SMARTCOP – AN AUTOMATED PLATFORM TO MITIGATE THE IMPACT OF ROAD ACCIDENTS

2020 - 052

Shangavie R.

(IT16167124)

BSc (Hons) in Information Technology

Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

September 2020

SMARTCOP – AN AUTOMATED PLATFORM TO MITIGATE THE IMPACT OF ROAD ACCIDENTS

2020 - 052

Shangavie R.

(IT16167124)

Dissertation submitted in partial fulfilment of the requirements for the Bachelor of Science (Hons) in Information Technology Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

September 2020

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology, the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Name	Student ID	Signature
Shangavie R	IT16167124	

The above candidate is carrying out research for the undergraduate dissertation under my supervision.

Signature of the supervisor

[Dr.Windya Rankothge] Date

ACKNOWLEDGEMENT

This project is done to achieve the 4th year research requirements of the subject of the Comprehensive Design Analysis Project. Encouragements, supports and guidance given to me by others are great value. Therefore, I would like to express my deep gratitude to all who has supported me to complete this significant task.

I am deeply thankful to Dr.Janaka Wijekoon, the lecture-in-charge of the CDAP module, Dr.Windya Rankothge, the supervisor of the project, Miss. Narmadha Gamagae, the co-supervisor of the project and Mr.Laksiri geethal, the senior superintendent of police for the suggestions, constant encouragement and support in the development of this research, particularly for the many simulating and instructive discussions. Hereby, I would thank all CDAP team members, group members, colleagues and friends for all their help, support, interest and valuable advices. Finally, I would like to thank all others whose names are not listed particularly but have given their provision in many ways and encouraged me to make this success. I will forever be thankful to my former lectures for supporting me during these past four years.

Abstract

Knowledge in post road accident responsibility is vital to everyone in terms of helping people in emergency situations since injuries that happen during road accidents are one of the leading causes for mortality and morbidity locally as well as globally. As a developing country, the road accident has become a huge problem for Sri Lanka also socially and economically. The Sri Lankan police team and rescue team are conducting many seminars about How should respond when road accident occurs in front of us? But they are bit primitive to provide a better solution to improve the road accident response awareness. Therefore, a game based mobile application has been developed to improve the awareness. According to the responses that were obtained through the survey, a quiz game has been developed. Furthermore, questions in the quiz were predicted based on the player's performance using random forest classifier algorithm in machine learning with 87.5% accuracy. And also time that needs to be given for question was predicted using the same algorithm and same set of features which were used for question prediction with 100% accuracy. Flask and Heroku has been used to connect both machine learning model and flutter mobile application.

Keywords: game based, mobile application, random forest classifier algorithm, machine learning, post road accident

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

Abbreviation	**Description**
API	Application Programming Interface
GUI	Graphical User Interface
ML	Machine Learning
RFC	Random Forest Classifier
SVM	Support Vector Machine
UI	User Interface

1. INTRODUCTION

1.1. Background Context

One of the major problems that are faced by many countries in the world is road traffic accidents. Road accidents not only affect the countries' economy but also people's lives. According to the Global status report on road safety 2018, released by World Health Organization (WHO) in December 2018[1] the yearly road traffic deaths has increased to 1.35 million. The main reason for road accidents is violating the traffic rules such as over speeding, drunken driving, driving without wearing seat belts. Road accidents can be reduced by following traffic rules perfectly.

However, every person in the world needs post road accident related responsibility knowledge to recover themselves and others from injuries and protect vehicles from crashes caused by road accidents. This knowledge also helped to alert other vehicles on the road during an accident. Police officers, Red Cross team, Rotary club and other clubs have conducted seminars and training programs to give knowledge on post road accident responsibility to people. However these methods did not provide significant results.

In order to improve the result, a survey was conducted to select the best method to increase the responsibility knowledge among people. The Figure 1 shows the survey result. Based on the responses we got most of the people chose the picture quiz option. According to that result the SmartCop- Safe Life Quiz game has been implemented. The implemented game contains video, audio and article based questions. Users will be allowed to attempt the game according to their knowledge. The machine learning algorithm is used to decide the game levels dynamically. Game levels are decided based on the player's skills.

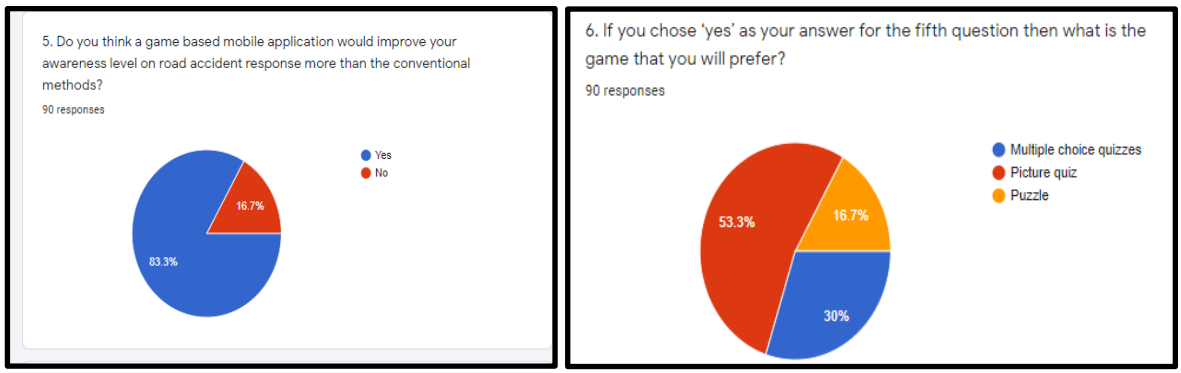

Figure 1: Survey results

1.2. Background Literature

There is massive amount of researche conducted in the field of first aid and game based learning. The research conducted in Dubai[2] is presented the appropriateness of a virtual environment for training traffic investigators of Dubai police force. The Torque game engine has used to build the game. In order to assess the result, the police officers were categorized into two groups as Novices and Experienced. The restriction placed in this study was that every participant in the experiment must have computer knowledge. To implement the game, 3D Studio Max was used to create a game world, interfaces of game were created using interface builder which was provided by Torque, OntRAT was used to create game model and game behaviour through API or Python scripting language. The research result shows that, game based learning given significant result than lectures and on-job- training methods, and performance of novices was higher than the performance of experienced group.

A 3D serious game for the first aid emergency named as FASim aims to learn the first aids [3]. The FASim game focused on the interaction between players, the adaption of the game to the profiles of the players, importance of awareness and prevention and finally makes quick and accurate decisions. And the solutions of FASim are based on the agent-based simulation. Development of game-based learning application for first aid awareness research aims to instruct on how to handle a situation and act on the situation when encounter minor accidents[4]. This system apply Gale Shapley algorithm for taking input as a preference outline and outputs a matching of different objects needed in the accident, and it uses Agile Model for development.

Game-based mobile platform to instruct first aid techniques study make first aid learning procedure with realistic models and scenarios[5]. This mobile application developed using a motion capture system, three-dimensional modelling and game engine technologies. It contains Simulation Courses, Applied Games and Examinations as its main modules. In Simulation Courses, a realistic simulation of a first aid scenario is provided to the user and the second section is an interactive environment in which users apply the information they learn in the Simulation Courses module and finally the learning levels of the players are measured in a simulation environment in the Examinations section.

The intention of the study[6] was assesses the efficiency of a web based learning environment to improve e-safety skills of children. For experiment, three primary schools with computer lab facility were selected and the research study followed a quasi-experimental, pre-test, post-test control group design. The selected participants were categorized into two groups as

experimental group who used the web-based learning environment and control group who did not use the web-based learning environment. The result shows, the higher performance result in the experimental group when compared to the control group.

