DESIGN AND DEVELOPMENT OF IOT BASED SCORECARD FOR GOLF SPORT

MICHAEL CHIN KAH CHOON

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DESIGN AND DEVELOPMENT OF IOT BASED SCORECARD FOR GOLF SPORT

MICHAEL CHIN KAH CHOON

This report is submitted in accordance with the requirement of the Universiti Teknikal Malaysia Melaka (UTeM) for the Bachelor of Electronics Engineering Technology (Telecommunications) with Honours.

FACULTY OF ELECTRICAL AND ELECTRONIC ENGINEERING

TECHNOLOGY

2021

DECLARATION

I hereby, declared this report entitled DESIGN AND DEVELOPMENT OF IOT BASED SCORECARD FOR GOLF SPORT is the results of my own research except as cited in references.

Signature: ....

Author: MICHAEL CHIN KAH CHOON

Date: 15/01/2021

APPROVAL

This report is submitted to the Faculty of Electrical and Electronic Engineering Technology of Universiti Teknikal Malaysia Melaka (UTeM) as a partial fulfilment of the requirements for the degree of Bachelor of Electronics Engineering Technology (Telecommunications) with Honours. The member of the supervisory is as follow:

Signature: ………………………………………………………

Supervisor: IR. DR. MOHD FARRIZ BIN HJ. MD
BASAR

ABSTRAK

Kad skor berasaskan IoT untuk sukan golf ialah sistem rakaman skor yang digunakan untuk mengumpulkan, merekod, dan memaparkan keputusan masa nyata semasa kejohanan golf. Perkara ini dapat dilakukan dengan menggunakan aplikasi berbasis Android sebagai peranti input bagi sistem tersebut untuk memasukkan skor yang diperoleh, dan menampilkan kedudukan pemain secara masa nyata. Objektif projek ini adalah untuk mereka bentuk dan membangunkan kad skor berasaskan IoT untuk sukan golf. Selain itu, projek ini akan menguji prestasi kad skor berasaskan IoT terhadap sistem rakaman skor tradisional. Projek ini menggunakan Google Firebase sebagai pangkalan data dalam talian untuk menyimpan dan memproses data yang dimasukkan oleh pengguna. Kemudian, keputusan pertandingan golf akan dipaparkan di monitor komputer atau TV pintar dengan akses ke Google Firebase. Jangkaan hasil daripada projek ini diketengahkan kad skor golf berasaskan IoT yang memberikan pengalaman berguna kepada pemain golf untuk merekodkan skor dan analisis pemenang dalam sebuah kejohanan. Selain itu, projek ini juga akan mengurangkan masa yang diperlukan untuk menamatkan sebuah kejohanan golf.

ABSTRACT

IoT based scorecard for golf sport is a score recording system used to collect, record, and display the real-time results during a golf tournament. This can be done by using an Android based application as the input device of the system to key in the score obtained and display the players' real-time ranking. The objective of this project is to design and develop the IoT based scorecard for golf sport. Other than that, this project will test the performance of the IoT based scorecard against the traditional score recording system. This project uses the Google Firebase as the online database to store and process the data enter by users. Then, the real-time results of the golf tournament will be display on a computer monitor or a smart TV with access to Google Firebase. The results of this project featured an IoT based golf scorecard provide handy experience to golfers regards to score recording and winner analysis in a tournament. Other than that, this project will also reduce the time taken to complete a golf tournament.

DEDICATION

This report is dedicated to my beloved parents who educated and supported me throughout the process of doing this project. I am also wanted to say thank you to my supervisor and my friends who have encouraged, guided and inspired me to complete this project.

ACKNOWLEDGEMENTS

This report is as a mark of my sincere appreciation to Universiti Teknikal Malaysia Melaka (UTeM) for giving me this chance to further study on bachelor's degree in Electronics Engineering Technology (Telecommunications) in Faculty of Electrical and Electronics Engineering Technology (FTKEE). I would like to thanks to my supervisor, Ir. Dr. Mohd Farriz Bin Hj. Md Basar for the guidance, advices, encouragement, inspiration and attention given throughout the day in development of my final year project and while writing this report entitled as Design and Development of IoT based Scorecard for Golf Sport. With this continuous support and interest, he was guiding me to complete this project with full commitment and dedication. My gratitude goes to my beloved family and my friends that always give courage and support me to achieve the goal of my project. Thanks to their moral support and care they had given to me up until this project done.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

IoT	Internet of Things
API	Application Programming Interface
SDK	Software Development Kit
CPU	Central Processing Unit
IFFA	Indonesian Flag Football Association

CHAPTER 1

INTRODUCTION

1.0 Introduction

This chapter purpose is to create the outline of this project. All the requirements need to be achieved by the end of this project will be included. The project background will be provided in brief and the architecture of the project will be precisely illustrated.

1.1 Background

According to Hardin, J. D. (1991) Golf scorecard is a sheet material used to register the identification of four players and record individual player stroke total in score boxes aligned next to the identification blocks which are now widely used in golf clubs during a golf tournament to be register and record the stroke score by the golfers. Figure 1.1 shows a golfer record score obtained by using golf scorecard.

Figure 1.1: Golfers Record Score Obtained by using Golf Scorecard
[Source: https://www.golf-monthly.co.uk/features/golf-scorecard-rules-simple-but-important-67737]

According to Lambourne, G. T. (1996) electronic golf scorecard is number pad and display packed in a casing to serve as a golf scorecard which will record initials of the players, handicap details and the stroke score of each player in the individual holes during a golf game. Development of an IoT based golf scorecard is to improve the efficiency of the score recording and analyzing process during a golf tournament. Furthermore, this will also reduce the burden on the analyzers of the golf tournament.

1.2 Problem Statement

At most of the golf clubs in Malaysia, golf tournament takes a very long time to be finished and analyze the winner of that golf tournament. This is because golfers in that tournament have to write down their stroke score manually in the given traditional golf scorecard. After the golfers finished their games and recorded their score, all the scorecard will be handed over to the tournament analyzers to be calculate and analyze the winner with the least stroke score in the tournament. The time taken to be analyzed the winner may longer than the time taken by the golfer to finish their games.

Other than that, the analyzation of the winner might have error in terms of the stroke score written by the golfers. There might be reading and writing error occur when recording and analyzing the stroke score. This will cause an unfair result for all the golfers participated.

1.3 Objectives

The objectives for this project are:

1. To design an IoT based golf scorecard.
2. To develop an IoT based golf scorecard by using MIT App Inventor II and Google Firebase Online Database.
3. To analyse the performance of the IoT based golf scorecard with online database system for tournament that having in golf clubs in terms of accuracy and efficiency.

