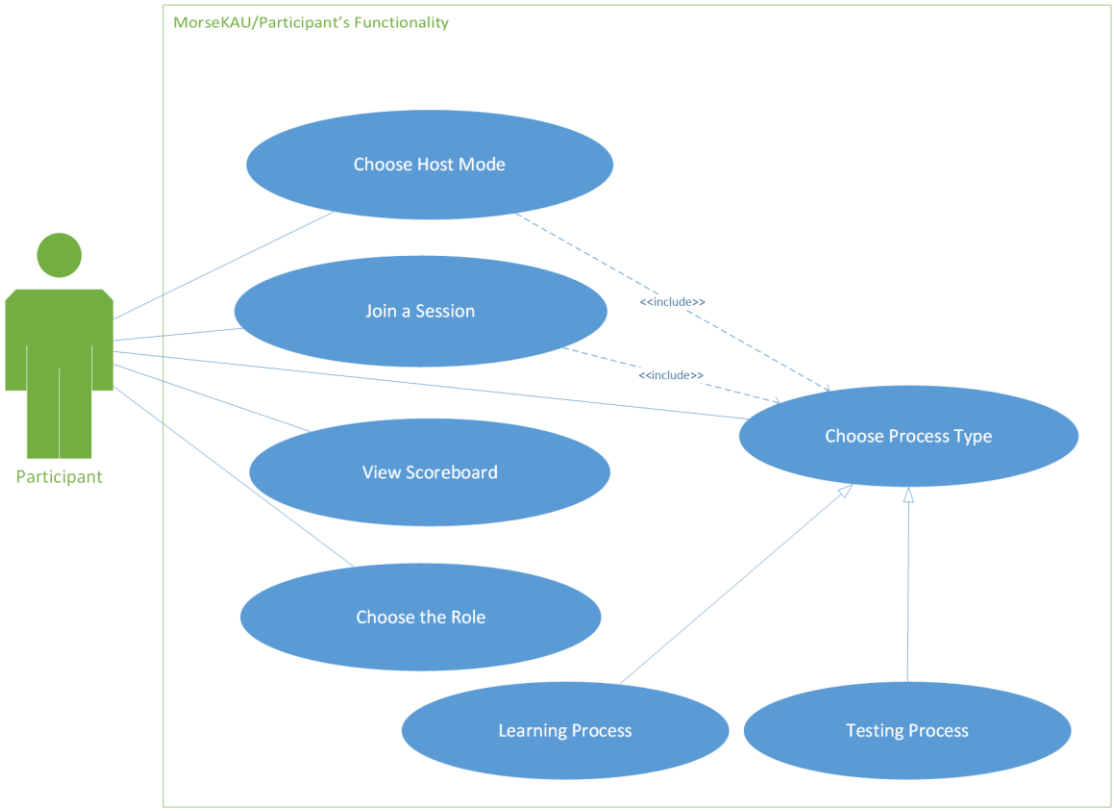


Figure 16: Use-Case Diagram for Participant's Functionality

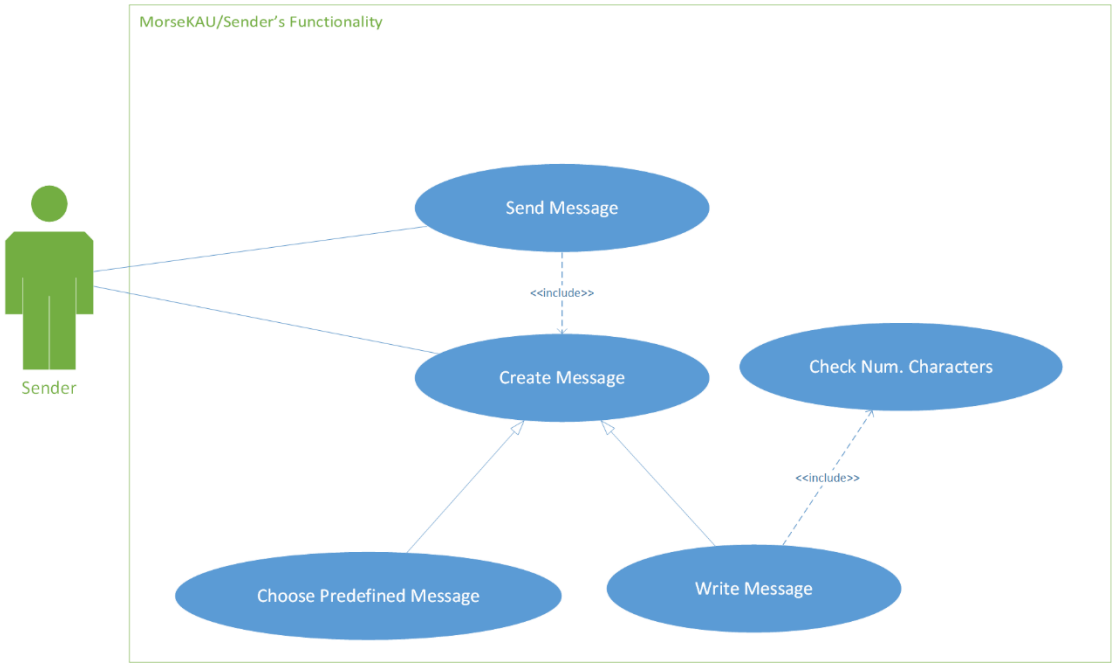


Figure 17: Use-Case Diagram for Sender's Functionality

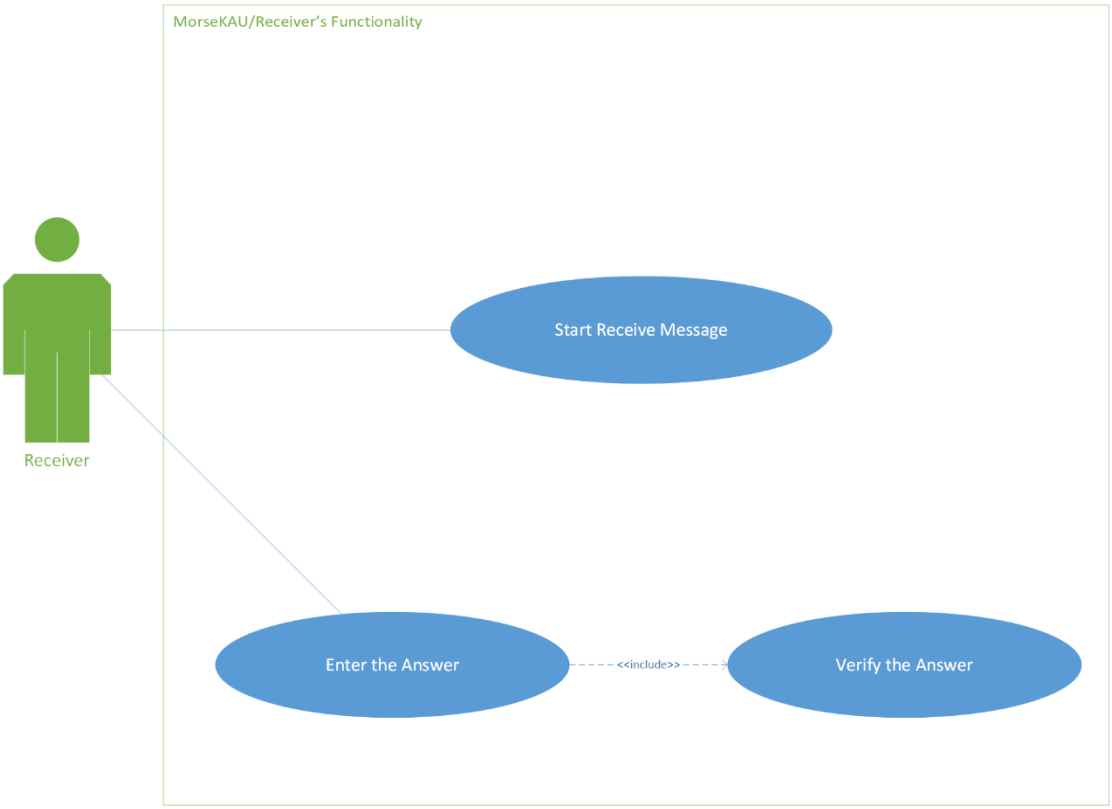


Figure 18: Use-Case Diagram for Receiver's Functionality

4.1 Functional Requirements

All users shall:

- FRU1. Enter natural language text or Morse code (dot/s and/or dash/s).
- FRU2. View his/her statistics.
- FRU3. View his/her session history.
- FRU4. View list of natural language letters to Morse code.
- FRU5. Change his/her username.

Morse Lighthouse shall:

- FRM1. Translate between Morse code text and natural language text using a translator.
- FRM2. Use a flashlight for Morse code generation.
- FRM3. Use the Gamification Learning technique.
- FRM4. Have a predefined question set. *
- FRM5. Have a training and exam process for Multi-User mode.

- FRM6. Have a training and exam process for Single-User. *
- FRM7. Have Single-User mode. *
    - FRM7.1. Use vibration for Morse code generation. *
    - FRM7.2. Use the Haptic Learning technique. *
    - FRM7.3. The trainee shall practice through the workshop. *
- FRM8. Have Multi-User mode.
    - FRM8.1. Only one user can be a host.

      The host shall:
    - FRM8.2. Create a session.
        - FRM8.2.1. The session has 2 to 5 users.
        - FRM8.2.2. The session has 2 to 10 questions.
    - FRM8.3. Start a session.
    - FRM8.4. Determine the difficulty level of questions e.g., Letters or Words. *
    - FRM8.5. Able to repeat the session.
    - FRM8.6. Able to end the session.

      All participants shall:
    - FRM8.7. Join the session through the IP Address.
    - FRM8.8. View the scoreboard at any time throughout the session.

      The Sender shall:
    - FRM8.9. Enter the message content for the question that will be sent.
        - FRM8.9.1. The number of characters must not exceed 20 characters.
    - FRM8.10 Send the question.

      All Receivers shall:
    - FRM8.11. Use the smartphone's camera (Computer Vision) to receive Morse code light signals.
    - FRM8.12. Enter the answer corresponding to the Morse signals.
- FRM9. Verify the correctness of the user's answer.
- FRM10. Record the user's performance.

4.2 Non-Functional Requirements

Morse Lighthouse shall:

- NFR1. Decoding speed for Morse code light signals should not exceed 3.5 seconds.
- NFR2. Minimum OS support android version 6.0 (Marshmallow)
- NFR3. Requires the following permissions:
    - NFR3.1. Camera Access
    - NFR3.2. Internet Access
    - NFR3.3. Wi-Fi State Access
- NFR3. Conduct a session using Wi-Fi.
- NFR4. Conduct a session using Bluetooth. *
- NFR5. Be compatible with Android smartphones.
- NFR6. Localized to the English language.
- NFR7. Localized to the Arabic language. *

4.3 Data Requirements

The Deep Learning technique required a big dataset, and there is almost no dataset appropriate to the project. Therefore, planning to create a new dataset to apply the machine learning technique is needed.

4.4 Hardware Requirements

For the minimum requirements for the Morse Lighthouse system:

- Processors:
    - Qualcomm Snapdragon Series 4 (2.0 GHz)
    - Exynos 7580 (1.6GHz)
- 2GB Ram
- 2MP

For the recommended requirements for the Morse Lighthouse system:

- Processors:
    - Qualcomm Snapdragon Series 8 (2.8GHz)
    - Exynos 9810 (2.9GHz)
- 4GB Ram
- 5MP

4.5 Equipment or Software Use

The equipment or software that is going to be included in this project are in the following categories:

- For management Software:
    - Microsoft Project
    - GitHub
    - Google Drive
- For communication Software:
    - Discord
    - WhatsApp
    - Zoom
- For design Software:
    - MS Visio
    - Draw.io
- For the implementation Software:
    - Android Studio (using Java or Kotlin)
    - PyCharm or Spyder (Python)
    - Google Cloud Platform
- And lastly for the libraries:
    - TensorFlow (Deep Learning)
    - OpenCV (Computer Vision)

These are the software that is going to be used.

CHAPTER 5

ANALYSIS STAGE

CHAPTER 5: ANALYSIS STAGE

The first graph type related to the analysis stage is the Flowchart. The following figures are an abstract view of how this system works.

Figure 19 represents the flow of the application in general, which starts by prompting the user to select Single-User mode or Multi-User mode, then it asks the user to choose whether he/she wants learning or testing. After that, the process (Learning or Testing) starts. When the user wants to stop the process, the system saves his/her progress in the states so the user can see his performance in learning or testing.