1.3. Research Gap

Even the study conducted in Dubai to assess the efficiency of game environments to train police traffic accident investigators[1] give the significant result, the proposed environment does not provide considerable difference in the navigational patterns between experienced personnel and novices. Currently the study on FASim 3D serious game[2] focuses only on cardiac arrest. So the FaSim game needs to focus on other cases and wants to implement the game in other languages as well. Additionally, add a group distribution strategy to the multiplayer mode to make the game more interesting.

In study [5] the involvement of persons, who play a major role in younger's e-safety skills improvement such as teachers, parents need to be improved to get a better result. The research[7] needs to be conducted with a large range of risky driving measures. According to the Table 1 result only the First Aid Quiz[8] displays the score that player scored during the game, other games[9],[10] and [11]did not display the scores. However the available games contain questions in English. So, the people who have less knowledge on English cannot attempt the game properly. For that the games need to be updated to support multiple languages.

Table 1: Comparison between implemented system and existing systems

Features	First Aid Quiz [8]	First Aid MCQ [9]	First Aid Ireland Pop Quiz [10]	First Aid Self-Test [11]	SmartCop – Safe Life Quiz (implemented application)
Measure the awareness level of player using their score	✔	**X**	**X**	**X**	✔
Measure the time that a player spent on each question to answer	**X**	**X**	**X**	**X**	✔
Maintaining a leader board	✔	**X**	**X**	**X**	✔
Dynamic Difficulty Adjustment in game levels.	**X**	**X**	**X**	**X**	✔

1.4. Research Problem

Road accidents can happen anytime in anywhere for anyone. One of the main reasons for the higher number of injuries and deaths caused by road accident is lack of awareness on post road accident responsibility. Even many traditional methods such as seminars and training programs conducted by Police department and rescue team do not provide significant improvement in responsibility knowledge. There are games such as quiz games and video games also available in the Google play store to improve first aid knowledge. However, most of the quiz games contain default questions. So, players play with same set of questions every time. As a result, player's interaction on the game might reduce and cannot measure the awareness level of the player perfectly. Therefore need to implement the game with dynamically adjustment levels. Evaluate the supervised learning algorithms to implement the dynamically adjustment level feature in the game.

Research questions derived from the research problem

- How post road accident responsibility knowledge will be improved through game-based learning?
- How to generate dynamic difficulty levels in quiz game using machine learning algorithms?

1.5. Research Objectives

1.5.1. Main Objective

To develop a game-based mobile application with dynamic difficulty level adjustment to improve the post road accident responsibility knowledge.

1.5.2. Specific Objectives

- To predict questions based on the player's performance using ML
- To predict time that need to given for the next question to answer based on the player's performance using ML

- To display obtained score based on the given input provided by the player

- To display player's position in the leader board

2. METHODOLOGY

2.1. Methodology

2.1.1. System overview

The SmartCop – Safe Life Quiz game is an android mobile application which can be accessible to the public through Google play store. The application is implemented using android studio with flutter. The scores of the players will be stored in the firebase database. The main feature of this game is questions will change dynamically by analyzing the player's performance. The current question type, answer status, time given by the system to answer the question and time taken by the player to answer the question are the main features that were considered to predict the next question and time needed to be given by the system for answering the next question. For prediction purposes RFC algorithm was used. RFC algorithm is a supervised learning algorithm. It predicts the next question with 87.5% accuracy and predicts the time needed for the next question to answer with 100% accuracy. The ML result and flutter application is integrated using Flask and Heroku. The Figure 2 explains the system overview.

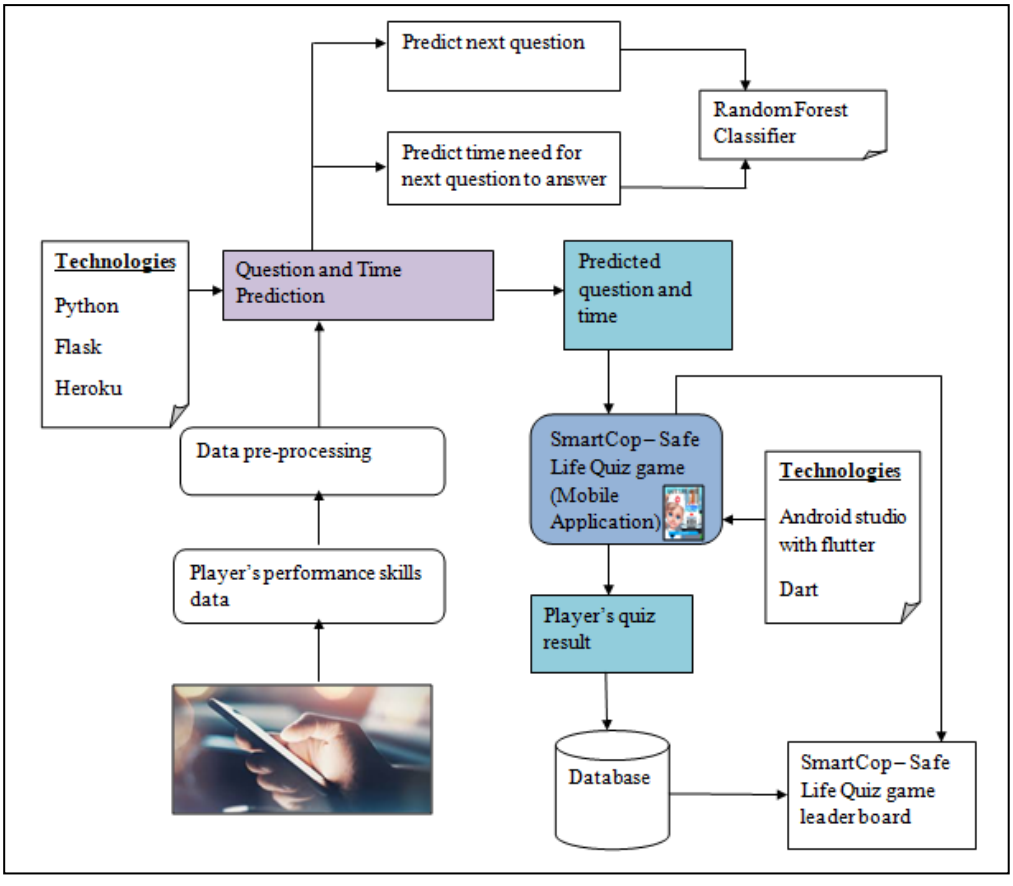

Figure 2: System Overview Diagram – Safe Life Quiz game

2.1.2. Feasibility study

The SmartCop – Safe Life Quiz mobile game has been implemented as a solution to improve post road accident responsibility knowledge among the public with dynamic difficulty adjustment level feature. It is a major requirement from the public because currently available quiz games contain default question levels.

The implementation and maintenance of the application is identified as feasible by analysing development time and cost, and maintenance time and cost. However, hosting the application on Google play store and adding or updating features charges a small cost.

The tools and technologies used in the application are freely offered in the beginning stage.