1.4 Scope of Work

This project purposely focuses on the design and development of the IoT based golf scorecard. The main product will be based on an android based application, an online database. The programming language used to develop the android-based application will be block language. The online database will be setup based on Google Firebase online database due to its simplicity and no cost will be needed. The data stored in the online database will then display through computer monitor or Smart TV.

1.5 Project Significant

Previously, golf score recorded manually by using physical scorecard and pencil. The winner analysis will start after the game finished by golfers and all the scorecards been submitted by participants. This project unique from work that previously done is it uses a real-time online database to store the stroke score recorded by golfers by the android-based application, and the data stored will then display through the computer or Smart TV on the spot. Thus, it will be improving the accuracy and efficiency of the score recording and analysing process.

CHAPTER 2

LITERATURE REVIEW

2.0 Introduction

This chapter consists of the previous project explanations and the findings from previous related project to be improve in this current project.

2.1 Previous Project Related

This section will discuss the related previous project. A comparison between this project and previous project will be present.

2.1.1 Golf Scorecard

According to Laakso, J. K. (1994) this golf scorecard had been invented back in 1991. It is used to record totals for individual and team stroke as well as total points for individual and team play. This invention is to provide a golf scorecard with a basic layout that can be used to record amounts of strokes and total game points for individual golfers and golf teams concisely and effectively. Other than that, it was also meant to provide a unique scoring system for the different known golf games which effectively helps speed play by simplifying the scoring process. The plan view of the scorecard is shown in Figure 2.1 below.

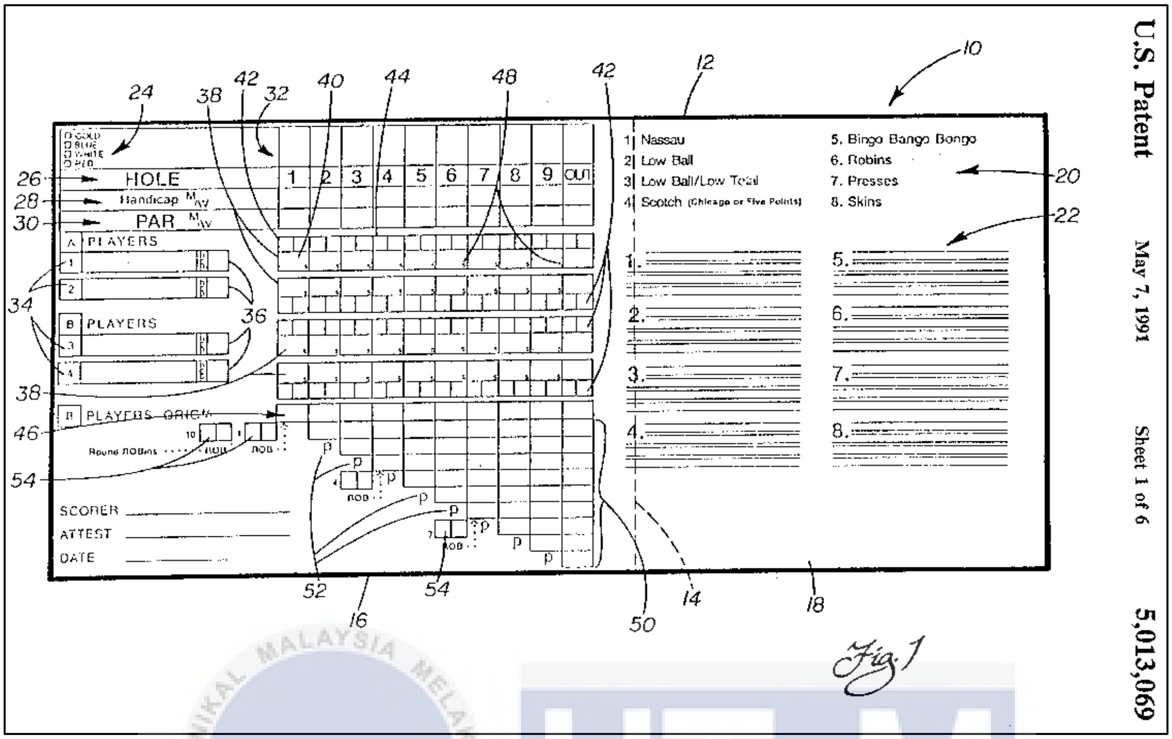
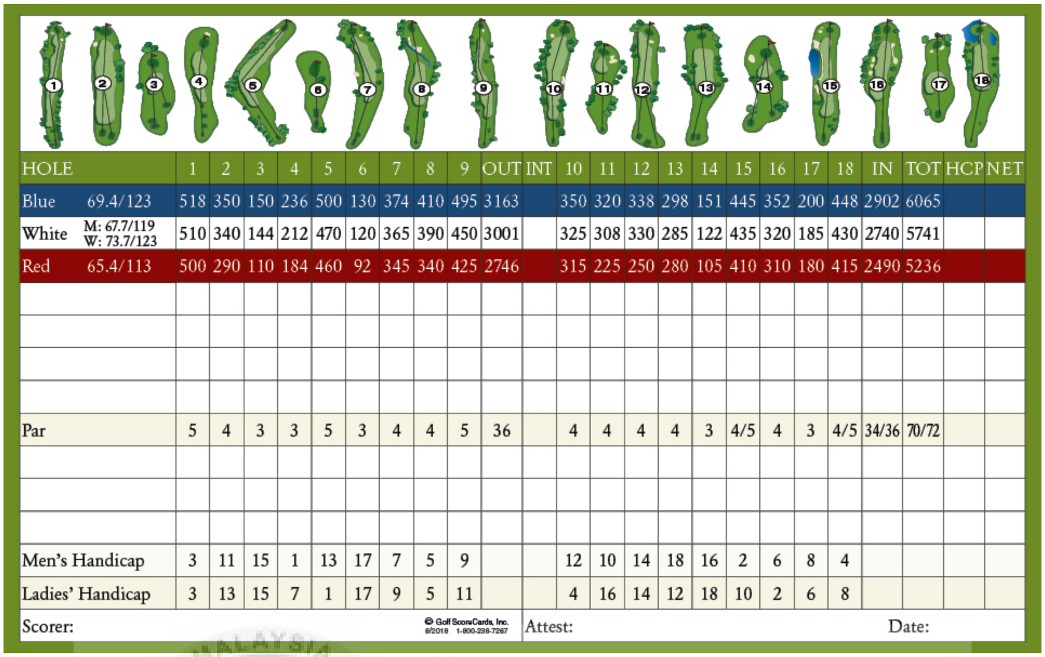