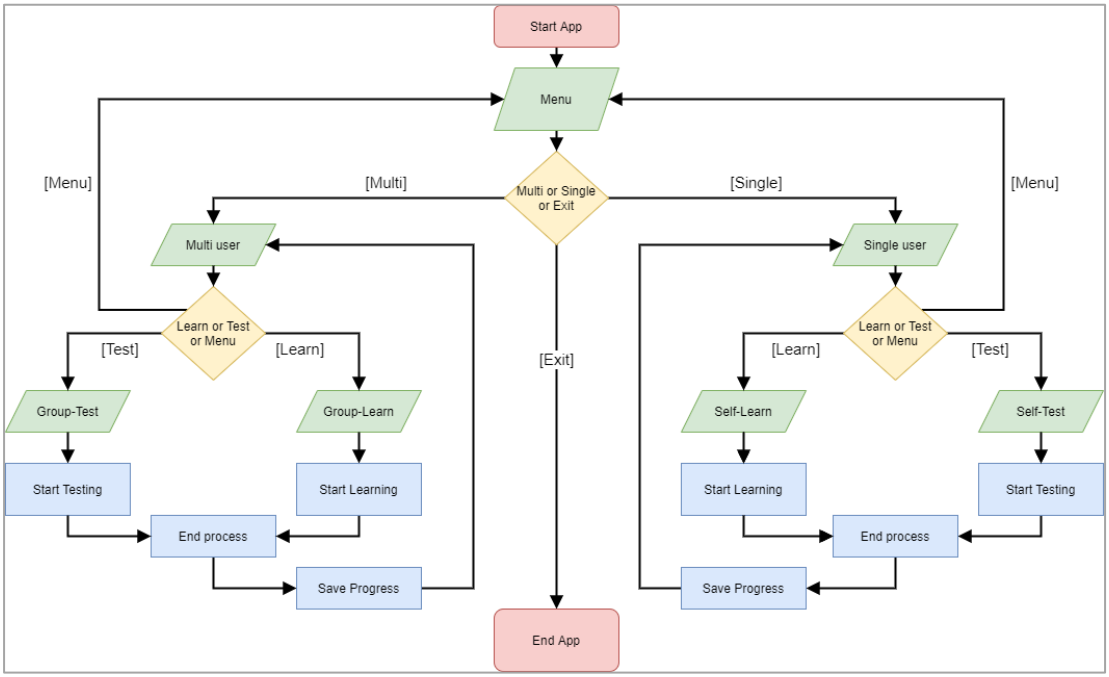


Figure 19: The flowchart of the application in general

The second one (**Figure 20**) represents the final model for learning and testing processes. There are two parallel scenarios: the sender for sending the signals through flashlight, and the receiver for receiving signals using the camera.

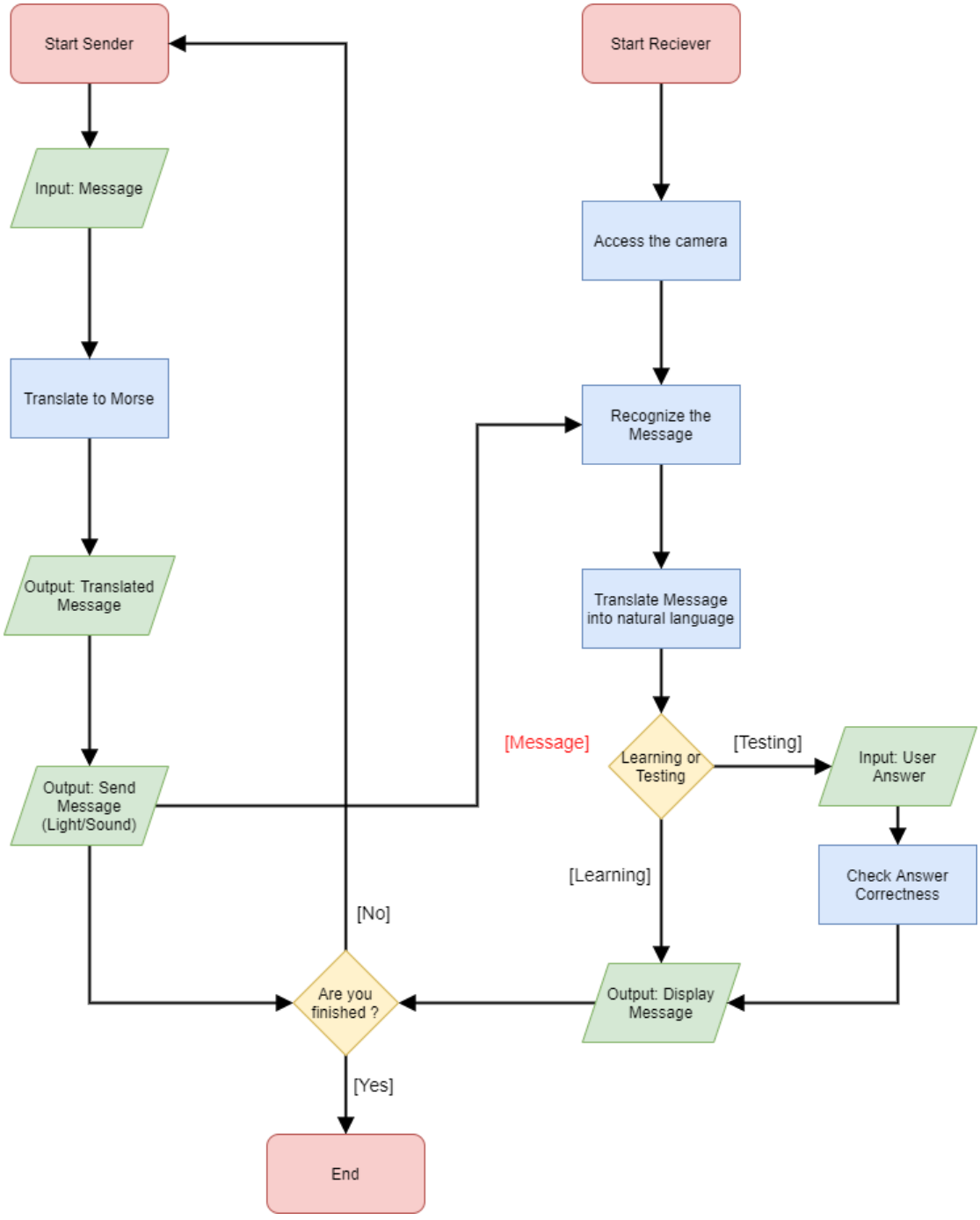


Figure 20: The flowchart of the initial model for learning and testing processes

CHAPTER 6

DESIGN STAGE

CHAPTER 6: DESIGN STAGE

In the design section, there are multiple types of diagrams for Morse Lighthouse system. The first one (**Figure 21**) is the Class Diagram, which shows the static view of the system's entities with their relationships.

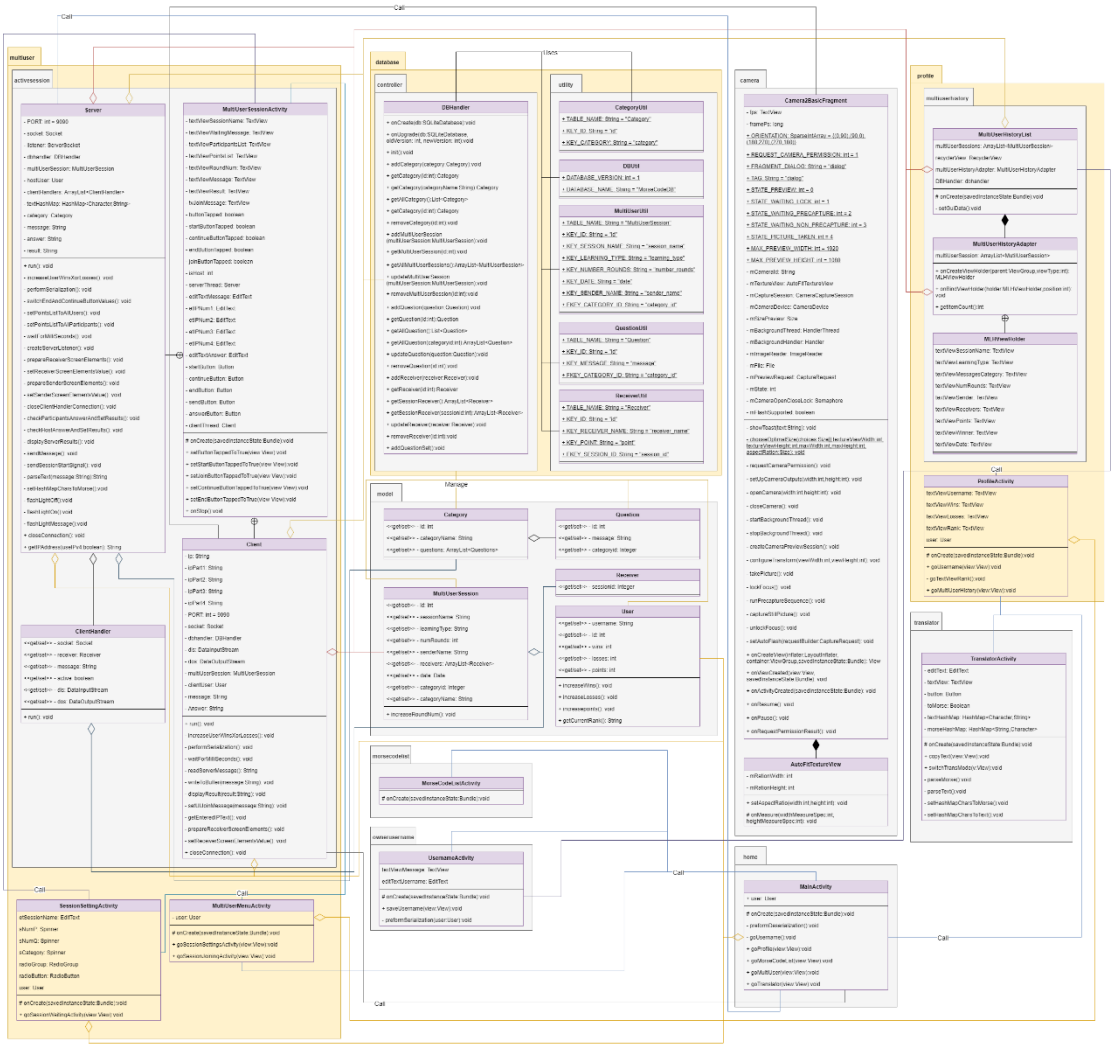


Figure 21: Morse Lighthouse's Class Diagram

The second one is under the behavior modeling, called the Sequence Diagram. This diagram shows how objects interact with each other in a use-case. The used type in **Figure 22** is *Design*.

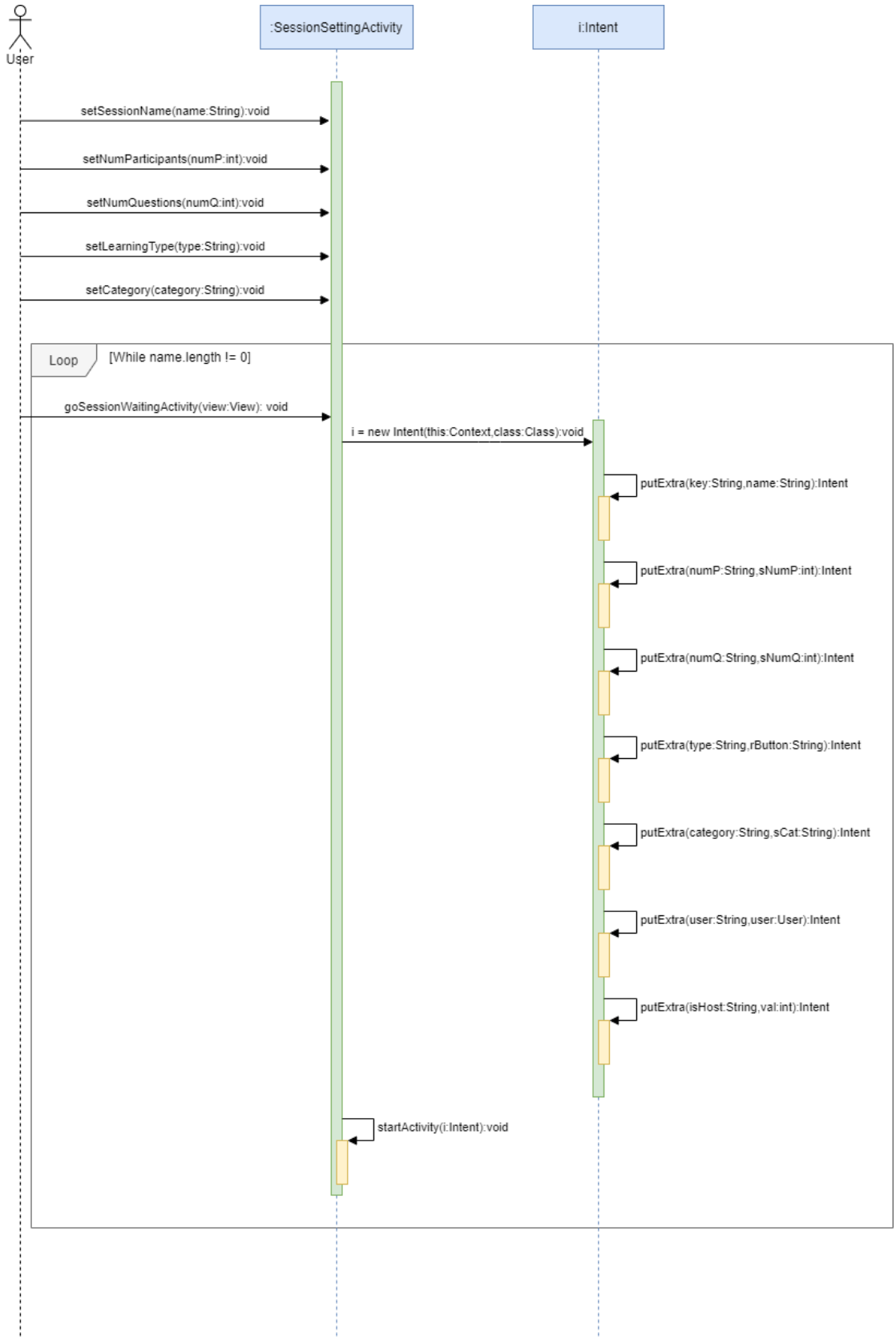


Figure 22: Morse Lighthouse's Sequence Diagram (Design)