- Android studio with flutter
- Firebase
- Heroku

2.1.3. Functions of the application

a. Login process using Google account

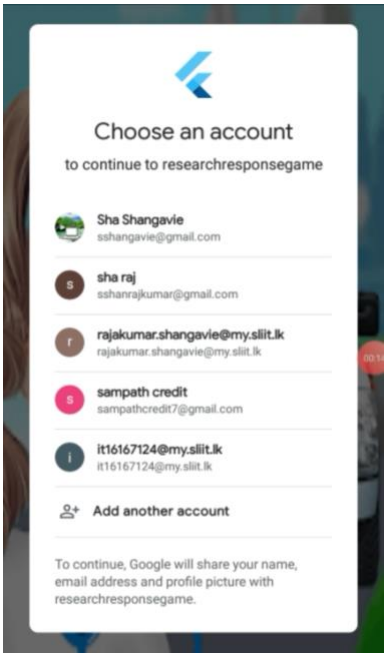

Figure 3: Login screen of the SmartCop – Safe Life Quiz game

Figure 4: Firebase authentication

For a first time the player wants to sign in to the game using any one of their Google accounts and then continues the game. After that the player can play the game any time without login again until sign-out from the game. Firebase authentication was used for this purpose. Figure 3 shows the login screen of the application and Figure 4 shows how login details stored in the firebase authentication.

b. Dashboard

After successful login, the game will move to the dashboard. The dashboard contains play quiz, user profile, leader board and about sections. Figure 5 shows the dashboard of the game.

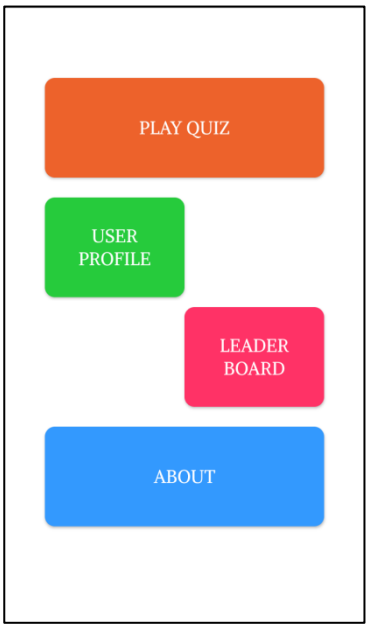

Figure 5: Dashboard of Smartcop – Safe Life Quiz game

c. Play quiz game

If the player clicks on the “PLAY QUIZ” button then the game will move the player to quiz mode page. The game contains three types of quizzes named as video based quiz, audio based quiz and article based quiz. Players can play the game according to their preference. To improve the player’s interaction on the game, an application will select a video, audio and article randomly from the pool and display it. So, the player gets different video, audio clips and articles every time when they attempt the game. Figure 6 shows the quiz mode page.

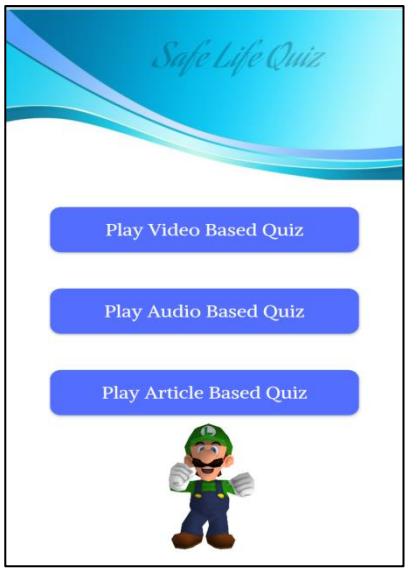

Figure 6: Quiz modes of Smartcop – Safe Life Quiz game

All three question modes contain multiple choice questions with four options, multiple choice questions with three options, true/false question and fill in the blank question. In video based quiz mode, the question will be displayed after the video clip played completely. In an audio and an article based quiz mode, users can move to the question at any time. Figure 7, Figure 8, Figure 9 displays the video based quiz page, audio based quiz page and article based quiz page respectively.

Figure 7: Video based quiz

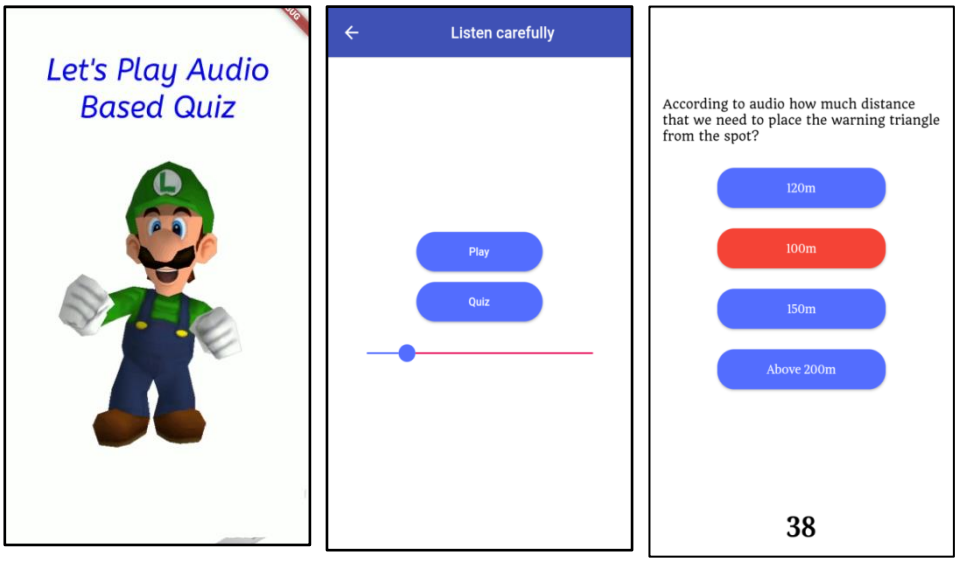

Figure 8: Audio based quiz

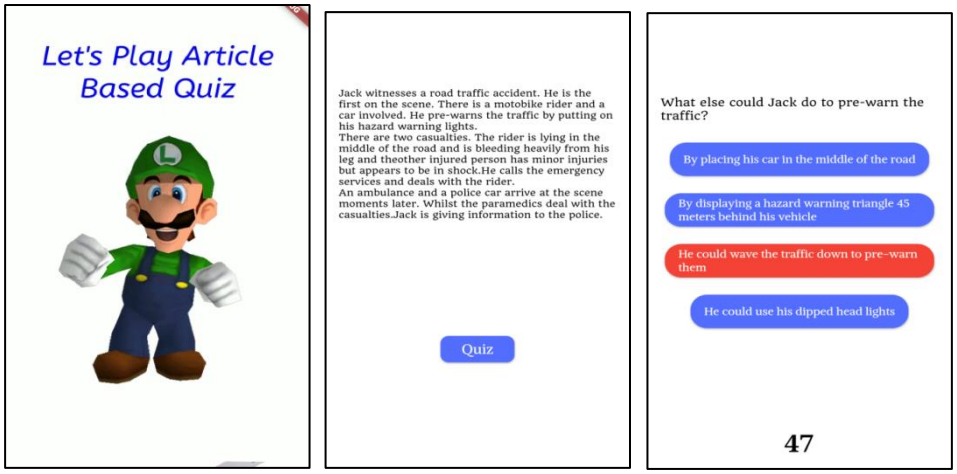

Figure 9: Article based quiz

The Scores of each section will be displayed in the result page after the player completes the quiz. Then those marks and player names will be stored in the firebase database with unique a user id. The Google account user id has been used as a unique id. The player can sign-out from the game and sign-in again using another account to play in the result page. The player cannot go back to the previous question during the game because the questions will generate based on the performance of the player in the previous question. Figure 10 shows the result screen of the game.

Figure 10: SmartCop – Safe Life Quiz game result screen

d. View leader board

If a player clicks on the “LEADER BOARD” option in the dashboard then he/she can view the rank based on the score they gained. The leader board displays rank, name of the user, score they scored. The name of the user and scores are retrieved from the database and stored to the list and then displayed in the descending order of the marks. In the leader board first three ranks will be displayed with different colour crowns and other ranks will be displayed as usual. Players can sign-out from the application in the leader board. Figure 11 shows the leader board of the game.