Figure 2.1: Plan view of one side of the scorecard
[Source: Laakso, J. K. (1994). Golf Scorecard]

The advantages of this golf scorecard effectively increase the interest of players in the different golf games established over the years, thereby increasing the camaraderie among players and their interest in golf. Moreover, by providing an easy-to-learn and easy-to - follow scoring system, the scorecard encourages quicker play from which everybody on the golf course profits by considerably reducing the time required to play a golf round. On the other hand, the time taken to analyse the winner of a tournament will be long due to the manual score recording by using the scorecard. Then, the probability of a wrongly record score is much higher due to the score is written by handwriting. A physical golf scorecard from Cresta Verde Golf Course, United States is shown in figure 2.2 below



HOLE		1	2	3	4	5	6	7	8	9	OUT	INT	10	11	12	13	14	15	16	17	18	IN	TOT	HCP	NET
Blue	69.4/123	518	350	150	236	500	130	374	410	495	3163		350	320	338	298	151	445	352	200	448	2902	6065		
White	M: 67.7/119 W: 73.7/123	510	340	144	212	470	120	365	390	450	3001		325	308	330	285	122	435	320	185	430	2740	5741		
Red	65.4/113	500	290	110	184	460	92	345	340	425	2746		315	225	250	280	105	410	310	180	415	2490	5236		
Par		5	4	3	3	5	3	4	4	5	36		4	4	4	4	3	4/5	4	3	4/5	34/36	70/72		
Men’s Handicap		3	11	15	1	13	17	7	5	9			12	10	14	18	16	2	6	8	4				
Ladies’ Handicap		3	13	15	7	1	17	9	5	11			4	16	14	12	18	10	2	6	8				
Scorer:									© Golf ScoreCards, Inc. 8/2018 1-800-238-7267			Attest:										Date:			

Figure 2.2: Golf Scorecard from Cresta Verde Golf Course
[Source: https://www.golfcrestaverde.com/aboutus/scorecard/]

2.1.2 Electronic Golf Scorecard

According to Lambourne, G. T. (1996) the electronic golf scorecard was introduced back in 1996. The electronic golf scorecard packed with a keypad and a screen, where the keypad is the input device for the scorecard to record player’s data and score on the holes during a game of golf. The data entered will be stored and processed in the scorecard, relevant information will be displayed so that player’s score can be calculated visually. The scorecard able to set with a sets of program mode which par values and golf course data can be entered. The housing includes a microprocessor which will store and process the command entered by the keypad.

The advantages bring by this electronic golf scorecard is it is much more user friendly in compare with the traditional paper golf scorecard. Other than that, it also helps to reduce waste as it will eliminates the use of paper golf scorecard and pencil. Furthermore, the compact size of the electronic golf scorecard is pocket friendly to most of the golfers and it will be more durable than the paper golf scorecard as the material use to build the electronic golf scorecard is weatherproof. The electronic golf scorecard also readily implements to most golf course around the entire world back then. The gross and net scores are calculatable for players of the games of golf. Although the electronic golf scorecard provides some convenient to golfers, but there also some disadvantages brings by this device. The time taken to analyse the winner of the tournament is not overcome by using this electronic scorecard due to this scorecard only provide a simple and easy-to-use scorecard for golfers. This scorecard do not provide any improvements to the winner analysis of a tournament. The design of the electronic golf scorecard is shown in the Figure 2.3 below.

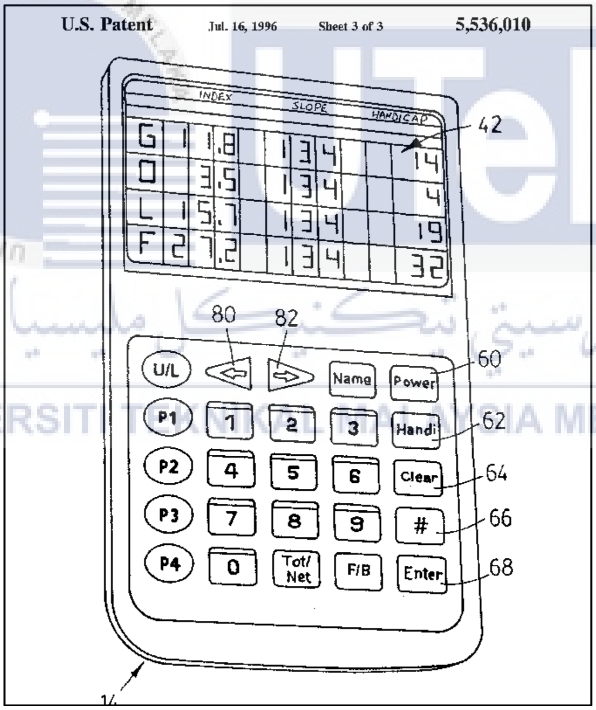


Figure 2.3: Electronic Golf Scorecard design back in 1996
[Source: Lambourne, G. T. (1996). Electronic Golf Scorecard]

2.1.3 Golf Score Recording System and Network

According to Remedio, J. W., & Appleton, M. R. (1990) the golf score recording system and network is a program for creating, processing and gathering golf data on a variety of golf players playing golf on a variety of courses in the country. This system includes a central computer module and a mobile computer module. The system works with the mobile computer module where golfers can enter their data into the mobile computer module, then the data will be sent to the central computer module as a record basis. The data entered include golfer's identification, stroke score or "par" values and golfer handicap.

This system can be applied to one golf course with a sets of central computer module and mobile computer module. If golf club has more than one golf course, then the golf club need to prepare other sets of the central computer modules and mobile computer modules as one set of system can only be implemented to on golf course. The mobile computer module includes a keypad and a display packed in a housing. The data can be input by using the keypad and the keyed in data will be displayed on the screen. Other than that, this score recoding system is a complicated system as it has too many components include in the system. Figure 2.4 shown the blocks of central computer module and Figure 2.5 shown the blocks of mobile computer module. As illustrated in the figure, the mobile computer module has lots of components include power supply, voltage regulation, program and data storage, data input device and display device all connected to a central processing unit (CPU). Figure 2.6 below shown the design of the mobile computer unit.