The third one is related to the database, called Entity-Relationship Diagram (ER), which illustrates the abstract view of the entities in the app (including Morse Lighthouse, Users with its types, …), and their relationships. (**Figure 23**)

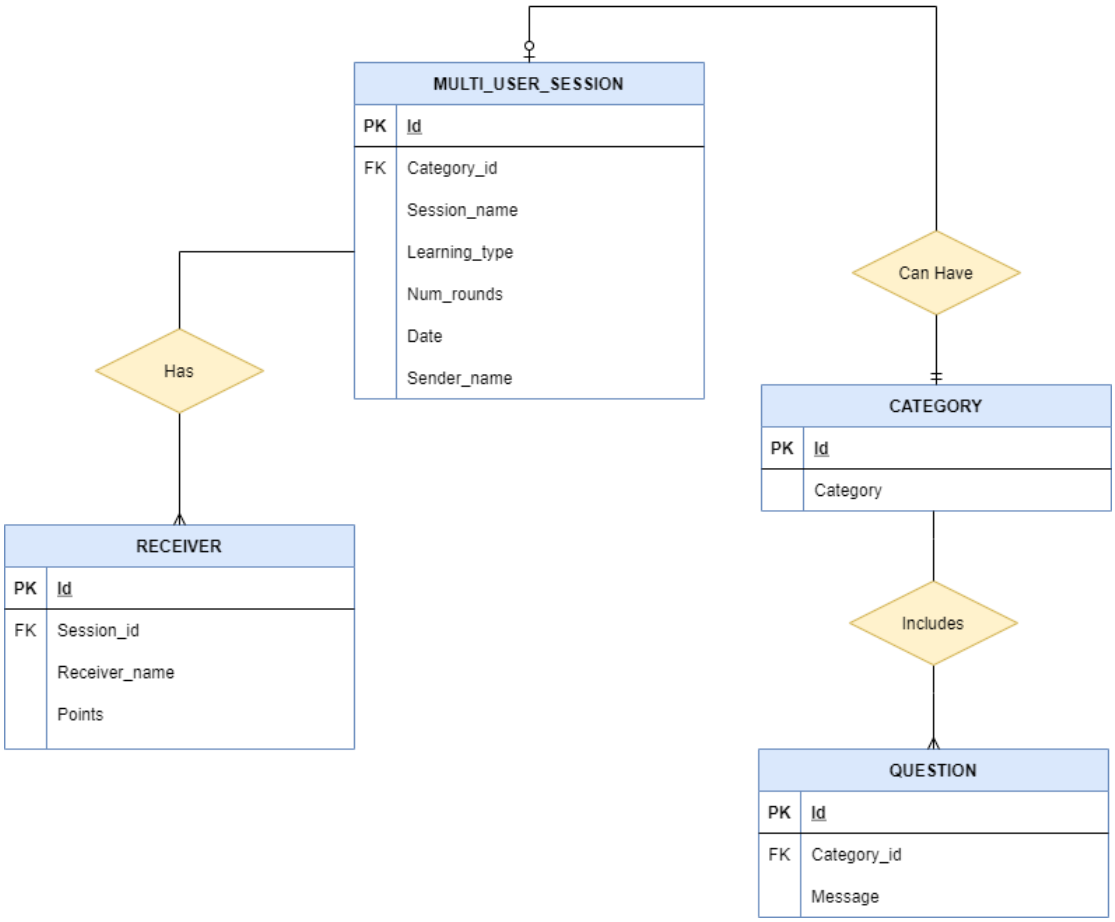


Figure 23: Morse Lighthouse's ER Diagram

The following diagram (**Figure 24**) is the architecture pattern, which is based on the MVC pattern. It shows the main components of each layer of the system.

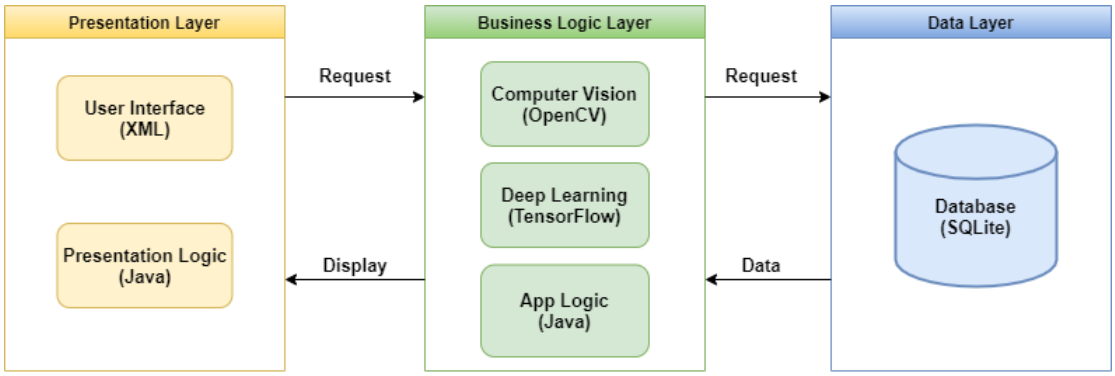


Figure 24: Morse Lighthouse's Architecture

CHAPTER 7

IMPLEMENTATION

CHAPTER 7: IMPLEMENTATION

This section is about the implementation phase, where all the user interfaces and all the requirements will be developed, how to develop these requirements, the techniques used to develop them, and the description about the development prosses. For more details about the implementation, you can check the project's JavaDocs.

7.1 User Interfaces Development

This screen is the main screen as shown in (**Figure 25**), so it will be the first screen that the user will see when he opens the application, the purple bars represent the buttons that contain the two modes that we have in our application, the user can choose between either of them.

At the bottom of the screen there are 3 buttons, the first button will open a list that contains morse code mapping to English characters and numbers, the second button will open the user's profile, and the third button will open the morse code to text translator.

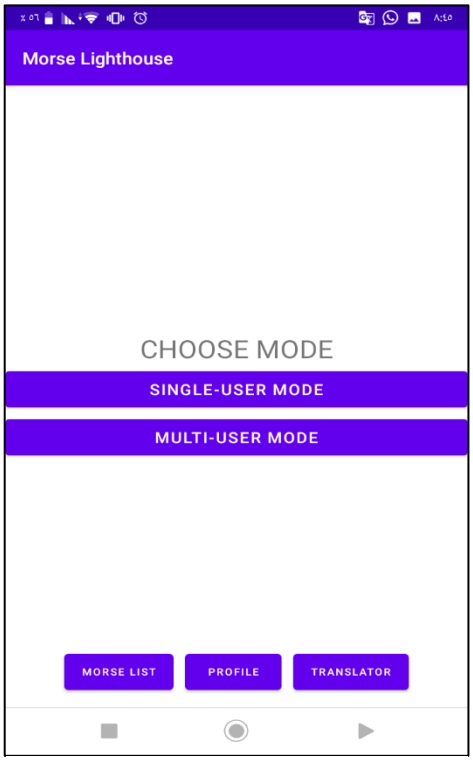


Figure 25: Main screen

This is the multi-user mode screen as shown in (**figure 26**), this screen can be accessed after the user selects the multi-user mode from the main screen, in this screen the user can choose between hosting a session or joining a session by selecting create session and join session, respectively.

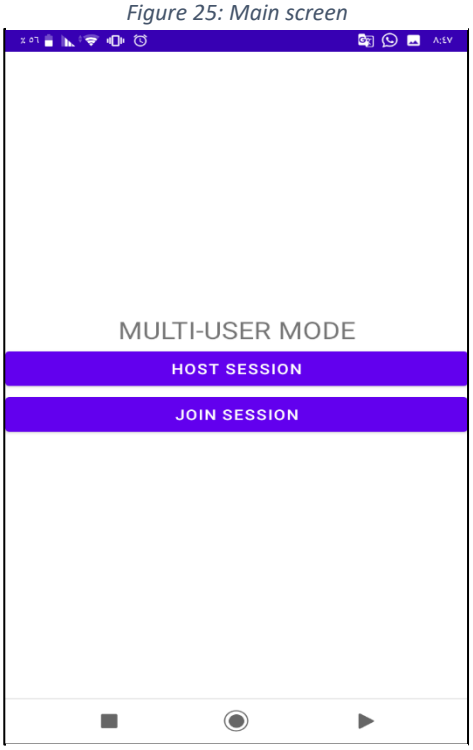


Figure 26: Multi-User mode screen

This is the session settings screen (**figure 27**), this screen can be accessed if the user selects the host session from the multi-user mode screen, here the user will be hosting a session, he can type the session name in the first text field. He can also select the number of participants and questions from the two drop boxes on the bottom left, also he can select a training session or an exam session by using the radio buttons on the bottom right, and the bottom drop box for selecting the questions' type, and the button at the bottom of the screen will create a session that other people can join in.

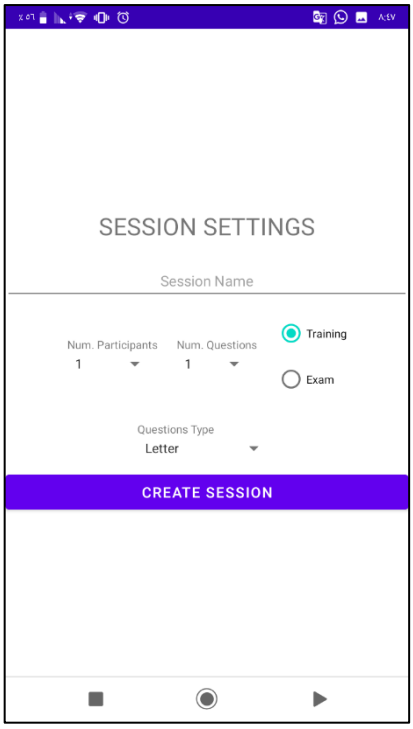


Figure 27: Session Setting Screen

This screen is the session starting screen for the host (**figure 29**), this screen can be accessed after the user clicks on the create session button from the session settings screen, at the top of the screen the name of the session will be visible, below the session name there is a waiting for participants message that indicates the IP address, and the users are not ready. In the middle of the screen, names of the participants will be visible, and after all participants join the session, the host can start the session by clicking on the start session button (**figure 30**) and the participants will wait until the host starts the session (**figure 28**).

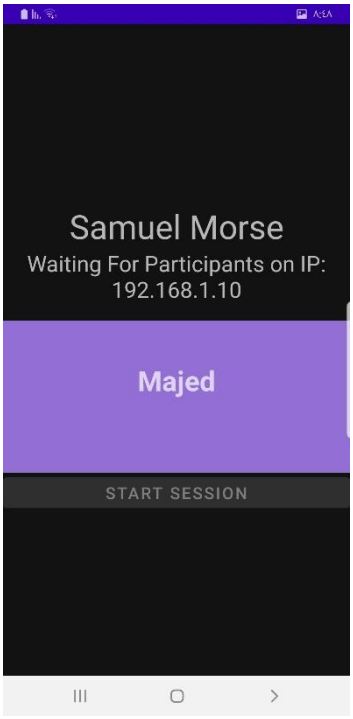


Figure 29: Host Starting Screen 1

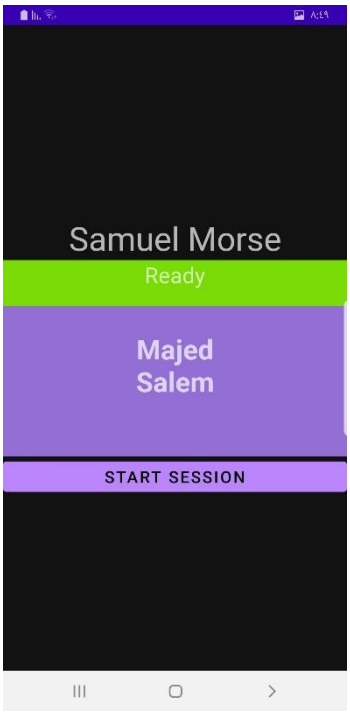


Figure 30: Host Starting Screen 2

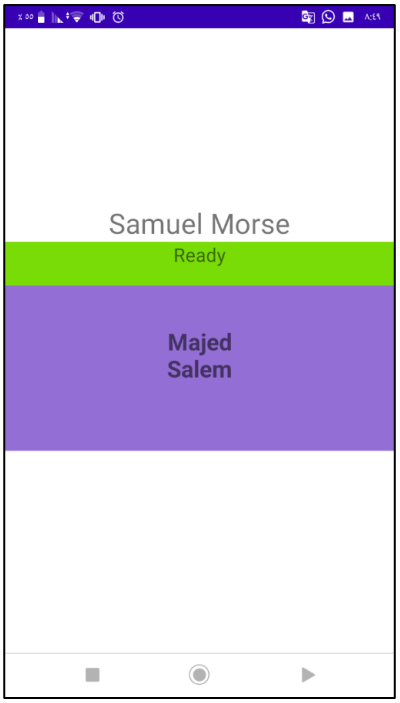


Figure 28: Receiver Starting Screen

This screen is the sender screen (**figure 31**), this screen can be accessed if the user's role was a sender in the session starting screen, at the top of the screen, the session name will be visible. The middle of the screen will be the name of the other participants, in the text field, the user can type the message he would like to send to the participants, and at the bottom of the screen he can click SEND to send the message.