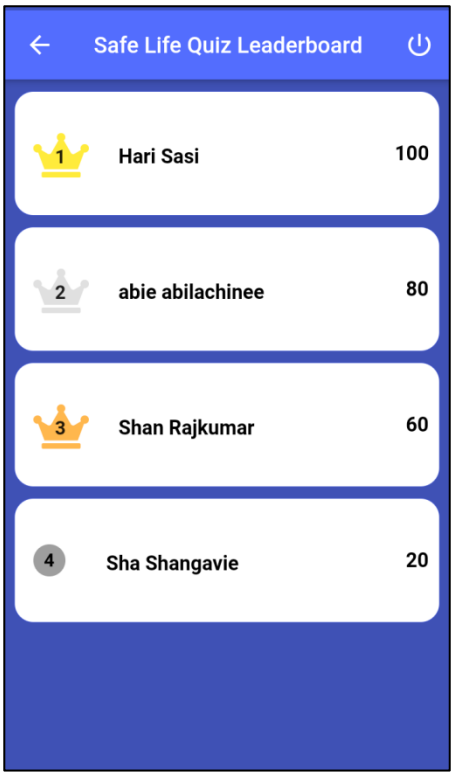

Figure 10: SmartCop – Safe Life Quiz game leader board screen

e. View user Profile

The player redirects to the user profile section when they click on the “USER PROFILE” button in the dashboard. Players can view the user name, email and user id they used to log-in to the application. The name, email and user id are obtained when they log-in to the application. The user profile shows the profile picture of the Google account used by the player to login the game. The player has a sign-out option in the user profile to log-out from the game. Figure 12 shows the user profile of the game.

Figure 12: SmartCop – Safe Life Quiz game user profile screen

f. View About Section

The player redirects to the about section of the game when they click on the "ABOUT" button in the dashboard. Users can gain information and rules of the game in the about section. So, it will help the player to get an idea about the game before they start to play. And also it displays the activities that the logged player can do. The sign-out option is available for players in the about section. Figure 13 shows the about section of the game.

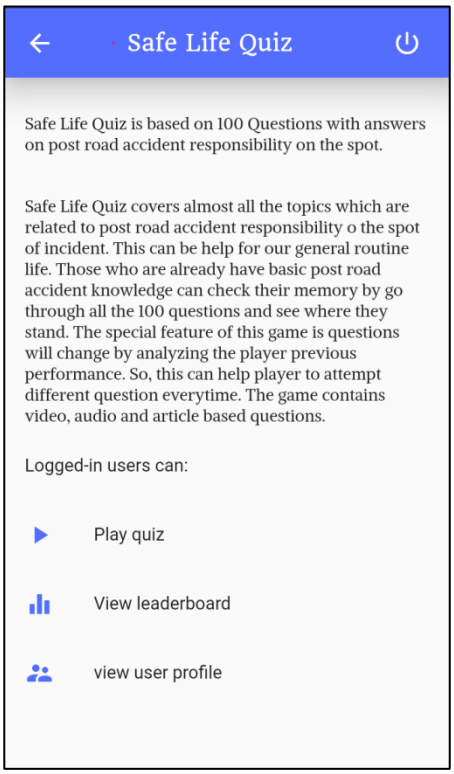

Figure 13: SmartCop – Safe Life Quiz game about screen

2.1.4. Application literacy

The SmartCop – Safe Life Quiz mobile game is a fully English based quiz game. The game will be available in Sinhala and Tamil language in future versions. Player needs an android mobile with good storage because the application is an android mobile application. And also, the game needs a good internet connection to generate questions.

2.1.5. Required resources

Software requirements

1. Android studio

Developers develop applications for android devices using Android studio software. It is available on Windows, Linux and macOS. It is open source and easy to use. In order to work with android studio computers need a minimum of 4GB RAM. Android studio has been used for both frontend and backend development in this project.

2. Flutter

Flutter is an open source and free SDK released by Google. It uses single code base to implement applications for both Android and IOS. It can be installed in Android Studio and IntelliJ IDEA. Flutter uses Dart language for implementation. Flutter becomes popular because it supports hot reload, expressive and flexible user interface animation, and single language for backend and layout.

3. Anaconda Navigator and Jupyter Notebook

Anaconda Navigator is a desktop graphical user interface which allows launching applications and simply managing environments, channels and conda packages without using command- line commands. Jupyter Notebook, JupyterLab, PyCharm are some applications available in anaconda navigator. In this project Jupyter Notebook was used to implement machine learning model.

4. Flask

Flask is a web application framework written in python. In this project flask was used as a backend for API to connect a machine learning model with a mobile application.

5. Firebase

Firebase is a realtime database that allows us to store and sync data in realtime. It is a platform created by Google for developing web based and mobile based applications. In here firebase has been used for user authentication and store player's marks.

6. Heroku

Heroku is a Platform as a Service (PaaS) used to manage, deploy and scale modern applications. It is very flexible and easy to use.

Hardware requirements

- A PC or laptop with minimum 4GB RAM and proper internet connection
- An android mobile with good storage and proper internet connection

2.2. Implementation

2.2.1. Login using Google account

Import firebase_auth.dart and google_sign_in.dart to sign in to the application using Google account. Figure 14 shows the imported packages.

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:google_sign_in/google_sign_in.dart';
```

Figure 14: Import packages for login

```
Future<bool>loginWithGoogle() async {
  try{
    GoogleSignIn googleSignIn = GoogleSignIn();
    GoogleSignInAccount account = await googleSignIn.signIn();
    if(account == null)
      return false;
    AuthResult res = await _auth.signInWithCredential(GoogleAuthProvider.getCredential(
        idToken: (await account.authentication).idToken,
        accessToken: (await account.authentication).accessToken
    ));
    if(res.user == null) {
      return false;
    }
    else {
      print(res.user.displayName.toString());
      logindata.setString('userName', res.user.displayName.toString());
      Navigator.push(context, MaterialPageRoute(
          builder: (context) => DashBoardScreen(user: res.user))); // MaterialPageRoute
      return true;
    }
  }
  catch(e){
    print("Error login with google");
    return false;
  }
}
```

Figure 15: Method to login using Google account

Based on the method implemented in the Figure 15, method return false if the account is null or user is null or any errors found during login. Method returns true if the login procedure is done successfully.

2.2.2. Play background music in the game

Background music will play in the welcome page of the SmartCop – Safe Life Quiz game when the player starts the game. The assets_audio_player package has been used in the flutter to play an audio. Figure 16 shows the implementation of playing background music in an application.

```
AssetsAudioPlayer _assetsAudioPlayer;
@override
void initState() {
  super.initState();
  _assetsAudioPlayer = AssetsAudioPlayer();
  _assetsAudioPlayer.open(
    Audio("assets/startup.mp3"),
  );
  _assetsAudioPlayer.playOrPause();
}

@override
void dispose() {
  _assetsAudioPlayer = null;
  super.dispose();
}
```

Figure 16: Play background music using assets_audio_player package

2.2.3. Play video clip for video based Quiz

The video_player package has been used to play video clips in flutter. In the application the next screen will be displayed after the video played completely. Figure 17 shows the implementation of playing video clips in an application.