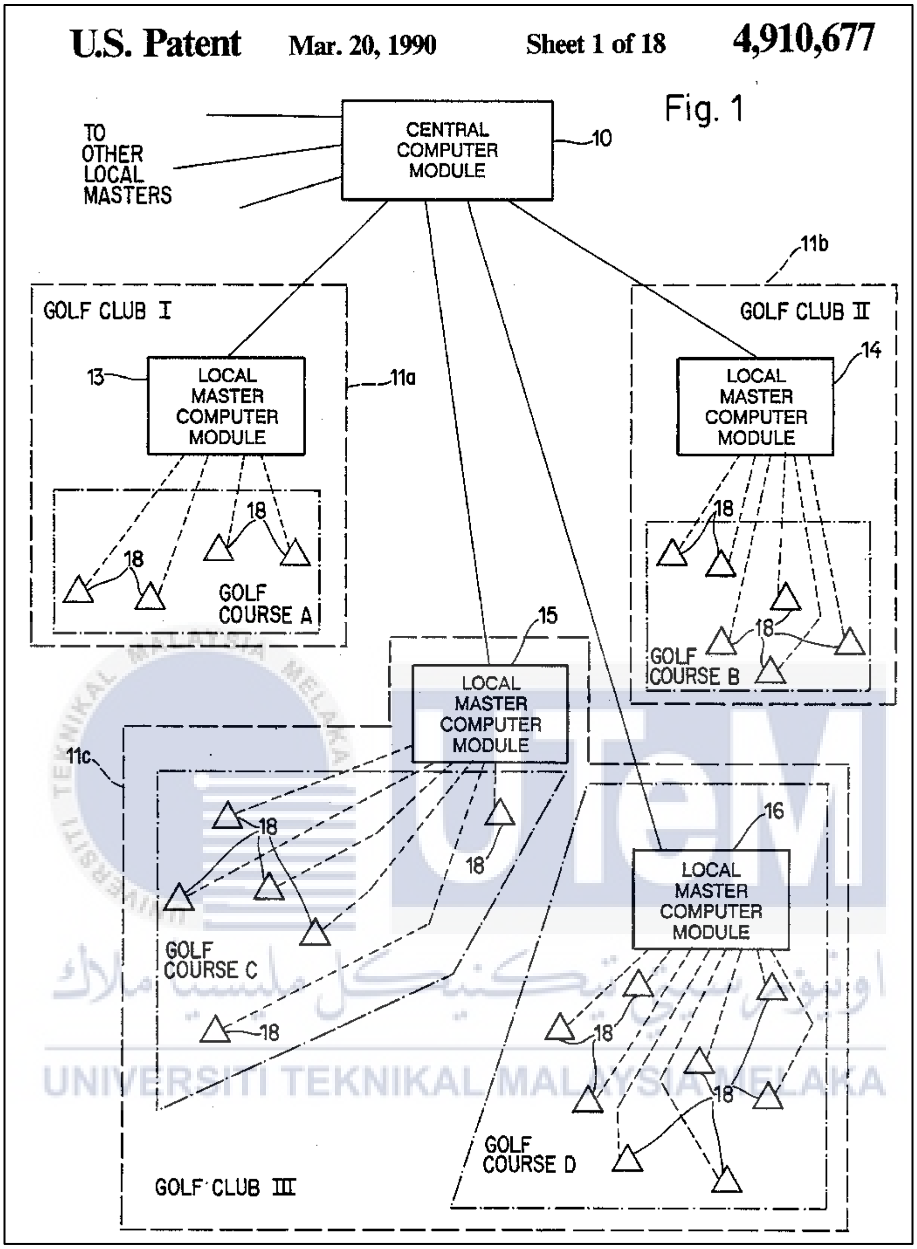


Figure 2.4: Blocks of Central Computer Module
[Source: Remedio, J. W., & Appleton, M. R. (1990). Golf Score Recording System and Network]

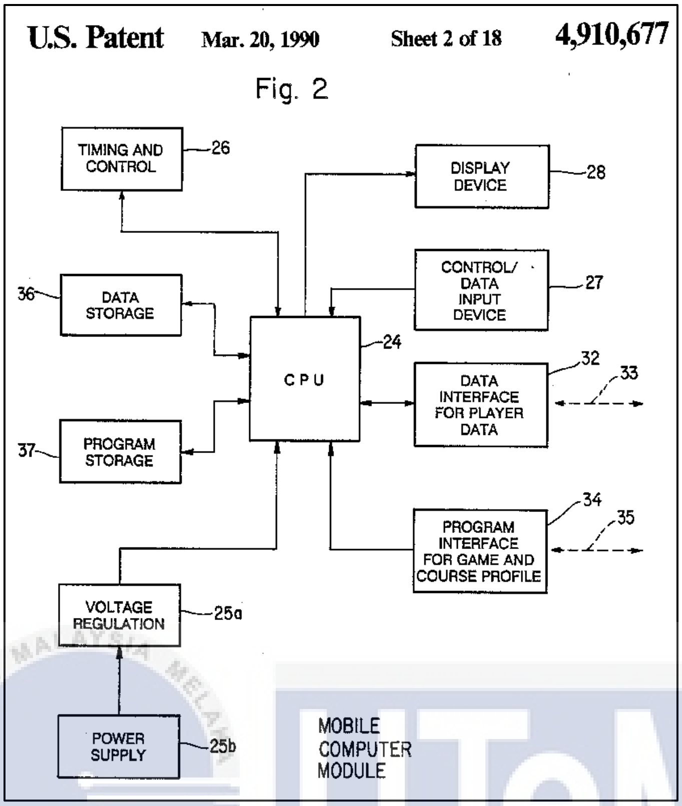


Figure 2.5: Blocks of Mobile Computer Module
[Source: Remedio, J. W., & Appleton, M. R. (1990). Golf Score Recording System and Network]