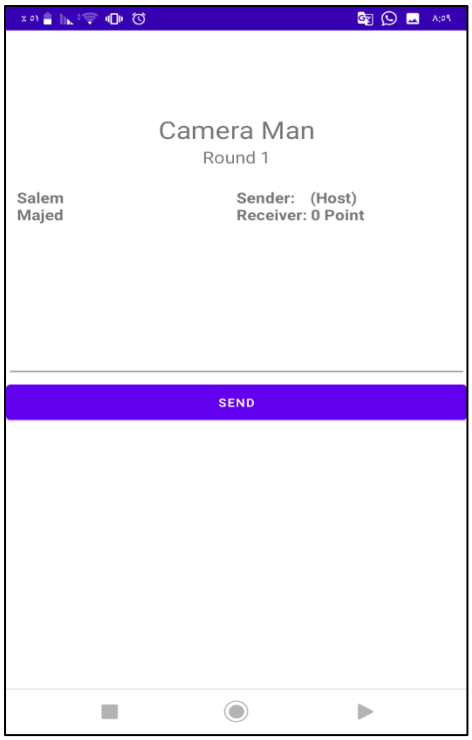


Figure 31: Sender Screen

This screen is the receiver screen (**figure 34**), this screen can be accessed if the user's role was a receiver in the session starting screen and the mode is Exam, at the top of the screen the session name will be visible, then beneath it the number of the current round of this session, in the middle of the screen the camera's feed will be visible for the user, beneath the camera the scoreboard will be available with the participants names, as well as their scored points, under the scoreboard the received message will appear, and at the bottom of the screen the user can enter his answer and if the answer is correct green message will be displayed (**figure35**) otherwise red message will be displayed (**figure32**) in the end of the session the host can end the session or continue at the same session (**figure33**).

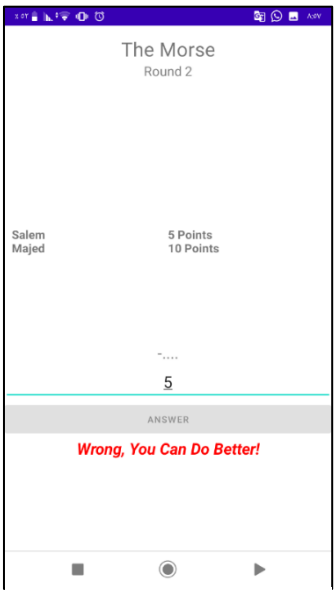


Figure 35: Receiver Screen training 1

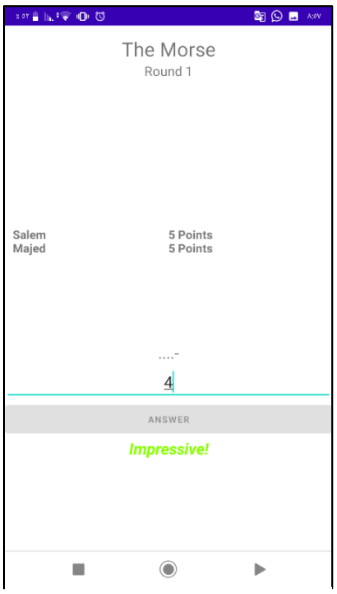


Figure 33: Receiver Screen training 2

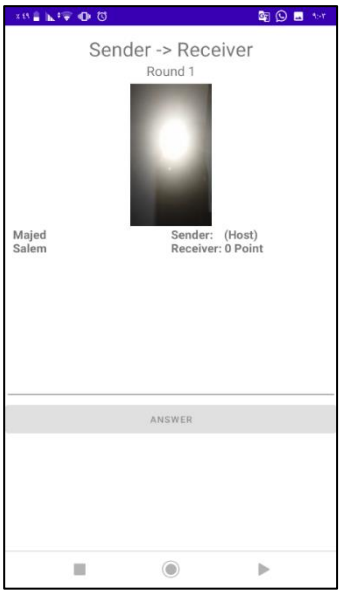


Figure 34: Receiver Screen Exam

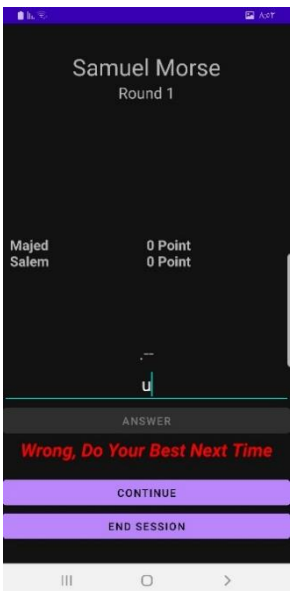


Figure 32: End or Continue.

This screen is the session joining screen (**figure 36**), this screen can be accessed if the user selects join session from the multi-user mode screen. In the middle the screen, the user can enter the IP address of the intended host to be connected, after that, he/she can click on JOIN button.

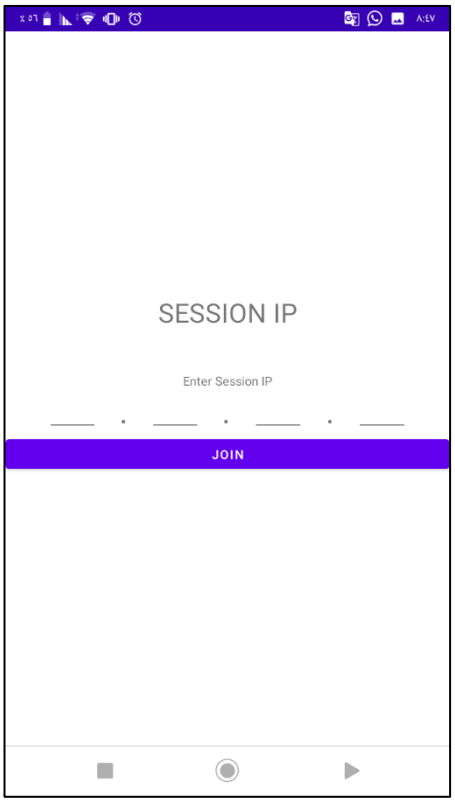


Figure 36: Session Joining Screen

This screen is the morse code list screen (**figure 37**), this screen can be accessed if the user clicks on the morse code list button from the main screen, this screen shows English letters, numbers, and punctuations, and these are their corresponding Morse code.

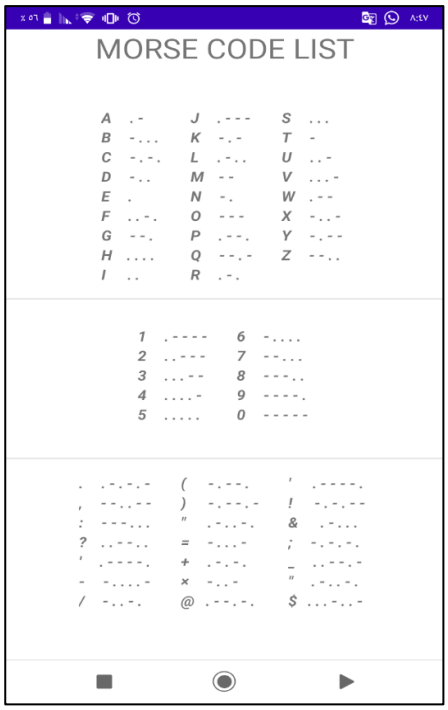


Figure 37: Morse Code List Screen

This screen is the user's profile screen (**figure 38**), this screen can be accessed if the user clicks on the dashboard button from the main screen, at the top of the screen the user name will be visible, below the name the user can view his records such as number of wins, number of losses, and rank, at the bottom there are two buttons where the user can view the history of his previous sessions in either the singe-user mode or the multi-user mode.

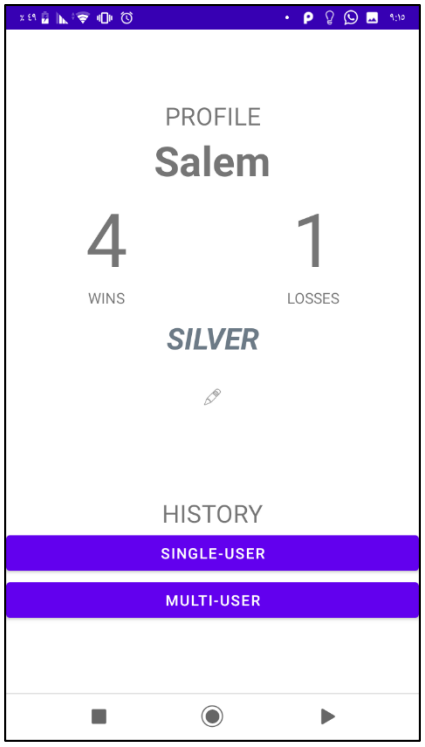


Figure 38: User Profile Screen

This screen is the session history (**figure 39**), this screen can be accessed if the user selects the multi-user button from the profile screen, this screen shows the past sessions that the user has joined in or hosted, and it views the scores in each of those sessions for example it shows the session name, the sender, participants' names with their points, the winner, and the date.

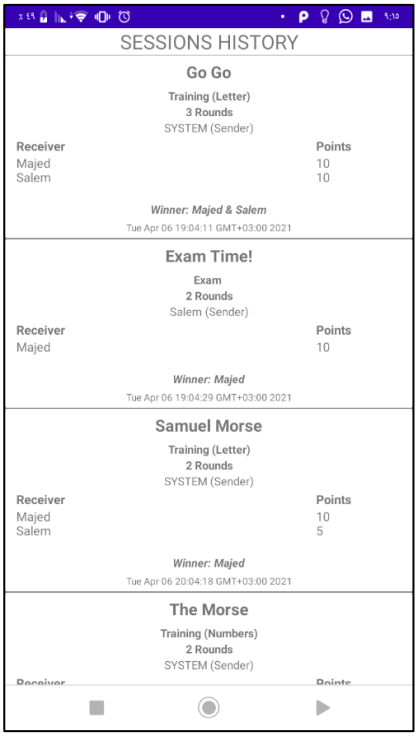


Figure 39: History Screen

This is the translator screen (**figure 40**), this screen can be accessed from the main screen by clicking on the TRANSLATOR button. In this screen, the user can type the message that he wants to be translated in the empty text field, the user can choose if he wants the message to from regular text to morse code or vice versa by clicking the button below the text field, and the bottom most button can be used to copy the translated message that will appear at the bottom of the screen.

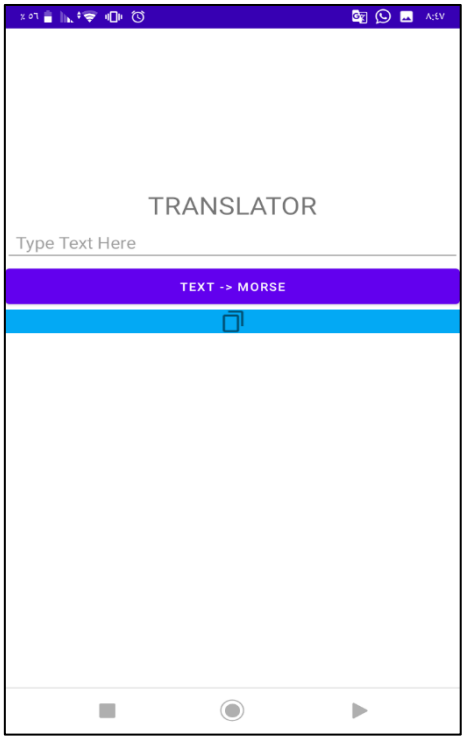


Figure 40: Translator Screen

This screen will allow the user to enter/change his/her username and as shown in **figure 41**.

The screen contains the application logo and a text field to write the new name, and there are some warning messages.

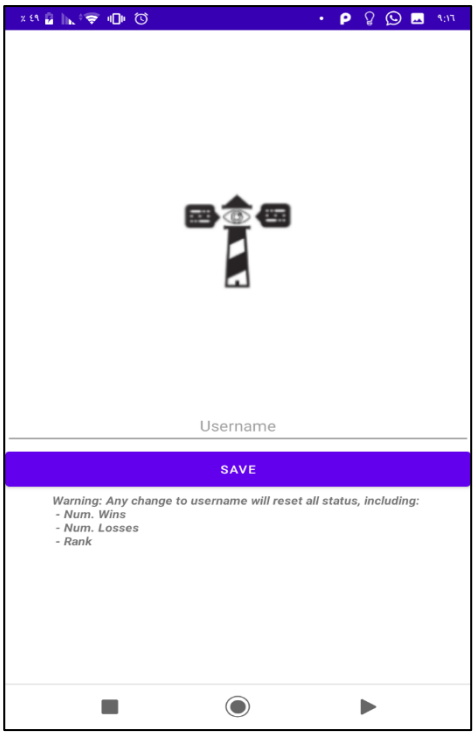


Figure 41: Enter/ Change username Screen

This screen shows the application logo and the application name (Morse Lighthouse) as shown in **figure 42**.