```
VideoPlayerController _controller;
int duration;
void initState(){
  super.initState();
  _controller = VideoPlayerController.asset('video/videobased.mp4')
    ..initialize().then((_){
      setState(() {
        duration=_controller.value.duration.inSeconds;
        print(duration);
        Timer(Duration(seconds:duration),(){
          Navigator.of(context).pushReplacement(MaterialPageRoute(builder: (context) =>VideoPage(),));
        }); // Timer
      });
    });
  _controller.play();
  _controller.setLooping(false);
}
```

Figure 17: Play video clip using video_player package

2.2.4. Predict next question and time using machine learning algorithm

Table 2: Characteristics of dataset

Number of instances	30
Number of attributes	06
Number of classes	04
Number of missing values	Null
Related task	Classification

Table 3: Attributes description of dataset

Attributes	Description	Domain
CureentQuestion	Type of the current question	1 – multiple choice question with 4 options 2 – multiple choice question with 3 options 3 – true/false question 4 – fill in the blank question
TimeGiven	Time given by the system to answer the question (in seconds).	--
TimeTakenToAnswer	Time spent by player to answer the question	1 - 40 sec < Time < 50sec 2 - 20 sec < Time < 40 sec 3 - 0 sec < Time < 20s sec
AnswerStatus	Status of the answer	0 – wrong answer 1 – correct answer
NextQuestion	Target class (Tyee of the next question)	1 – multiple choice question with 4 options 2 – multiple choice question with 3 options 3 – true/false question 4 – fill in the blank question
TimeNeedToGiven	Target class (Time that need to given for the next question to answer)	--

Table 2 describes the characteristics of the dataset which was used in the prediction task. The dataset was stored in .csv file format. The Table 3 describes the description and domain of each attribute in the dataset. CurrentQuestion, TimeGiven, TimeTakenToAnswer and AnswerStatus were used to predict the next question type and time that need to be given for a next question to answer. As a first step the dataset was pre-processed to check whether the dataset is cleaned or not. It displayed 0 for the number of null values.

Supervised learning algorithm was decided to be used for prediction purposes. Because the labels are known in the training dataset. The RFC algorithm was selected to predict the next question type and time needed for the next question to answer. Because the RFC algorithm works well with a small dataset, more flexible and easy to use.

The pandas library was imported to read the dataset. The train_test_split function was used to split the dataset as test dataset and train dataset. The RandomForestClassifier is imported because the dataset uses RFC algorithm for prediction activity. The sklearn.metrics was imported to calculate the accuracy of the outcome. A joblib was imported to access both machine learning model and column names in flask server. The Figure 18 shows the list of libraries imported for prediction tasks.

```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.ensemble import RandomForestClassifier
4 import sklearn.metrics as sm
5 import joblib
6 import json
```

Figure 18: List of imported libraries

```
#Read the dataset
dataset = pd.read_csv('E:\\GameDataset.csv')
```

Figure 19: Read the dataset from the given location

	CurrentQuestion	TimeGiven	TimeTakenToAnswer	AnswerStatus	NextQuestion	TimeNeedToGiven
count	30.000000	30.000000	30.000000	30.000000	30.000000	30.00000
mean	2.800000	43.000000	2.000000	0.500000	2.966667	40.00000
std	1.186127	5.186189	0.830455	0.508548	1.066200	7.19195
min	1.000000	35.000000	1.000000	0.000000	1.000000	30.00000
25%	2.000000	40.000000	1.000000	0.000000	2.000000	35.00000
50%	3.000000	45.000000	2.000000	0.500000	3.000000	40.00000
75%	4.000000	45.000000	3.000000	1.000000	4.000000	45.00000
max	4.000000	50.000000	3.000000	1.000000	4.000000	50.00000

Figure 20: Statistics summary of the dataset

The Figure 19 shows a dataset read from a given location using read_csv method in the pandas library and Figure 20 describes the statistics summary of the dataset. Figure 21 shows that column 1 to column 4 was defined as features (independent variables) and the 5th column (target columns) was defined as the predictQuestion. The 5th is a dependent variable.

Figure 22 shows that the original dataset is split into train dataset and test dataset in the 75:25 ratios respectively using train_test_split method. After that, the random forest classifier algorithm was used to predict the output. The training model and column names have been saved as .pkl files using dumb function of joblib library. The Figure 23 shows the steps to store model and columns as .pkl file. The same code base has been used to predict time. The ML model code for predict questions and predict time is attached in appendix. (Appendix A – predicts question implementation, Appendix B – predicts time implementation)

```
features =  dataset.iloc[:,0:4].values
predictQuestion = dataset.iloc[:,4].values
```

Figure 21: Split the variables as independent and dependent variables

```
X_train,X_test,Y_train,Y_test = train_test_split(features,predictQuestion,test_size = 0.25, random_state = 20)

classifier_Question = RandomForestClassifier(n_estimators =35, criterion ='entropy',random_state = 35)

res = classifier_Question.fit(X_train,Y_train)


Y_pred = classifier_Question.predict(X_test)
```

Figure 22: Split the dataset into train dataset and test dataset then predict the result

```
joblib.dump(res, 'questionmodel.pkl')
cols=["CurrentQuestion","TimeGiven","TimeTakenToAnswer","AnswerStatus"]
joblib.dump(cols, "questionmodel_cols.pkl")
```

Figure 23: Store training model and column names as .pkl file

2.2.5. Connect machine learning results with mobile application.

Creating a Flask server

Here Flask is used as a backend for API. Figure 24 shows the code used for following activities.

- The POST request from the flutter application will be sent to the "/predict" route.
- The column names (.pkl files) and ML model are fetched and used for predicting the next question.
- The predicted question is returned in the JSON format.

Flask server code for both models has been attached in the appendix. (Appendix C – Flask server code for predicting question, Appendix D – Flask server code for predicting time)

```
from flask import Flask, jsonify, request
import pandas as pd
import numpy as np
import joblib
import traceback
from flask_restful import reqparse
app = Flask(__name__)

@app.route("/", methods=['GET'])
def hello():
    return "hey"

@app.route('/predict', methods=['POST'])
def predict():
    lr = joblib.load("questionmodel.pkl")
    if lr:
        try:
            json = request.get_json()
            model_columns = joblib.load("questionmodel_cols.pkl")
            temp=list(json[0].values())
            vals=np.array(temp)
            prediction = lr.predict(temp)
            print("here:",prediction)
            return jsonify({'prediction': str(prediction[0])})