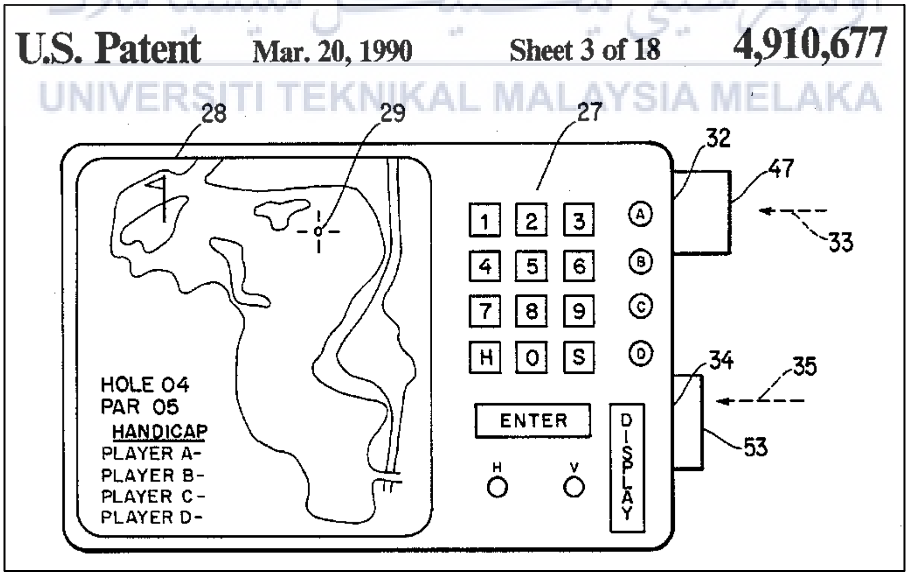


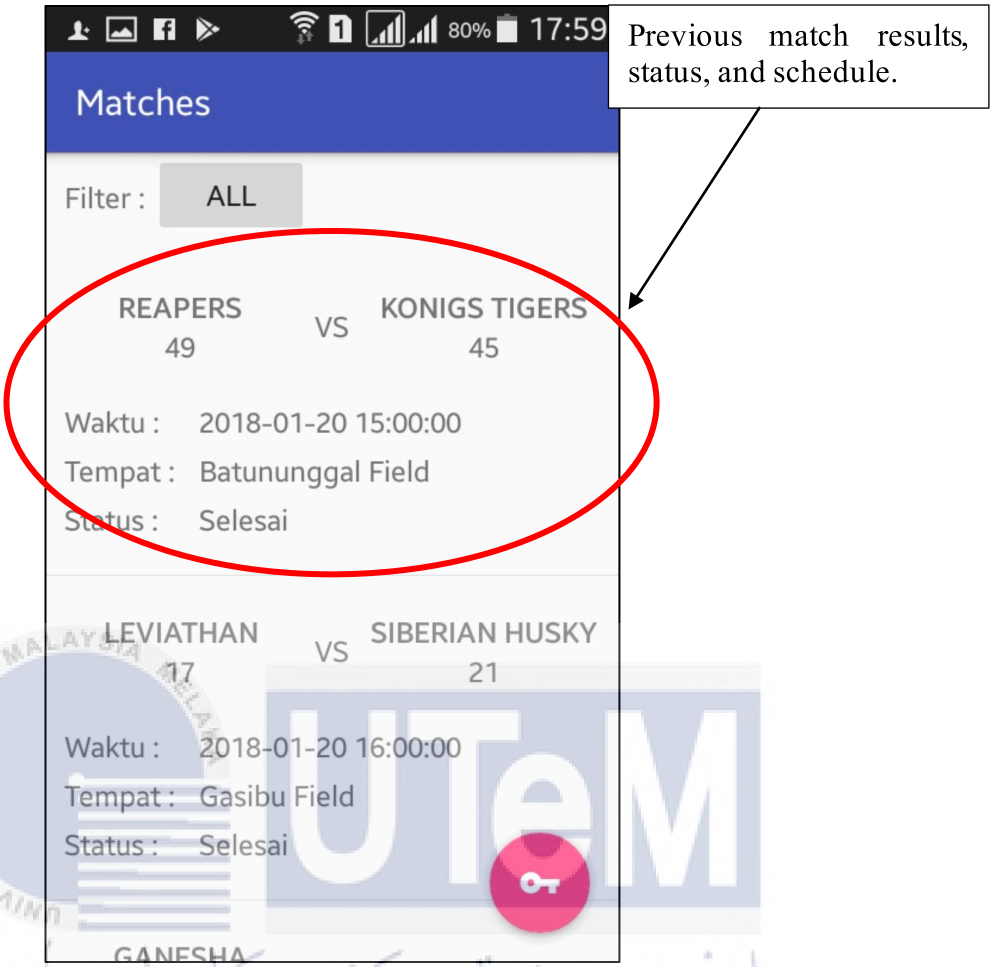
Figure 2.6: Mobile Computer Module
[Source: Remedio, J. W., & Appleton, M. R. (1990). Golf Score Recording System and Network]

2.1.4 Android based Live Score Application for Flag Football

According to Ramadhan, A. F., & Suryana, T (2018). the Live Score application for Flag Football is to show the results of a match of flag football in real-time to provide user an updated information. Figure 2.7 below illustrated a sample picture of flag football league.

Figure 2.7: American Flag Football League

The application is based on Android operating system created by using Android SDK software involving Java and PHP programming language. The live score system consists of Application Programming Interface (API) to use as the database of the system as it will store and process the data and convert the values into text format and display in the Android based Live Score application. With the presence of this application, users get to access the latest information of each flag football match in real-time. Match schedule interface is shown in Figure 2.8 below.



80% 17:59

Matches

Filter : ALL

REAPERS 49 VS KONIGS TIGERS 45

Waktu : 2018-01-20 15:00:00
Tempat : Batununggal Field
Status : Selesai

LEVIATHAN 17 VS SIBERIAN HUSKY 21

Waktu : 2018-01-20 16:00:00
Tempat : Gasibu Field
Status : Selesai

GANESHA

Previous match results, status, and schedule.

Figure 2.8: Match Schedule Interface on Mobile
[Source: Ramadhan, A. F., & Suryana, T. Live Score Application Development for Flag Football Based on Android]

The match information will be updated by the admin of Indonesian Flag Football Association (IFFA). Then, the data will be store and process in the API server. After that, the data stored will be convert into text format and send out to the Android live score application. Users can be conveniently access to the match information, real-time score, and results. Figure 2.9 below illustrate the mobile data match system input interface. The significant of this Live Score application is to provide a more convenient way for user to check the live results of Flag Football.

On the other hand, golf sport does need a similar system with add on of analysis process in the system to shorten the analysis time taken and also the time taken of a tournament.