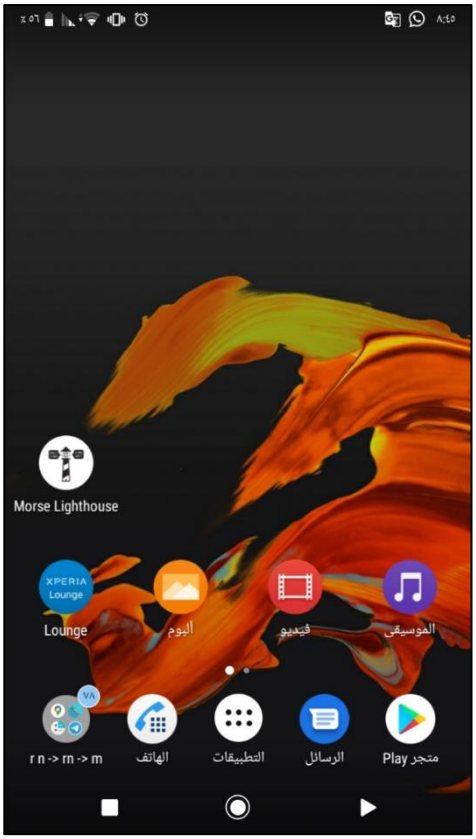


Figure 42: App logo

7.2 Database

SQLite was used to develop the database. SQLite is embedded into android also it is very lightweight, and it does not require any setup making it more suitable for the app. As shown in **figure 23** the database contains 4 tables to save the premade question and the users history. The database is created when the application starts, each table can perform CRUD operations using methods in the "DBhandler" class this class manages all the operations with the database.

7.3 Learning Model Development

At first, when the development process started, the plan was to develop the model from scratch in python, the dataset is created by recording two videos and extracting the frames as a dataset for training the model. Python is used to process the collected dataset, and then TensorFlow is used to train the model.

The trained model using python gave a good accuracy, after that the model was converted from SavedModel format to TFlite format that is used for Android mobile applications because TFlite is requires small amount of storage and has low latency with a small compromise in accuracy.

But then, many problems occurred with the model when the development reached the integration phase. So, the team decided to stop developing the model from scratch in python and to try other methods to train the AI model.

After searching for other solutions, the team decided to use Google Cloud "cloud.google.com" to train the model with the type: Object Detection. Additionally, the team collected a new dataset to fit into the model as per the requirements of the cloud.

7.3.1 Dataset

The team have collected 286 pictures that show a smartphone with a flashlight turned-on (**figure 43**). After uploading the pictures to Google Cloud, the team added bounding boxes two these pictures, here is an example of a bounding box (**figure 44**).

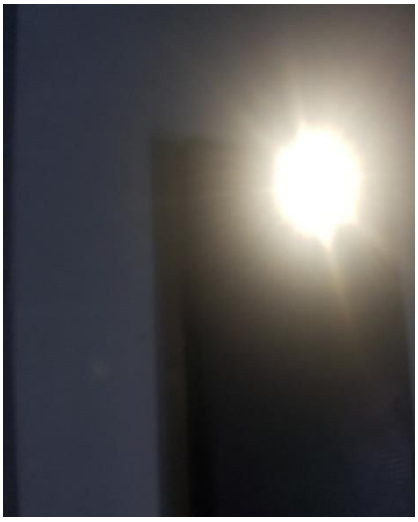


Figure 43: Without bounding box.

7.3.2 Training Phase

After adding the bounding boxes, the training of the model started. It took a couple of hours. Then the model was exported as a TFlite file format, then the team tried to integrate the model to android model.

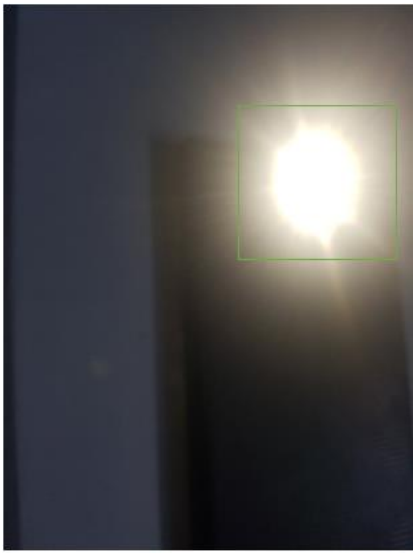


Figure 44: With bounding box

The data split of the training as shown in **figure 45**:

Data split	Randomly assigned (80/10/10)
Total items	286
Training items	229 (80.1%)
Validation items	29 (10.1%)
Test items	28 (9.8%)

Algorithm	AutoML
Objective	Image object detection

Figure 45: Data split

7.3.3 Evaluation Phase

Here is the precision, the average precision, the confidence, and the recall of the model as shown in **figure 46**.

As it is shown in the **figures 47**, the performance measures assure that the model will fit its purpose in the communication of Multi-User mode.

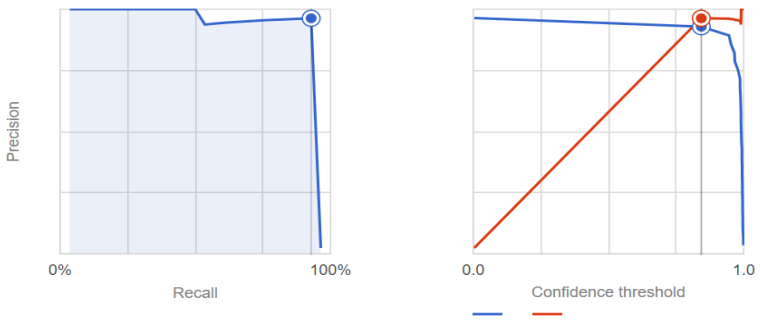


Figure 47: Performance graph

Average precision	0.751
Precision	96.3%
Recall	92.9%
Created	Apr 6, 2021, 12:00:24 PM
Total images	286
Training images	229
Validation images	29
Test images	28

Figure 46: Performance

7.4 Networking

This section discusses the networking role with the application. The networking part takes place when the users start connecting and playing with each other through the Multi-User mode.

The Two-Tier Client-Server architecture is used with the Thin-client model, where the host acts as a server that manages the connectivity, communications, and most of the application processing between the clients and itself, and the participants act as clients who connect to the host, and both (clients and host) play within a created session by the host. The **figure 48** illustrates the components and how these interact with each other:

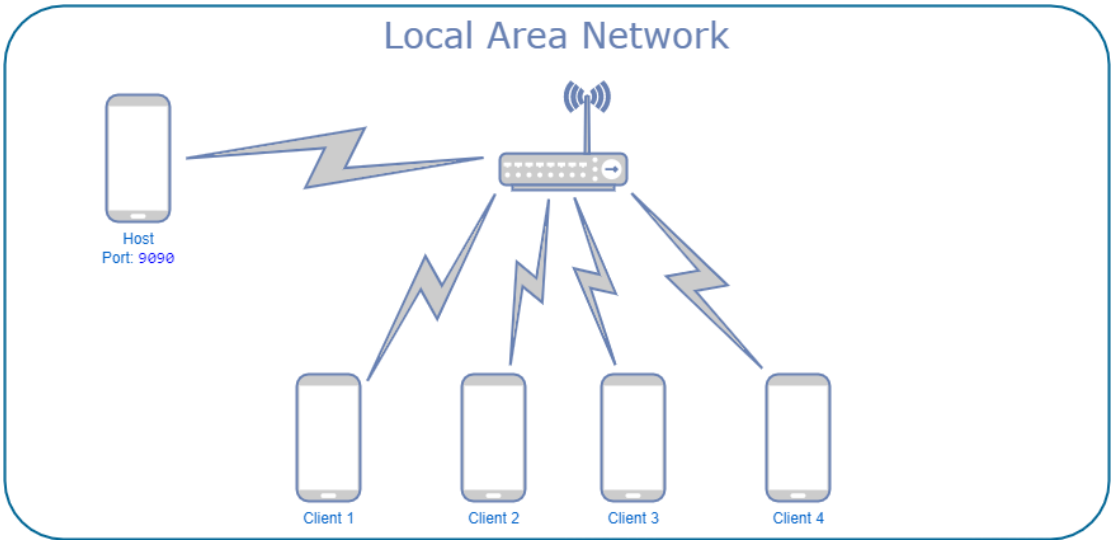


Figure 48: Networking session

As it is shown, the clients (maximum = 4) connect to the listening host by its IP Address (appears on host's phone screen) and port number: 9090, through an access point like Small Office/Home Office (SOHO) device.

7.5 Flashlight

To access the smartphone's flashlight, camera permissions are required. These permissions are required to access the camera services, and the flashlight is one of them, then the camera manger is used to turn on and off the flashlight. The flashlight then is used to send morse signals by turning the flashlight on for half a second if the message contains a dot, or by a second and a half if the message contains a dash.

7.6 User Profile

Each user in the application has his/her profile. The profile saves the user's information:

- ID
- Username
- Number of Wins
- Number of Losses
- Points

The application stores and retrieves the user's profile through Serialization and Deserialization, respectively. When the user starts the application, the application will first check if the file exists within the phone, and there are two cases:

- The file does not exist: In that case, the application will prompt the user to enter his/her new username. After that, the system creates a new profile and stores it within the smartphone's internal storage.
- The file exists: In that case, the system loads the user's profile.

There are two cases where the system updates the user's profile:

- When the host in Multi-User mode ends a session, the host and connected clients update their profile information (Wins and Losses).
- When the user wishes to change his/her name, the system changes it and resets the profile's information.

Figure 49 demonstrates the serialization and deserialization process:

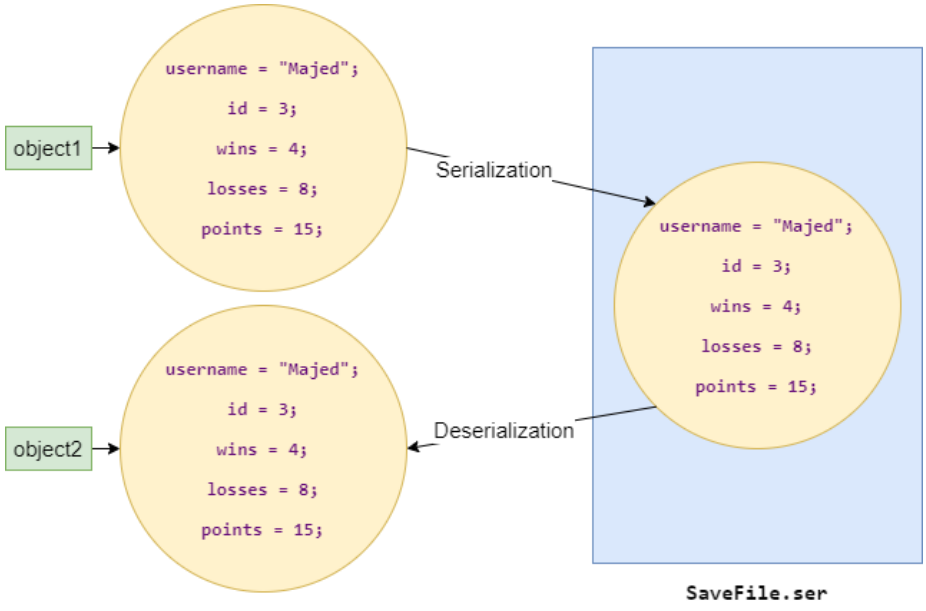


Figure 49: Serialization

CHAPTER 8

TESTING

CHAPTER 8: TESTING

In this section testing will be applied for these requirements:

- FRU1. Enter natural language text or Morse code (dot/s and/or dash/s).
- FRU2. View his/her statistics.
- FRU3. View his/her session history.
- FRU4. View list of natural language letters to Morse code.
- FRU5. Change his/her username.
- FRM1. Translate between Morse code text and natural language text using a translator.
- FRM2. Use a flashlight for Morse code generation.
- FRM3. Use the Gamification Learning technique.
- FRM5. Have a training and exam process for Multi-User mode.
- FRM8. Have Multi-User mode.
    - FRM8.1. Only one user can be a host.
    - FRM8.2. Create a session.
        - FRM8.2.1. The session has 2 to 5 users.
        - FRM8.2.2. The session has 2 to 10 questions.
    - FRM8.3. Start a session.
    - FRM8.5. Able to repeat the session.
    - FRM8.6. Able to end the session.
    - FRM8.7. Join the session through the IP Address.
    - FRM8.8. View the scoreboard at any time throughout the session.
    - FRM8.9. Enter the message content for the question that will be sent.
        - FRM8.9.1. The number of characters must not exceed 20 characters.
    - FRM8.10 Send the question.
    - FRM8.12. Enter the answer corresponding to the Morse signals.
- FRM9. Verify the correctness of the user's answer.
- FRM10. Record the user's performance.