        except:
            return jsonify({'trace': traceback.format_exc()})
    else:
        return ('No model here to use')
```

Figure 24: Code snippet for Flask server

Deploying the API

Heroku has been used to deploy a flask app. The Procfile and requirements.txt file are important to deploy a flask app. A requirements.txt file contains the dependencies of the app and the Procfile contains the commands that are executed on app startup. Then push Procfile, requirements.txt, flask server code(app.py) and .pkl files(questionmodel.pkl, questionmodel_cols.pkl) to a github repository. Figure 25, Figure 26 and Figure 27 shows the Procfile, requirements.txt and github repository respectively.

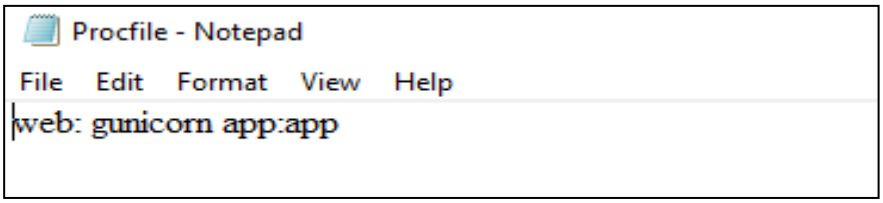

Figure 25: Procfile

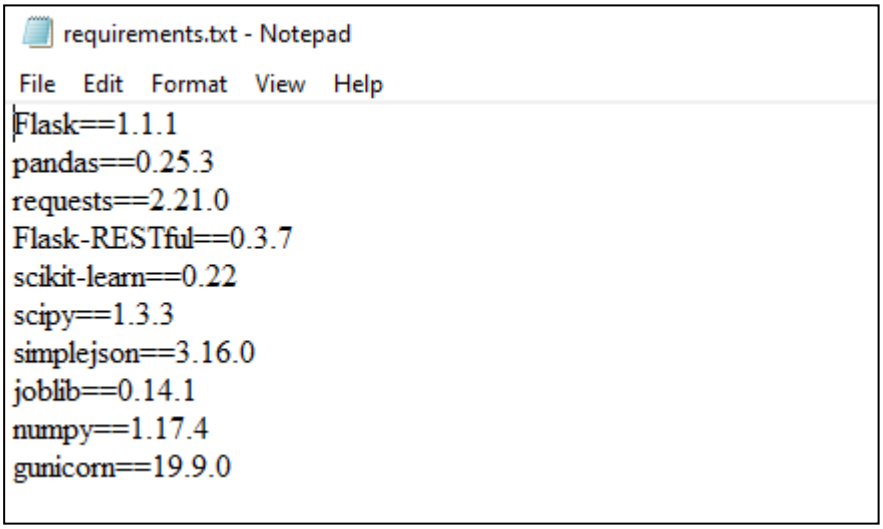

Figure 26: requirements.txt file

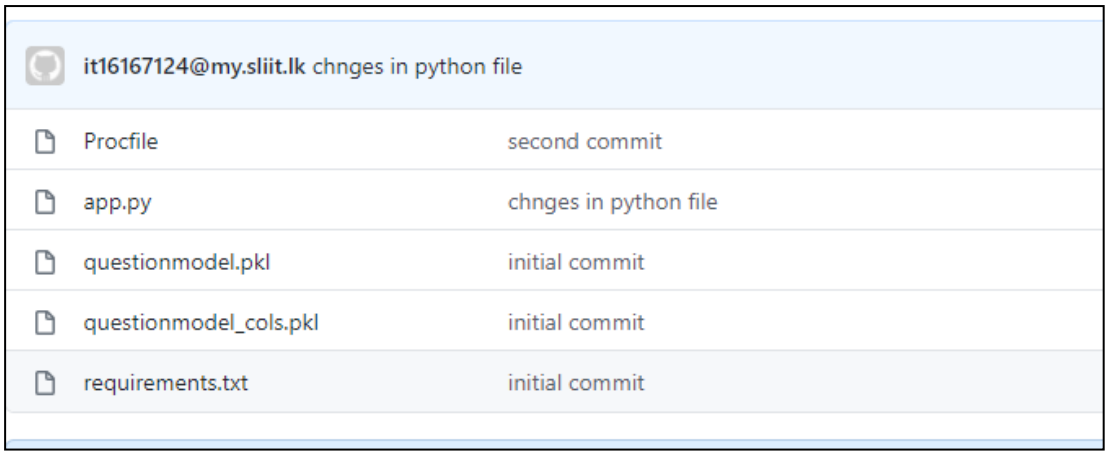

Figure 27: Github repository

Then, deploy the repository branch in Heroku and get the generated URL. Finally use that URL in the flutter application to connect both the machine learning model and the flutter mobile application. Figure 28 explains the dart code written to connect both ml model and flutter application using URL generated by Heroku.

```
Future<http.Response> predictDifficultyLevel(var body) async {

  try {
    var url = "https://research-1-app.herokuapp.com/predict";
    //encode Map to JSON
    var body1 = jsonEncode(body);
    print(body1);
    var response = await http.post(url, headers: {"Content-Type": "application/json"}, body: body1);
    print(response);
    print("${response.statusCode}");
    print("${response.body}");
    var responsebody = response.body;

    predictedDifficulty = responsebody
        .toString()
        .replaceAll("{", "")
        .replaceAll("}", "")
        .split("\"prediction\":")
        .removeLast();
    print("*****************************");
    print(predictedDifficulty);