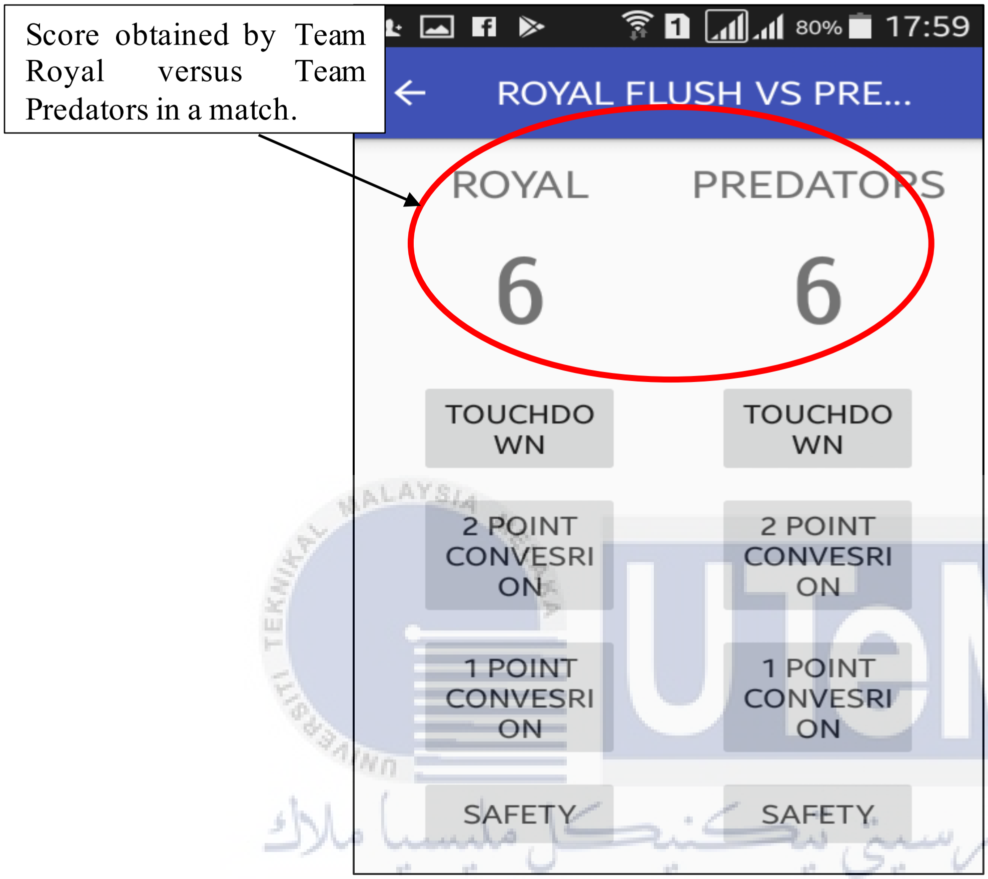
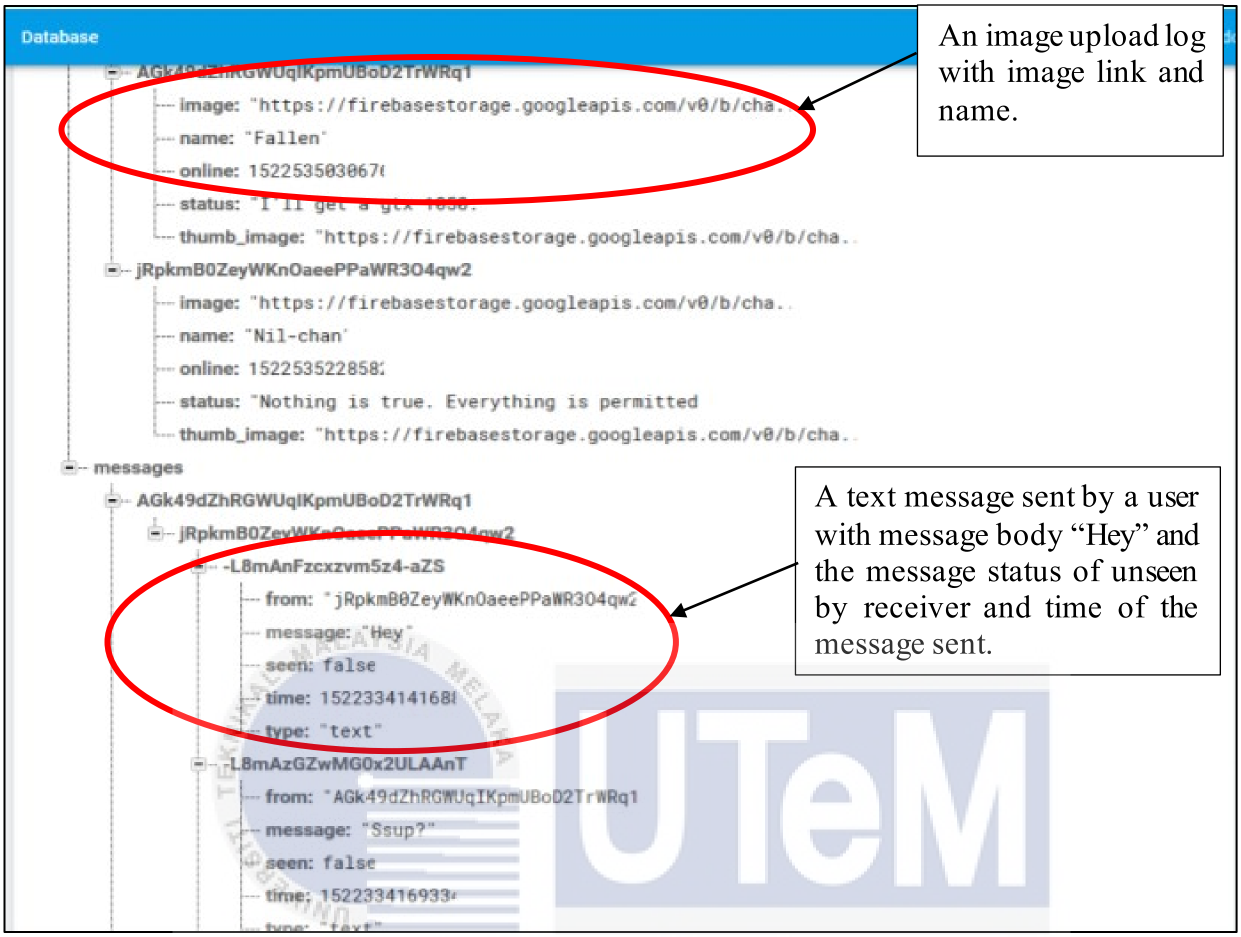

Score obtained by Team Royal versus Team Predators in a match.

Figure 2.9: Mobile Data Match System Input Interface
[Source: Ramadhan, A. F., & Suryana, T. Live Score Application Development for Flag Football Based on Android]

2.1.5 Real-time Communication Application Based on Android Using Google Firebase

According to Chatterjee, N., Chakraborty, S., Decosta, A., & Nath, A. (2018). Real-time Communication Application Based on Android Using Google Firebase is an application that allow operations such as sharing audio files, media files, text and voice messages by using Google firebase as backend of transmission operation. The Firebase framework was developed primarily as an in-house database with its APIs, allowing users to store and synchronize data and information between users. It also can be used as portable applications for businesses that required real-time database to synchronize all the updated record among the users immediately. The android application is to provide users to communicate based on Firebase database system. There are few modules included in this application. First, the start module is used to identify the availability of network connection between users. After that, the register module is to provide user identification by creating a new account for the user to login to their account and an authentication for the Firebase server. Besides, login module will provide user identification by logging in with their own user ID and password. Finally, the chat module is a platform to allow users sharing their files or messages by uploading it into the Firebase Storage. Then, the system will identify the file format uploaded by user.