These requirements will not be tested:

These are **extended requirement** and are non-essential for the project.

- FRM4. Have a predefined question set. *
- FRM6. Have a training and exam process for Single-User. *
- FRM7. Have Single-User mode. *
    - FRM7.1. Use vibration for Morse code generation. *
    - FRM7.2. Use the Haptic Learning technique. *
    - FRM7.3. The trainee shall practice through the workshop. *
        - FRM8.3. Determine the difficulty level of questions e.g., Letters or Words. *
- FRM8.11. Use the smartphone's camera (Computer Vision) to receive Morse code light signals.

8.1 Testing Criteria:

FRU4:

- Pass:
    - if the username is between 1 and 15 characters.
- Fail:
    - if the username is less than 1 or over 15 characters.

FRM1:

- Pass:
    - If the message is between 1 and 200 characters.
    - If the entered message is in English or Morse code.
- Fail:
    - If the message over 200 characters.
    - If the entered message is not in English or Morse code.

FRM8.6:

- Pass:
    - Every field in the IP must be a number between 0 and 255.
- Fail:
    - If the user entered a number more than 255 or less than 0.
    - Any character other than digits.

FRM8.10:

- Pass:
    - If the message is between 1 and 20 characters.
    - If the message is in English.
- Fail:
    - if the message is less than 1 and more than 20 characters.
    - If the message is not in English.

FRM8.12:

- Pass:
    - If the message is between 1 and 20 characters.
    - If the message is in English.
- Fail:
    - if the message is less than 1 and more than 20 characters.
    - If the message is not in English.

FRM8.2:

- Pass:
    - The session title must be between 1 and 20 characters.
    - If dropdown boxes choices are selected.
    - One of the radio buttons must be selected.
- Fail:
    - The session title is less than 1 or more than 20 characters.
    - If one of the dropdown boxes choice not selected.
    - If none of the radio buttons are selected.

8.2 Unit Testing Scenario:

This section provides examples and techniques that was used for **testing the most important functionalities in the project**. Unit testing in this section will be **black box testing techniques.**

- FRU4.

  For this feature, the used technique is **Boundary Value Analysis technique** to impose the constraints on the character length for changing a user's username. The following scenarios are for changing username:

  1. "Mohammed" Is a valid username.
  2. "HelloMyNameIsMohammedSalehAlharbi" Is not a valid username.

Table 16: Test FRU4

Invalid partition – valid partition Lower boundary.		Invalid partition – valid partition upper boundary.	
Boundary value just below the boundary	Boundary value just above the boundary	Boundary value just below the boundary	Boundary value just above the boundary
0	1	15	16

- FRM1.

  For this feature, the used technique is **Decision Table technique** to impose the constraints on the translator. The following scenarios are for inserting a message in the translator:

  1. "Hello World" Is a valid message.
  2. "يا هلا ويا مرحبا بالعالم" Is not a valid message.

Table 17: Test FRM1

Conditions	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Characters' length < 200	Y	N	Y	Y	Y	Y	Y
Characters' length > 0	Y	Y	N	Y	Y	N	Y
English characters	Y	Y	_	N	Y	N	N
Morse code	Y	Y	_	Y	N	Y	N
Outcome							
Translate	N	N	N	Y	Y	N	
Message	"Not valid input"	"Message is too long"	"Please enter a message"	–	–	"Message is too long"	"Not valid input"

- FRM8.6.
  For this feature, the used technique is **Boundary Value Analysis technique** to impose the constraints on this feature. The IP address is composed of four fields, each field should take a number between 0 and 255. The following scenarios are for inserting an IP address:
  1. "192.169.1.1" Is a valid IP address.
  2. "99999.335jfs34.31sd3.2rf3" Is not a valid IP address.

Table 18: FRM8.6

Invalid partition – valid partition Lower boundary.		Invalid partition – valid partition upper boundary.	
Boundary value just below the boundary	Boundary value just above the boundary	Boundary value just below the boundary	Boundary value just above the boundary
-1	0	255	256

- FRM8.10.
  For this feature, the used technique is **Decision Table technique** to impose the constraints on sending a question in an ongoing session. The following scenarios are for sending a question:
  1. "SOS" Is a valid question.
  2. "سؤال" Is not a valid question.

Table 19: Test FRM8.10

Conditions	Test 1	Test 2	Test 3	Test 4
Characters' length < 20	Y	N	Y	Y
Characters' length > 0	Y	Y	N	Y
English characters	Y	Y	_	N
Outcome				
Send question	Y	N	N	N
Message	_	"Message is too long"	"Please enter a message"	"Not valid input"

- FRM8.12.
  For this feature, the used technique is **Decision Table technique** to impose the constraints on answering a question in an ongoing session. The following scenarios are for answering a question:
  1. "NON" Is a valid answer.
  2. "初めまして" Is not a valid question.

Table 20: Test FRM8.12

Conditions	Test 1	Test 2	Test 3	Test 4
Characters' length < 20	Y	N	Y	Y
Characters' length > 0	Y	Y	N	Y
English characters	Y	Y	_	N
Outcome				
Send question	Y	N	N	N
Message	_	"Message is too long"	"Please enter a message"	"Not valid input"

- FRM8.2.

For this feature, the used technique is **Decision Table technique** to impose the constraints on creating a session. The following scenarios are for creating a session:

Table 21: Test FRM8.2s

Conditions	Test 1	Test 2	Test 3	Test 4	Test 5	Test6	Test 7	Test 8	Test 9	Test 10	Test 11	Test 12
Session name length < 20	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Session name length > 0	Y	Y	N	Y	Y	Y	N	Y	Y	N	Y	N
Dropdown boxes selected	Y	Y	Y	N	Y	Y	N	N	Y	N	N	N
Exam Radio button	Y	Y	Y	Y	N	Y	Y	N	N	N	N	N
Training Radio Button	Y	Y	Y	Y	Y	N	Y	Y	N	Y	N	N
Outcome												
Create Session	N	N	N	N	Y	Y	N	N	N	N	N	N
Message	"Error, both radio buttons are selected"	"Error, both radio buttons are selected"	"Please enter session name"	"Please select the dropdown boxes"	–	–	"Please enter session name"	"Please select the dropdown boxes"	"Please select one of the radio buttons"	"Please enter session name"	"Please select the dropdown boxes"	"Please enter session name"

8.3 Integration Testing:

For integration testing the incremental approach was used in a bottom-up fashion, and these are the integrated and tested components so far:

- Database
- Translator
- LAN Component
- Training mode in Multi-User Mode
- Exam mode in Multi-User Mode
- Camera

The component that is yet to be integrated is:

- AI Model

8.4 System Testing:

Not all the functionalities are completed, and therefore the app is still not integrated fully so system testing is yet to be started.

CHAPTER 9

CONCLUSION & RECOMMENDATION

CHAPTER 9: CONCLUSION & RECOMMENDATION

This document suggests a smartphone application that teaches Morse code by using gamification, haptic learning, and collaborative learning techniques. This document offers two learning modes, one for a Single-User mode, and the other is for a Multi-User mode. These modes and techniques make use of deep learning, computer vision for decoding Morse code.

These modes and techniques allow the application users to understand and learn Morse code in a simple and efficient way.

This document has discussed the software phases of both CPCS-498 and CPCS-499, these phases include the planning phase, analysis phase, design phase, the implementation phase, testing phase, and developing a functioning prototype for the proposed system.

In conclusion, it can be established that the developed application can enhance the learning of Morse code in an engaging and enjoyable way, which allows the users to practice Morse code in an efficient manner.

REFERENCES

[1] T. E. o. E. Britannica, "Morse Code," September 18, 2020. [Online]. Available: https://www.britannica.com/topic/Morse-Code. [Accessed October 20, 2020].

[2] " ،Morse Code Agent،" Tsai Chieh-Ping .[متصل] ،Available: https://play.google.com/store/apps/details?id=com.erdatsai.morsecodeagent [تاريخ الوصولOctober 20, 2020.[.

[3] "Morse code - learn and play "،LimesDevelopment .[متصل] ،Available: https://play.google.com/store/apps/details?id=com.limesdevelopment.morsecode . [تاريخ الوصولOctober 20, 2020.[

[4] J. Ritter و R. Brownstein ،"Teaching Morse Code Using a Virtual Classroom Environment،" The CW Operators' Club.2012 ،

[5] N. YANES and I. BOUOUD, "Using Gamification and Serious Games for English Language Learning", 2019 International Conference on Computer and Information Sciences (ICCIS), 2019. Available: 10.1109/iccisci.2019.8716451 [Accessed 21 October 2020].

[6] M. Tarkowski, P. Woznica and L. Kulas, "Efficient algorithm for blinking LED detection dedicated to embedded systems equipped with high performance cameras", IEEE EUROCON 2015 - International Conference on Computer as a Tool (EUROCON), 2015. Available: 10.1109/eurocon.2015.7313723 [Accessed 21 October 2020].

[7] C. Seim et al., "Towards haptic learning on a smartwatch", Proceedings of the 2018 ACM International Symposium on Wearable Computers - ISWC '18, 2018. Available: 10.1145/3267242.3267269 [Accessed 21 October 2020].

[8] W. Li and K. Wang, "Research on Automatic Decoding of Morse Code Based on Deep Learning", 2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS), 2019. Available: 10.1109/icicas48597.2019.00107 [Accessed 21 October 2020].

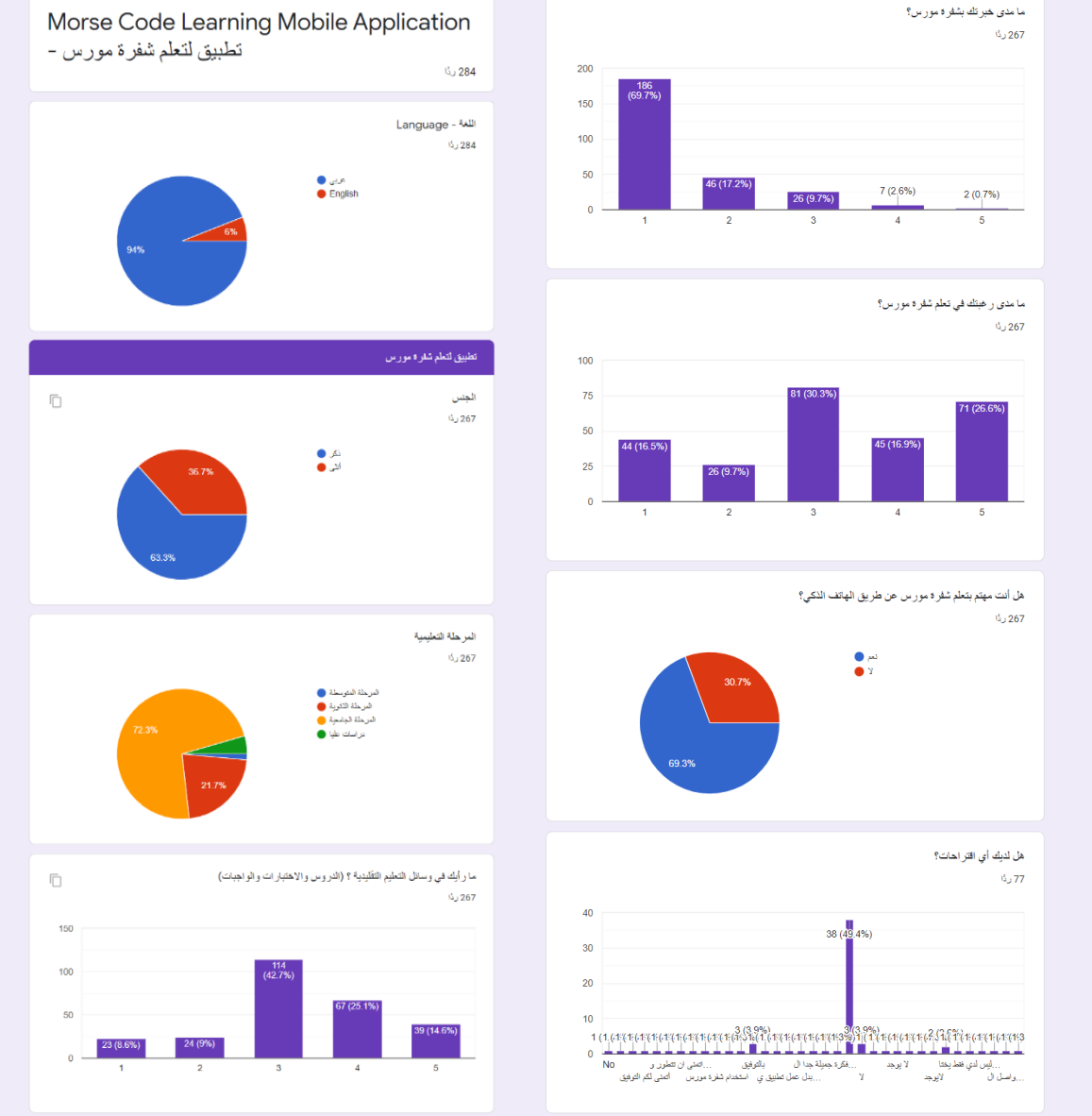
[9] J. Pan, J. Li, P. Tsai and L. Jain, Advances in Intelligent Information Hiding and Multimedia Signal Processing. Singapore: Springer Singapore, 2020, pp. 235-242.