    return response;
  } catch (e) {
    print(e);
  }
}
```

Figure 28: Dart code snippet to connect ml model and flutter application using URL generated by Heroku

2.3. Commercialization Aspects of the Product

The process of introducing brand new products or services to market is called commercialization. The SmartCop-Safe Life Quiz game is a sub part of the SmartCop application. This game improves post road accident responsibility knowledge among the public. Since there are many first aid related quiz games available in Google play store. However, those games contain default question levels. But the SmartCop- Safe Life Quiz game contains dynamic difficulty adjustment level as a special feature when compared to other available quiz games. So, the player can measure their knowledge on post road accident responsibility more accurately than other quiz games. SmartCop-Safe Life Quiz game is available through Google play store. So, the user can download an application and use a one month trial package for free. The player needs to activate the game again by paying money online after one month of free trial package.

The player gains following benefits by playing SmartCop – Safe Life Quiz game,

- The questions will be generated dynamically based on the player's performance in the previous question.
- Compare player's knowledge level with others using leader board
- Measure the time spent for each question to answer.

Currently SmartCop – Safe Life Quiz game is available for android users only. However, in future the game will be implemented for ios users as well with more new features. The amount of money the player needs to pay will increase slightly when the system updates with more features in future.

2.4. Testing

In software development life cycle testing phase plays a main role. Software testing is very useful to detect bugs during the development of the system so it will help to increase the quality of the software product. Testing should be performing for each and every segment that developed to improve the product quality. The software testing can be done as unit testing, integration testing, system testing and user acceptance testing.

- Unit Testing

Unit testing should be done for a piece of code in the component to ensure that the code works as expected. In this project unit testing is done to make sure the following activities are working as expected.

- ✓ Login procedures
- ✓ Video, audio and article are displayed as expected
- ✓ Answer colours are changed according to the player selection
- ✓ The result page displays the correct marks based on the user inputs
- ✓ Store the user name and marks in the firebase database.

- Integration Testing

Integration testing should be done between two or more components to ensure the components work as expected after integration. In this project integration testing is done to make sure the following activities are working as expected.

- ✓ Redirect to the relevant page when click the button in the dashboard
    - Example: Redirect to the leader board page when player clicks on "Leader board" button in the dashboard
- ✓ Redirect to the quiz page when click the "Quiz" button in both audio based quiz and article based quiz

- System Testing

The system testing should be done for the completed product before introduced to the market. The SmartCop – Safe Life Quiz game is one of the sub part of the SmartCop application. So, this testing will be done after connecting all sub parts of the SmartCop application.

- User Acceptance Testing

The user acceptance testing should be done by the actual end user to ensure that the product will satisfy all requirements which are requested by the user.

Some test cases performed for SmartCop – Safe Life Quiz game are listed below. The test cases are divided into two sections.

- Frontend test cases
- Backend test cases

2.4.1. Frontend test cases

Table 4: Test case for login using listed Google account

Test case ID	TC - 01
Test case scenario	Test for user login with valid Google account
Test steps	Select the Google account from the list which is prompt by an application
Test data	--
Expected results	Login to the application successfully
Actual results	Login to the application successfully
Pass / fail status	Pass

Table 5: Test case for login using Google account which is not listed in the list

Test case ID	TC - 02
Test case scenario	Test for user login using Google account which is not listed in the list which is prompt by an application
Test steps	Enter valid email address Click on next button Enter valid password Click on submit button
Test data	Email = abielachinee@gmail.com Password = abie1996@
Expected results	Login to the application successfully
Actual results	Login to the application successfully
Pass / fail status	Pass

Table 6: Test case for check whether the video is being played in the video based quiz section

Test case ID	TC - 03
Test case scenario	To test whether the video is being played in the video based quiz section in the game
Test steps	Click on the “Play quiz” button in the dashboard Select the “Play video based quiz” button
Test data	--
Expected results	Application will play the accident related video clip and display the question related to video
Actual results	Application will play the accident related video clip and display the question related to video

Pass / fail status	Pass

Table 7: Test case for check whether the audio is being played in the audio based quiz section

Test case ID	TC -04
Test case scenario	To test whether the audio is being played in the audio based quiz section in the game
Test steps	Click on the “Play quiz” button in the dashboard Select the “Play audio based quiz” button Click on the Play button
Test data	--
Expected results	Application will play the accident related audio clip
Actual results	Application will play the accident related audio clip
Pass / fail status	Pass

Table 8: Test case for check whether the article is being displayed in the article based quiz section

Test case ID	TC - 05
Test case scenario	To test whether the article is being displayed in the article based quiz section in the game
Test steps	Click on the “Play quiz” button in the dashboard Select the “Play article based quiz” button
Test data	--
Expected results	Application will display the accident related article

Actual results	Application will display the accident related article
Pass / fail status	Pass

Table 9: Test case for check whether the question is being displayed in the video based quiz section

Test case ID	TC - 06
Test case scenario	To test whether the user is getting question after video played completely in the video based quiz section
Test steps	Click on the “Play quiz” button in the dashboard Select the “Play video based quiz” button
Test data	--
Expected results	The question will be displayed after the video played completely
Actual results	The question will be displayed after the video played completely
Pass / fail status	Pass

Table 10: Test case for check whether the question is being displayed in the audio based quiz section

Test case ID	TC -07
Test case scenario	Test whether the user gets question when clicks on “Quiz” button in the audio based quiz
Test steps	Click on the “Play quiz” button in the dashboard Select the “Play audio based quiz” button Click on “Quiz” button

Test data	--
Expected results	The question will be displayed
Actual results	The question will be displayed
Pass / fail status	Pass

Table 11: Test case for check whether the question is being displayed in the article based quiz section

Test case ID	TC -08
Test case scenario	Test whether the user gets question when clicks on “Quiz” button in the article based quiz
Test steps	Click on the “Play quiz” button in the dashboard Select the “Play article based quiz” button Click on “Quiz” button
Test data	--
Expected results	The question will be displayed
Actual results	The question will be displayed
Pass / fail status	Pass

Table 12: Test case for question with correct answer

Test case ID	TC -09
Test case scenario	Test whether the user gets response when select correct answer for a question
Test steps	Select answer from given options
Test data	Select correct answer

Expected results	The answer will be changed from blue colour to green colour
Actual results	The answer colour will be changed from blue to green
Pass / fail status	Pass

Table 13: Test case for question with incorrect answer

Test case ID	TC -10
Test case scenario	Test whether the user gets response when select incorrect answer for a question
Test steps	Select answer from given options
Test data	Select incorrect answer
Expected results	The answer colour will be changed from blue to red
Actual results	The answer colour will be changed from blue to red
Pass / fail status	Pass

Table 14: Test case for leader board

Test case ID	TC -11
Test case scenario	Test whether the leader board is being displayed for player
Test steps	Click “Leader board” button in the dashboard
Test data	--
Expected results	The leader board will be displayed
Actual results	The leader board will be displayed

Pass / fail status	Pass

Table 15: Test case for user profile

Test case ID	TC -12
Test case scenario	Test whether the user profile is being displayed for player
Test steps	Click "User profile" button in the dashboard
Test data	--
Expected results	The user profile will be displayed
Actual results	The user profile will be displayed
Pass / fail status	Pass

Table 16: Test case for about section

Test case ID	TC -13
Test case scenario	Test whether the about section is being displayed for player
Test steps	Click "About" button in the dashboard
Test data	--
Expected results	The about section will be displayed
Actual results	The about section will be displayed

Pass / fail status	Pass

2.4.2. Backend test cases

Table 17: Test case for store user name and marks in firebase database

Test case ID	TC -14
Test case scenario	Test whether the user name and marks he/she gained is being stored in the firebase
Test steps	Answer all given questions in the game. Calculate total marks and store them in the firebase database.
Test data	Select answer for each question
Expected results	The final marks will be stored in the firebase database and displayed in the leader board.
Actual results	The final marks will be stored in firebase database and displayed in the leader board.
Pass / fail status	Pass

3. RESULTS AND DISCUSSION

3.1. Results

3.1.1. Application result

To evaluate the performance of the SmartCop – Safe Life Quiz game, the game was given to a set of randomly selected people including 5 students, 5 teenagers who have suffered from road accidents recently, and 5 non-medical professionals. They were asked to attempt the game three times and recorded their performances for all attempts and compared their improvements from attempt number 1 to attempt number 3.

Table18: Results for the students

Attempt	Marks				
	S1	S2	S3	S4	S5
1	60	50	63	71	30
2	68	65	77	83	50
3	74	80	82	92	60

Table 19: Results of the teenagers who have suffered from road accidents recently

Attempt	Marks				
	T1	T2	T3	T4	T5
1	65	70	50	49	38
2	70	83	61	55	47
3	78	92	70	63	63

Table 20: Results of the non-medical professional

Attempt	Marks				
	N1	N2	N3	N4	N5
1	35	23	60	59	64
2	48	35	75	68	70
3	55	50	83	82	88

The test results shown in Table 18 to Table 20, display an improvement in the scores of the players, when they re-try the activities. Also, there is an improvement of the scores through attempt 1 to attempt 3. So it can be said that the game will improve post road accident responsibility knowledge among the public.

3.1.2. Machine learning model result

The SmartCop – Safe Life Quiz game uses a RFC algorithm to predict the next question and time needed to answer a question based on the player's performance. Initially the RFC algorithm predicts the question with 66% accuracy. Then the accuracy has increased to 87.5% by increasing the number of features and number of data. And also predicts the time with 100% accuracy using the same features which were used to predict the question. Figure 29 and Figure 30 shows the accuracy score obtained for predicting questions and predicting time respectively.

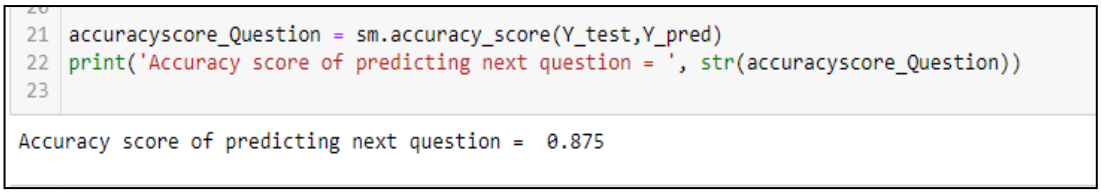

Figure 29: Accuracy score obtained for predicting question

```
23
24 accuracyscore_Time = sm.accuracy_score(Y_test1,Y_pred1)
25 print('Accuracy score of predicting time need to given for next question to answer= ', str(accuracyscore_Time))
```