The application and Firebase services provided a very convenient platform between users to share their information or files. The Firebase services are relevant as these tools have rendered the development of these applications in more effectively and much quicker way. Besides, the firebase online database is a simple and user-friendly online database system. The implementation of firebase online database to the IoT golf scorecard system will contribute a lot of simplicity to the score analysis process to analyse the winner of a tournament quicker. This is proven by referring to the communication application based on Google Firebase, because it provides a real-time platform to the users, meaning that all the data will be display once the user sent the data. The Firebase online database interface with user entries and message entries is shown in Figure 2.10 below.



```
AGk49dZhRGWUqIKpmUBoD2TrWRq1
    image: "https://firebasestorage.googleapis.com/v0/b/cha...
    name: "Fallen"
    online: 152253503067(
    status: "I'll get a gtx 1050.
    thumb_image: "https://firebasestorage.googleapis.com/v0/b/cha...
jRpkmB0ZeyWKnOaeePPaWR3O4qw2
    image: "https://firebasestorage.googleapis.com/v0/b/cha...
    name: "Nil-chan"
    online: 152253522858:
    status: "Nothing is true. Everything is permitted
    thumb_image: "https://firebasestorage.googleapis.com/v0/b/cha...
messages
    AGk49dZhRGWUqIKpmUBoD2TrWRq1
        jRpkmB0ZeyWKnOaeePPaWR3O4qw2
            -L8mAnFzcxzvm5z4-aZS
                from: "jRpkmB0ZeyWKnOaeePPaWR3O4qw2
                message: "Hey"
                seen: false
                time: 152233414168(
                type: "text"
            -L8mAzGZwMG0x2ULAAnT
                from: "AGk49dZhRGWUqIKpmUBoD2TrWRq1
                message: "Ssup?"
                seen: false
                time: 152233416933(
                type: "text"
```

An image upload log with image link and name.

A text message sent by a user with message body “Hey” and the message status of unseen by receiver and time of the message sent.

Figure 2.10: The Firebase Database with user entries and message entries [Source: Chatterjee, N., Chakraborty, S., Decosta, A., & Nath, A. (2018). Real-time Communication Application Based on Android Using Google Firebase]

2.2 Summary

This project featured an Android based scorecard application and Google Firebase as the online database. The Google Firebase is chosen to use as the online database of this project because it is free to use. Other than that, the simple user interface of Google Firebase online database provides a user-friendly experience to user and developer. According to Chatterjee, N., Chakraborty, S., Decosta, A., & Nath, A. (2018) Google Firebase online database can be used as portable applications real-time database to synchronize all the updated record among the users immediately.

CHAPTER 3

METHODOLOGY

3.0 Introduction

This section will discuss about the methodology being used to conduct this project. The explanation will start from the beginning of the project to the development of the project. The flow of project development will be shown.

3.1 Flowchart for the Project

Figure 3.1 shows the flowchart of the development process for this project. First, the project started by research for related journal and similar previous project. Most of the previous project that have similar methodology with this project will be revisited as a reference. Besides, the advantages and disadvantages of previous project will be point out and make improvement to it. Then, design the layout for the software. This layout basically includes the user interface of the Android based application, the file format to be export is .apk file. Other than that, the layout will also mainly be based on simple golf scorecard layout to make it user friendly for golfers.

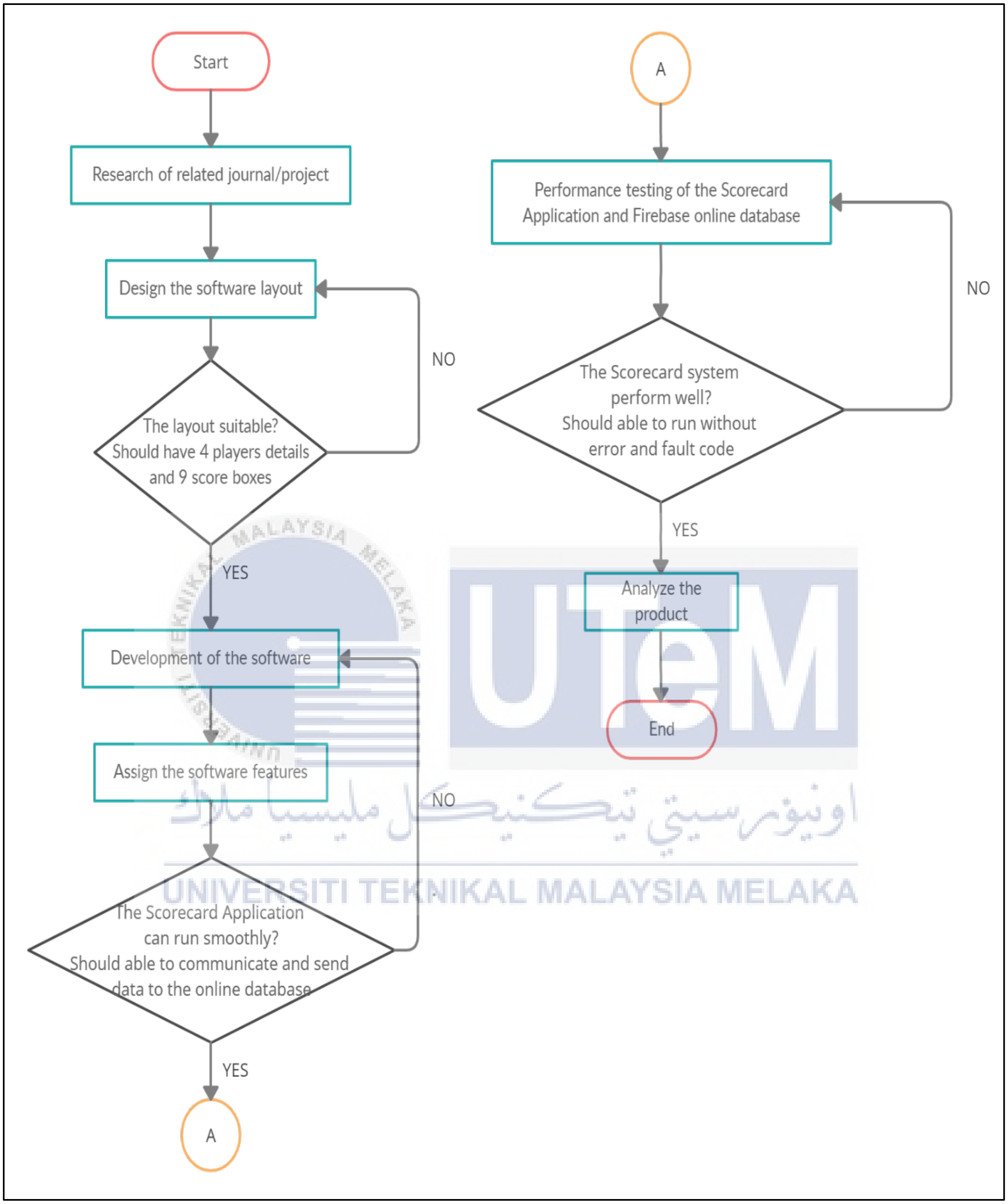
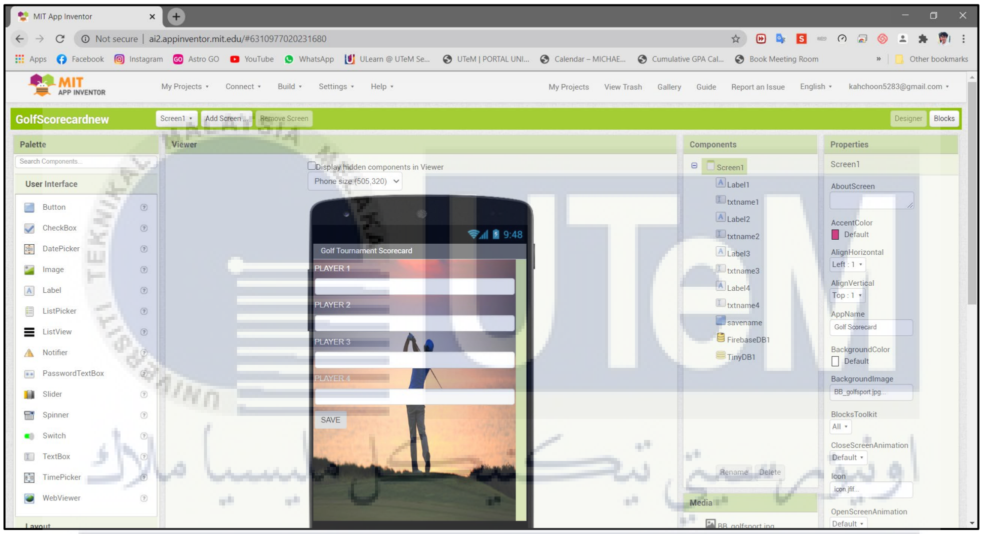