APPENDIX A

QUESTIONNAIRE

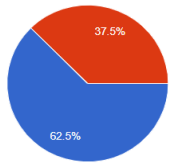
APPENDIX A: QUESTIONNAIRE

This is the questionnaire that has been made and published to the public through "Google Forms". This will be helpful to understand users' data for the project.



هل استخدمت تطبيقات تعليمية على الهواتف الذكية؟

267 ردًا



إذا كان جوابك في السؤال السابق نعم, فما اسم هذا التطبيق؟

146 ردًا

بلاك بورد

بلاك بورد

تيمز

Duolingo

حلول

Blackboard

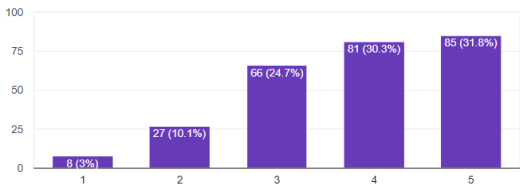
دوليتقو

مدرستي

التيمز

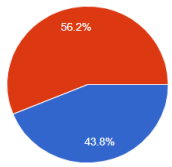
ما رأيك في التطبيقات التعليمية على الهواتف الذكية؟

267 ردًا



هل سمعت بشفرة مورس؟

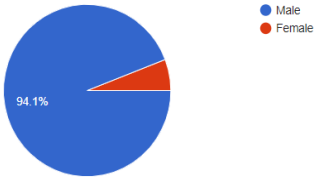
267 ردًا



Morse Code Learning Mobile Application

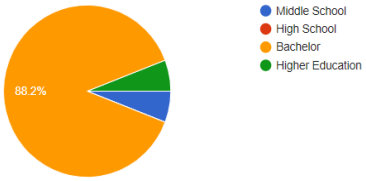
Gender

17 ردًا



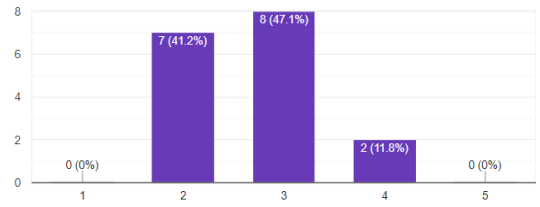
Education Level

17 ردًا



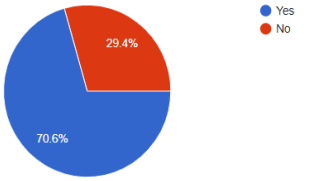
?What is your opinion on traditional education

17 ردًا



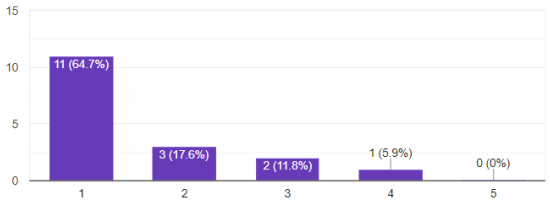
?Did you use any educational application on smart phones

17 ردًا



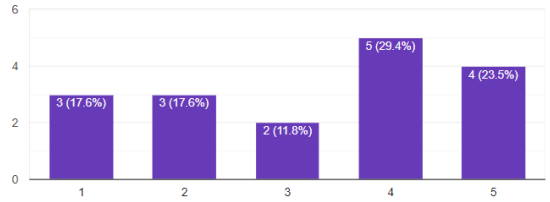
?How proficient are you in Morse code

17 ردًا



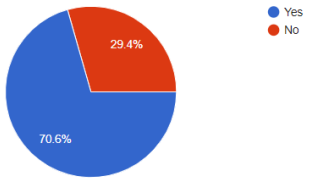
?How interested are you in learning Morse code

17 ردًا



?Would you be interested in learning Morse code using a smartphone

17 ردًا



?Do you have any suggestions

5 ردود

.Best of luck

Interact with other people by sending morse code online

Fall project

.Make this app user friendly and easy to use, all the best

Make it learning py doing not just read and read and stupid quiestion answer make it as projects you code use some IDE or sandbox to code using the mobile phone

If you answered yes in the previous question, what is the name of that ?application

11 ردًا

Blackboard & Cambly

Dulingo

Notion, ManhattanGRE, Blackboard

Mathlab, Kahoot and some other mathematic apps

Indian guy on YouTube

Canvas Student

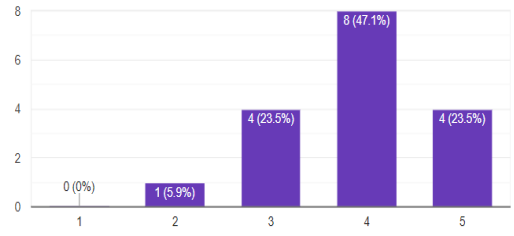
Duolingo

Blackboard

Black board

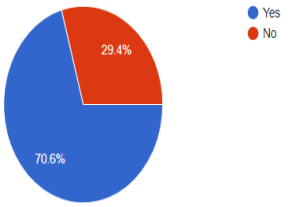
?What is you opinion about educational applications on smart phones

17 ردًا



?Did know about Morse code

17 ردًا



APPENDIX B

USE-CASE TABLES

APPENDIX B: USE-CASE TABLES

The following use-case tables describe the use-case diagram figures in this document. (see **Figure 13, 14, 15, 16, 17, 18**)

Table 22: UC1

Use Case Documentation	
Use Case ID:	UC1
Use Case Name:	List the Morse Code's Letters
Actors:	
Spectator	
Flow of Events of the Primary Scenario:	
Select " Morse Code List" from the main menu.	
Postconditions:	
The app shows the Morse code list.	

Table 23: UC2

Use Case Documentation	
Use Case ID:	UC2
Use Case Name:	Show User's Status
Actors:	
Spectator	
Flow of Events of the Primary Scenario:	
Select "User's Status" from the main menu.	
Postconditions:	
The app shows the user's status.	

Table 24: UC3

Use Case Documentation	
Use Case ID:	UC3
Use Case Name:	Go to Translator
Actors:	
User	
Flow of Events of the Primary Scenario:	
Select "Translator" from the main menu.	
Postconditions:	
The app shows the translator.	

Table 25: UC4

Use Case Documentation	
Use Case ID:	UC4
Use Case Name:	Go to Settings
Actors:	
User	
Flow of Events of the Primary Scenario:	
Select "Settings" from the main menu.	
Postconditions:	
The app shows the settings, where the user can change theme, language, and other things…	

Table 26: UC5

Use Case Documentation	
Use Case ID:	UC5
Use Case Name:	Choose Mode
Actors:	
Spectator	
Flow of Events of the Primary Scenario:	
Select "Single Mode" from the menu.	
Flow of Events of the Alternative Scenarios:	
Select "Group Mode" from the menu.	
Postconditions:	
The app shows "Single Mode" or "Group Mode" menu.	

Table 27: UC6

Use Case Documentation	
Use Case ID:	UC6
Use Case Name:	Choose Process Type
Actors:	
Participant, Trainee	
Flow of Events of the Primary Scenario:	
Select "Learn" from the menu.	
Flow of Events of the Alternative Scenarios:	
Select "Test" from the menu.	
“Used” Use Cases:	
UC5: Choose mode (Single Mode or Group Mode)	
Postconditions:	
The app shows "Learning" or "Testing" menu.	

Table 28: UC7

Use Case Documentation	
Use Case ID:	UC7
Use Case Name:	Join a Session
Actors:	
Participant	
Preconditions:	
Under "Group Mode"	
Flow of Events of the Primary Scenario:	
1. Select "Join Session" from the menu. 2. Choose a session with its name. 3. Enter the session's password	
Flow of Events of the Exception Scenarios:	
If the entered password is wrong, then the system will ask you to enter the right one.	
“Used” Use Cases:	
UC6: Choose Process Type (Learning Process or Testing Process)	
Postconditions:	
The app connects to session that a host conducted.	

Table 29: UC8

Use Case Documentation	
Use Case ID:	UC8
Use Case Name:	Choose Host Mode
Actors:	
Participant	
Preconditions:	
Under "Group Mode"	
Flow of Events of the Primary Scenario:	
Select "Host Mode" from the menu.	
“Used” Use Cases:	
UC6: Choose Process Type (Learning Process or Testing Process)	
Postconditions:	
The app shows "Host" menu	

Table 30: UC9

Use Case Documentation	
Use Case ID:	UC9
Use Case Name:	Practice
Actors:	
Trainee	
Preconditions:	
Under "Single Mode"	
Flow of Events of the Primary Scenario:	
Select "Practice on Letters" from the menu.	
Flow of Events of the Alternative Scenarios:	
Select "Practice on Words" from the menu.	
“Used” Use Cases:	
• UC10: Use Predefined Question Sets • UC11: Save Progress	
Postconditions:	
The app starts "Practice on Letters" or "Practice on Words" process	

Table 31: UC10

Use Case Documentation	
Use Case ID:	UC10
Use Case Name:	Use Predefined Question Sets
Actors:	
Morse Lighthouse (in Single Mode) or Host (in Group Mode)	
Preconditions:	
Under either Single Mode or Group Mode	
Flow of Events of the Primary Scenario:	
For Morse Lighthouse actor (Single Mode): 1. Select a Question set. 2. Start showing question i (i = 1 to num. Questions) 3. Show the next question (i + 1) after user's answer 4. Repeat step 2 until finishing all questions from the given test bank	
Flow of Events of the Alternative Scenarios:	
For Host actor (Group Mode): 1. Select "Question Sets" from the menu. 2. Select a Question set with the desired title.	
Postconditions:	
In Single Mode: the user finished training or testing process. In Group Mode: the chosen question set will be available for use in the current session.	

Table 32: UC11

Use Case Documentation	
Use Case ID:	UC11
Use Case Name:	Save Progress
Actors:	
Morse Lighthouse	
Preconditions:	
Under either Single Mode or Group Mode	
Flow of Events of the Primary Scenario:	
For Single/Group Mode: Save user's performance after learning or testing process.	
“Used” Use Cases:	
UC9: Practice (Only in Single Mode)	
Postconditions:	
In Single/Group Mode: The user's performance is saved in the system	

Table 33: UC12

Use Case Documentation	
Use Case ID:	UC12
Use Case Name:	Check Num. Characters
Actors:	
Morse Lighthouse	
Preconditions:	
Under Group Mode	
Flow of Events of the Primary Scenario:	
Check if Num. characters of user's input did not exceed 20	
Flow of Events of the Exception Scenarios:	
If Num. characters of user's input exceeded 20, then the system asks you to write a message which does not exceed 20 characters to be accepted	
Postconditions:	
The system accepts the entered message.	

Table 34: UC13

Use Case Documentation	
Use Case ID:	UC13
Use Case Name:	Verify the Answer
Actors:	
Morse Lighthouse	
Preconditions:	
Under either Single Mode or Group Mode	
Flow of Events of the Primary Scenario:	
Check if the user's input is corresponding to the received Morse code	
Postconditions:	
If the user's input is corresponding to the received Morse code, then print "Correct!!", otherwise, print "Wrong Answer"	

Table 35: UC14

Use Case Documentation	
Use Case ID:	UC14
Use Case Name:	Start Session
Actors:	
Host	
Preconditions:	
Under "Group Mode"	
Flow of Events of the Primary Scenario:	
Tap on "Start Session" button	
Postconditions:	
The system will start the learning or testing process	

Table 36: UC15

Use Case Documentation	
Use Case ID:	UC15
Use Case Name:	Set Difficulty Level
Actors:	
Host	
Preconditions:	
Under "Group Mode"	
Flow of Events of the Primary Scenario:	
1. Tap on "Difficulty Level" button 2. Choose either "Letters" or "Words".	
Postconditions:	
The difficulty level is set based on the host choice.	

Table 37: UC16

Use Case Documentation	
Use Case ID:	UC16
Use Case Name:	Set Num. Questions
Actors:	
Host	
Preconditions:	
Under "Group Mode"	
Flow of Events of the Primary Scenario:	
1. Tap on " Num. Questions" button 2. Enter a number: 2 to 10 questions	
Extension Points:	
2a. You can use "Predefined Question Sets" option	
Postconditions:	
The system is set with the entered Num. questions with (optionally) predefined question set	

Table 38: UC17

Use Case Documentation	
Use Case ID:	UC17
Use Case Name:	Repeat the Session
Actors:	
Host	
Preconditions:	
Under "Group Mode"	
Flow of Events of the Primary Scenario:	
Tap on " Repeat the Session" button	
Postconditions:	
The system is repeat the session.	

Table 39: UC18

Use Case Documentation	
Use Case ID:	UC18
Use Case Name:	End the Session
Actors:	
Host	
Preconditions:	
Under "Group Mode"	
Flow of Events of the Primary Scenario:	
Tap on " End the Session" button	
“Used” Use Cases:	
UC11: "Save Progress"	
Postconditions:	
The system will end the session and take you to the Group mode menu.	