```
Accuracy score of predicting time need to given for next question to answer=  1.0
```

Figure 30: Accuracy score obtained for predicting time

3.2. Research Findings

The basic idea of the project is to implement a game based application which can improve post road accident responsibility knowledge among people. A deep research was done to identify the best algorithm to predict the question. The SVM and RFC were selected to predict the question. In order to finalize the algorithm, the dataset was trained using both SVM and RFC. As a result SVM produced less accuracy than RFC. So, finally RFC was selected to predict the question. Initially current question type, time given and answer status were used as features to predict the question and it gives 66% accuracy. To improve the accuracy the time given by the system also used as a feature to predict the question and it gives 87% accuracy.

For this project an android studio was used as an IDE to develop this game application. However it takes more processing power and time to run the application. In order to reduce the processing power and time it is better to use IntelliJ IDEA because it also supports flutter SDK.

3.3. Discussion

Currently the SmartCop – Safe Life Quiz game uses a supervised learning algorithm to predict questions based on the player's performance. In future, the questions in the game will be decided to predict using reinforcement learning algorithms to gain more accuracy. Since people with poor knowledge in English are facing language issues because the game is only available in English. So, in order to improve the performance of the people the game will be launched with multiple languages feature in future.

The application will be created for ios users easily because the flutter supports ios platform. So, the application will be created for ios platform without major modifications in code base and the app will be available through Apple store in future.

4. CONCLUSION

The implemented application helps people to improve and test their post road accident knowledge level more than theoretical approaches. In theoretical approaches people just memorize the activities they need to do when they meet an accident but in this application people do not need to memorize the activities. Because in the video based quiz section they experience the accident by watching the video. So it will help them to make decisions quickly on the spot. Likewise using an audio based and an article based quiz they can assist others to perform tasks on the spot. In future the game application will be available for all android and ios users with multiple language and other trending game features.

REFERENCES

[1] W. H. Organization. "Global status report on road safety 2018." https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/ (accessed 01-Sep-2020.

[2] A.Binsubaih, S.Maddock, and D.Romano, "A serious game for traffic accident investigators."

[3] S. Benkhedda and F. Bendella, "FASim: A 3D Serious Game for the First Aid Emergency," vol. 50, no. 6, pp. 690-710, 16-Aug-2019. [Online]. Available: https://journals.sagepub.com/doi/10.1177/1046878119865913.

[4] F. Balahadia, "Development of game-based learning application for first aid awareness," Mar-21 2019. [Online]. Available: https://www.researchgate.net/publication/331928166_Development_of_game-based_learning_application_for_first_aid_awareness.

[5] H.Erdogan, Y.A.Acar, O.Cinar, E.Surer, and P.Karagoz, "Game-based mobile platform to instruct first aid techniques."

[6] I. Nicolaidou and A. Venizelou, "Improving Children's E-Safety Skills through an Interactive Learning Environment: A Quasi-Experimental Study," 9 April 2020.

[7] E. Vingilis, J. Seeley, D. L. Wiesenthal, C. M. Wickens, P. Fischer, and R. E. Mann, "Street racing video games and risk-taking driving: An Internet survey of automobile enthusiasts," *Accident Analysis & Prevention*, vol. 50, pp. 1-7, 2013/01/01/ 2013, doi: https://doi.org/10.1016/j.aap.2012.09.022.

[8] a. T. Limited. "First Aid Quiz Game." https://play.google.com/store/apps/details?id=com.avistechbd.quiz (accessed 12-Jan-2020.

[9] E. Apps. "First Aid MCQ." https://play.google.com/store/apps/details?id=com.jdapp.firstaidmcq (accessed 12-Jan-2020.

[10] eoinzy. "First Aid Ireland Pop Quiz." https://play.google.com/store/apps/details?id=com.eoinzy.firstaid (accessed 12-Jan-2020.

[11] N. S. Games. "First Aid Self Test." https://play.google.com/store/apps/details?id=uk.co.nightsnow.first_aid_self_test (accessed 12-Jan-2020.

APPENDICES

Appendix A

Machine learning model for predicting questions

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
import sklearn.metrics as sm
import joblib
import json
import requests
import numpy as np
#Read the dataset
dataset = pd.read_csv('E:\\GameDataset.csv')
print(dataset)
dataset.describe()
#Split the data set as feature columns and prediction column
features =  dataset.iloc[:,0:4].values
predictQuestion = dataset.iloc[:,4].values
#Split the data set into train dataset and test dataset
X_train,X_test,Y_train,Y_test = train_test_split(features,predictQuestion,test_size =
0.25, random_state = 20)
#Train the model using RFC
classifier_Question = RandomForestClassifier(n_estimators =35, criterion
='entropy',random_state = 35)
res = classifier_Question.fit(X_train,Y_train)
#Predict the questions
Y_pred = classifier_Question.predict(X_test)
#Calculate accuracy score
accuracyscore_Question = sm.accuracy_score(Y_test,Y_pred)
print('Accuracy score of predicting next question = ', str(accuracyscore_Question))
joblib.dump(res, 'questionmodel.pkl')
cols=["CurrentQuestion","TimeGiven","TimeTakenToAnswer","AnswerStatus"]
joblib.dump(cols, "questionmodel_cols.pkl")
```

Appendix B

Machine learning model for predicting time

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

import sklearn.metrics as sm

import joblib

import json

import requests

import numpy as np

#Read the dataset

dataset = pd.read_csv('E:\\GameDataset.csv')

print(dataset)

dataset.describe()

#Split the data set as feature columns and prediction column

features =  dataset.iloc[:,0:4].values

predictTime = dataset.iloc[:,5].values

#Split the data set into train dataset and test dataset

X_train1,X_test1,Y_train1,Y_test1 = train_test_split(features,predictTime,test_size
= 0.25, random_state = 20)

#Train the model using RFC

classifier_Time = RandomForestClassifier(n_estimators =35, criterion
='entropy',random_state = 35)

res1 =classifier_Time.fit(X_train1,Y_train1)

#Predict the time

Y_pred1 = classifier_Time.predict(X_test1)

#Calculate accuracy score

accuracyscore_Question = sm.accuracy_score(Y_test,Y_pred)

accuracyscore_Time = sm.accuracy_score(Y_test1,Y_pred1)

print('Accuracy score of predicting time need to given for next question to answer= ',
str(accuracyscore_Time))

joblib.dump(res1, 'timemodel.pkl')

cols=["CurrentQuestion","TimeGiven","TimeTakenToAnswer","AnswerStatus"]

joblib.dump(cols, "timemodel_cols.pkl")
```

Appendix C

Flask server code for predicting questions

```
from flask import Flask, jsonify, request
import pandas as pd
import numpy as np
import joblib
import traceback
from flask_restful import reqparse
app = Flask(__name__)

@app.route("/", methods=['GET'])
def hello():
    return "hey"

@app.route('/predict', methods=['POST'])
def predict():
    lr = joblib.load("questionmodel.pkl")
    if lr:
        try:
            json = request.get_json()
            model_columns = joblib.load("questionmodel_cols.pkl")
            temp=list(json[0].values())
            vals=np.array(temp)
            prediction = lr.predict(temp)
            print("here:",prediction)
            return jsonify({'prediction': str(prediction[0])})

        except:
            return jsonify({'trace': traceback.format_exc()})
    else:
        return ('No model here to use')
```

Appendix D

Flask server code for predicting questions

```
from flask import Flask, jsonify, request
import pandas as pd
import numpy as np
import joblib
import traceback
from flask_restful import reqparse
app = Flask(__name__)

@app.route("/time", methods=['GET'])
def hello():
    return "hey"

@app.route('/predicttime', methods=['POST'])
def predict():
    lr = joblib.load("'timemodel.pkl")
    if lr:
        try:
            json = request.get_json()
            model_columns = joblib.load("'timemodel _cols.pkl")
            temp=list(json[0].values())
            vals=np.array(temp)
            prediction = lr.predict(temp)
            print("here:",prediction)
            return jsonify({'prediction': str(prediction[0])})

        except:
            return jsonify({'trace': traceback.format_exc()})
    else:
        return ('No model here to use')
```