Table 40: UC19

Use Case Documentation	
Use Case ID:	UC19
Use Case Name:	Choose the Role
Actors:	
Participant	
Preconditions:	
Under "Group Mode"	
Flow of Events of the Primary Scenario:	
1. Tap on "Choose the Role" button 2. Select "Sender" or "Receiver"	
Postconditions:	
The system will assign you as "Sender" or "Receiver"	

Table 41: UC20

Use Case Documentation	
Use Case ID:	UC20
Use Case Name:	View Scoreboard
Actors:	
Participant, Trainee	
Preconditions:	
Under either Single or Group Mode	
Flow of Events of the Primary Scenario:	
Tap on "View Scoreboard" button	
Postconditions:	
The system shows the scoreboard.	

Table 42: UC21

Use Case Documentation	
Use Case ID:	UC21
Use Case Name:	Create Message
Actors:	
Sender	
Preconditions:	
It must be under "Sender" menu.	
Flow of Events of the Primary Scenario:	
1. Tap on " Create Message" button 2. Write the Message	
Flow of Events of the Alternative Scenarios:	
1. Tap on " Create Message" button 2. Select the predefined message	
“Used” Use Cases:	
UC12: "Check Num. Characters" after the 2nd step of primary scenario.	
Postconditions:	
The message will be ready to be sent	

Table 43: UC22

Use Case Documentation	
Use Case ID:	UC22
Use Case Name:	Send Message
Actors:	
Sender	
Preconditions:	
It must be under "Sender" menu.	
Flow of Events of the Primary Scenario:	
1. Tap on "Send Message" button 2. Choose: "Send Message Through Flashlight"	
Flow of Events of the Alternative Scenarios:	
1. Tap on "Send Message" button 2. Choose: "Send Message Through Speaker"	
“Used” Use Cases:	
UC21: Create Message	
Postconditions:	
The system will send the message through the flashlight or speaker.	

Table 44: UC23

Use Case Documentation	
Use Case ID:	UC23
Use Case Name:	Start Receive Message
Actors:	
Receiver	
Preconditions:	
It must be under "Receiver" menu.	
Flow of Events of the Primary Scenario:	
1. Tap on "Receive Message" button 2. Choose: " Receive Message Through Camera"	
Flow of Events of the Alternative Scenarios:	
1. Tap on "Receive Message" button 2. Choose: " Receive Message Through Mic"	
Postconditions:	
The system will start receives the message that comes through light or sound signals.	

Table 45: UC24

Use Case Documentation	
Use Case ID:	UC24
Use Case Name:	Enter the Answer
Actors:	
Receiver	
Preconditions:	
The receiver must have the received message	
Flow of Events of the Primary Scenario:	
1. Enter the answer that is correspond to the given Morse code message 2. Tap on "Enter"	
“Used” Use Cases:	
UC13: "Verify the Answer" after the 2nd step.	
Postconditions:	
The system will print the result based on the following Use-Case, UC13: "Verify the Answer.	

APPENDIX C

MEETINGS SCHEDULE

APPENDIX C: MEETINGS SCHEDULE

The following table (**Table 40**) shows the meetings schedule through this semester in CPCS-498.

Table 46: Meetings Schedule

CPCS-498 Meetings schedule				
DATE	What has been done	Start	Duration	Attendance
13 September	- Select a new title (Morse Code Communication App (MCCA)) - Planning how to collect the requirements - Gathering some useful resources - Planning for the next meeting	5:00 PM	2H	All
15 September	- Strat the Domain analyses - Review the requirements - Planning for the next meeting	9:00 PM	2H	All
20 September	- Start Prepare the first presentation - Review the Domain analyses - Finalize the project proposal - Planning for the next meeting	9:00 PM	2H	All
22 September	- Fill the weekly progress form - Divide the new tasks among the team members - Planning for the next meeting	9:30 PM	2.5H	All
27 September	- Select a new title (Morse KAU: An AI based Morse Code Communication App) - Reevaluating problem definition and the project scope - Start the feasibility study - Planning for the next meeting	9:00 PM	2H	All
29 September	- Finalize Project Scope - Determine target users - Planning for the next meeting	9:30 PM	2.5H	All
30 September	- Fill the weekly progress form - Start the Work breakdown Structure - Planning for the next meeting	9:00 PM	2H	All
4 October	- Update Work breakdown Structure - Update feasibility study - Planning for the next meeting	9:00 PM	2H	All
6 October	- Start the building the schedule in MS Project - Planning for the next meeting	9:00 PM	2H	All
7 October	- Search for paper - Start the literature review - Planning for the next meeting	6:00 PM	4H	All
11 October	- Defined the research questions - selected 2 research papers to be reviewed - updated the schedule - Collected more research papers - Planning for the next meeting	9:30 PM	2H	All
13 October	- Found more research papers - Selected 2 more papers - Planning for the next meeting	9:00 PM	2H	All

Continue…

…Continuation

14 October	- Summarize 2 research paper in the literature review - Work on the report 1 format and introduction - Found more research papers - Planning for the next meeting	8:00 PM	4H	All
16 October	- write the report introduction - write suggested solution - write what is the equipment and software used in the project - Planning for the next meeting	9:30 PM	2H	All
17 October	- design the flowchart - write future work - Edit the report structure - Planning for the next meeting	6:00 PM	3H	All
18 October	- design the implementation flowchart - write requirement and scope in the report - Planning for the next meeting	9:00 PM	3H	All
19 October	- Design the report structure - Planning for the next meeting	10:00 PM	2H	All
20 October	- Finalize the evaluation of the papers - Divide the paper between the team members to check the summery - Edit the network diagram - Planning for the next meeting	9:00 PM	2H	All
21 October	- Continue the work on the paper summery - Add the references to the report - Planning for the next meeting	9:30 PM	3H	All
22 October	- take Dr. Asif feedback and edit the report according to it - review and edit the report content - add the WBS to the report - design the presentation slides - submit the report and presentation 1 - Planning for the next meeting	6:00 PM	6H	All
25 October	- discuss about the report 2 - Planning for the next meeting	10:00 PM	1H	All
28 October	- discuss about the report 2 - prepare the questionnaire - Planning for the next meeting	9:00 PM	2H	All
1 November	- Write the changes in the project - Write the methods to find the information - Planning for the next meeting	10:00 PM	2H	All
3 November	- Finalize the questioner - Identify the skills we need to learn - Planning for the next meeting	10:00 PM	2H	All
4 November	- Write the final requirements - Planning for the next meeting	10:00 PM	2H	All
5 November	- Write the final requirements - Planning for the next meeting	6:45 PM	2H	All
8 November	- Review the final requirements - Planning for the next meeting	10:00 PM	1.5H	1740166 1742589 1740373
9 November	- Use case design - Planning for the next meeting	6:30PM	2H	All
10 November	- Use case tables - Class diagram design - Planning for the next meeting	10:00 PM	2H	AII

Continue…

…Continuation				
11 November	- Update use case diagram and tables - Class diagram design - Planning for the next meeting	9:30 PM	2.5H	All
15 November	- Class diagram design - Planning for the next meeting	9:30 PM	2.5H	All
16 November	- Finalize class diagram - Planning for the next meeting	9:30 PM	2.5H	1740166 1742589 1742403
17 November	- Create the sequence diagram - Create architecture diagram - Planning for the next meeting	9:30 PM	2.5H	All
18 November	- Create ER diagram - Update use case diagram - Update class diagram - Planning for the next meeting	9:30 PM	3 H	1740166 1742589 1742403
19 November	- Write and design the report 2 - Design the presentation 2 - Planning for the next meeting	5:00 PM	6H	All
22 November	- Planning for final report and poster - Planning for the next meeting	9:00 PM	1H	1740166 1742589
24 November	- Write Final report - Planning for the next meeting	6:00 PM	3H	1740166 1742589
25 November	- Write Final report - Planning for the next meeting	9:00 PM	3H	1740166 1742589 1740373
26 November	- Write Final report - Planning for the next meeting	9:00 PM	3H	1740166 1742589
28 November	- Write Final report - Design the final report - Planning for the next meeting	9:30 PM	2.5H	1740166 1742589 1742403
29 November	- Write Final report - Design the final presentation - Planning for the next meeting	9:30 PM	2.5 h	All
30 November	- Write Final report - Design the final presentation - Planning for the next meeting	9:30 PM	2H	1740166 1742589 1740373
1 December	- Review the final report - Practice for the final presentation	9:00 PM	3H	All

Table 47: Meeting table CPCS499

CPCS-499 Meeting Schedule				
DATE	What has been done	Start	Duration	Attendance
24 January	- Search for a useful information about computer vision. - Begin working on GUI design. - Agree on the meeting schedule. - Planning for the next meeting	9:30PM	2h	All
26January	- Searching for Information about computer vision - Working on the GUI design - Planning for the next meeting	9:30PM	2.5h	all
27 January	- Preparing for the KAUWEP contest - Planning for the next meeting	9:30PM	3h	1742589 1742403 1740373

Continue…

…Continuation

31 January	- Design the App GUI (main Screens) - Divide the work among the team members. - Planning for the next meeting	9:30PM	2h	all
2 February	- Design the App GUI (main Screens) - Start Building the dataset for the training model. - Start writing the first report (Q1&2) - Planning for the next meeting	9:30PM	2.5h	all
3 February	- Start Design the App GUI (malty user) - Building the dataset for the training model. - Start writing the first report (Q3) - Planning for the next meeting	9:30PM	2.5h	all
7 February	- collecting the data - we started on developing the networking part. - Planning for the next meeting	9:30PM	1.5h	1742589 1740166 1740373
9 February	- Working on the GUI design - Working on report 1 - Working on the presentation for report 1 - Planning for the next meeting	9:30PM	1.5h	all
10 February	- writing the first report (Q4) - Working on the presentation for report 1 - Planning for the next meeting	9:30PM	2.5h	all
14 February	- Finalize report 1. - Finalize presentation 1	9:30PM	2.5h	all
16 February	- Report 1 Completed. - Presentation 1 Completed. - Planning for the next meeting	9:30PM	2.5 h	all
17 February	- Prepare for the first presentation. - Planning for the next meeting	9:30 PM	1.5 h	all
21 February	- Review the code. - Test some functional requirement. - Discus about learning and testing process. - Planning for the next meeting	9:30 PM	2.5 h	all
23 February	- Work on the learning model - Start working integrate the learning model with Android - Planning for the next meeting	9:30 PM	2.5h	1742589 1740166 1740373
24 February	- Review the GUI design - Planning for training part in multi-user mode - Planning for the next meeting	9:30 PM	1.5 h	All
28 February	- Review the learning model. - Integrate the ML model with the android system. - Planning for the next meeting	10:30 PM	1.5 h	All
3 March	- Integrate the ML model with the android system. - Planning for the next meeting	9:00 PM	2 h	All
7 March	- Design database structure - Integrate the ML model with the android system. - Planning for the next meeting	9:30 PM	3.5 h	All
9 March	- Use google cloud to train the model. - Implement the database. - Planning for the next meeting	9:30 PM	2.5 h	All
10 March	- Implement the database. - Planning for the next meeting	9:30 PM	2.5 h	All

Continue…

…Continuation				
14 March	- Integrate the database with the android system. - Start working on the second report. - Planning for the next meeting	9:30 PM	2 h	All
16 March	- Writing report 2 - Planning for the next meeting	10:00 PM	2h	1740166 1742403 1740373
17 March	- Integrate the database with the android system. - Planning for the next meeting	9:30 PM	2.5h	All
21 March	- Integrate the database with the android system. - Writing report 2 (Summery of previous work) - Planning for the next meeting	9:30 PM	3h	All
23 March	- Integrate the database with the android system. - Writing report 2 (implementation progress) - Planning for the next meeting	9:30 PM	3h	All
24 March	- Integrate the database with the android system. - Writing report 2 (Testing) - Planning for the next meeting	9:30 PM	2h	All
28 March	- Writing report 2 - Working on the presentation for report 2 - Planning for the next meeting	9:30 PM	3h	all
29 March	- Writing report 2 - Working on the presentation for report 2 - Planning for the next meeting	9:30 PM	2h	all
30 March	- Strat writing the final report(testing). - Redesign ER diagram - Redesign sequence diagram - Planning for the next meeting	9:30 PM	3h	all
31 March	- Review the code. - Redesign the app GUI interfaces. - Planning for the next meeting	9:30 PM	2.5h	all
3 April	- Create the code to make Flashlight send Morse Code signals - Integrate the learning model with the system. - Planning for the next meeting	9:30 PM	2 h	1740166 1742403 1740373
4 April	- Writing the final report (Implementation) - Creating java documentation - Redesign sequence diagram - Planning for the next meeting	9:30 PM	2.5 h	1742589 1742403 1740166
5 April	- Writing the final report (Implementation) - Creating java documentation - Redesign Class diagram - Strat design the final presentation. - Planning for the next meeting	7:00PM	5 h	All
6 April	- Writing the final report (Implementation) - Finish java documentation - Finish class diagram - Planning for the next meeting	7:00 PM	5h	1742589 1742403 1740166
7 April	- Finish and review the report and presentation	7:00 PM	5h	All