Start
Research of related journal/project
Design the software layout
The layout suitable?
Should have 4 players details
and 9 score boxes
NO
YES
Development of the software
Assign the software features
The Scorecard Application
can run smoothly?
Should able to communicate and send
data to the online database
NO
YES
A
A
Performance testing of the Scorecard
Application and Firebase online database
The Scorecard system
perform well?
Should able to run without
error and fault code
NO
YES
Analyze the
product
End

Figure 3.1: Flowchart of the Development of IoT based Golf Scorecard

Further, the layout should be designed with 9 score boxes for 9 holes golf sport and 4 players details with players' names included. In this case, the application will be developed by using MIT App Inventor II due to its simplicity. After that, the designed layout will be verified, if there is any layout error or the layout did not suitable for golf sport, then the layout will be redesign again. This bring a meaning if the layout does not have 9 score boxes for 9 holes golf sport and 4 players details including players' names displayed, the layout will be redesigned. On the other hand, if the layout designed is suitable for golf sport with 9 score boxes for 9 holes golf sport and 4 players details including players' names displayed, then will proceed to the next stage which is the development of the software. Figure 3.2 below shows the interface of MIT App Inventor II.


MIT App Inventor
Not secure | ai2.appinventor.mit.edu/#6310977020231680
Apps Facebook Instagram Astro GO YouTube WhatsApp ULearn @ UTeM Se... UTeM | PORTAL UNI... Calendar – MICHAE... Cumulative GPA Cal... Book Meeting Room Other bookmarks
MIT APP INVENTOR
My Projects Connect Build Settings Help
My Projects View Trash Gallery Guide Report an Issue English kahchoon5283@gmail.com
GolfScorecardnew
Screen1 Add Screen ... Remove Screen
Designer Blocks
Palette
Search Components...
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Viewer
Display hidden components in Viewer
Phone size (505,320)
9:48
Golf Tournament Scorecard
PLAYER 1
PLAYER 2
PLAYER 3
PLAYER 4
SAVE
Components
Screen1
Label1
txtname1
Label2
txtname2
Label3
txtname3
Label4
txtname4
savename
FirebaseDB1
TinyDB1
Rename Delete
Media
BB_golfsport.jpg
Properties

Screen1
AboutScreen
AccentColor: Default
AlignHorizontal: Left : 1
AlignVertical: Top : 1
AppName: Golf Scorecard
BackgroundColor: Default
BackgroundImage: BB_golfsport.jpg...
BlocksToolkit: All
CloseScreenAnimation: Default
Icon: icon.jfif...
OpenScreenAnimation: Default

Figure 3.2: Interface of MIT App Inventor II
[Source: http://ai2.appinventor.mit.edu/#6310977020231680]

For next stage, all the features for the application will be assigned, and the layout will be confirmed. The features will be assigned to the application include storage for 4 players details, score that enter to the 9 score boxes and the preview of score obtained by each player and the overall score ranking in a tournament. After all the layout assignation done, the project stage will proceed to the first testing stage.

For first testing stage, the scorecard application is being tested if it can run smoothly by able to communicate and send data to Firebase online database. If the scorecard application is not able to communicate and send data to the Firebase online database, then will go back to the development stage to analyze the error. On the other hand, if the scorecard application can communicate and send the data to Firebase online database, then the project will proceed to performance testing.

The performance testing stage is to analyze both the scorecard application and the Firebase online database can run and communicate with each other without any error or fault code appear. If any error or fault code occur, then the project will go back to the performance testing stage to identify the problem and improvements will be done. Whereas, if the scorecard application and firebase online database able to communicate with each other without any error, then the project will proceed to the analyzing stage.

3.2 Milestones of the Project

There are total of five (5) milestones to be achieve at the end of this project illustrated in table 3.5 below. First, revisit previous related project is the way to start this project. Previous related project will be revisited and analysed the advantages and disadvantages compared with the current project. The information gathered from the previous related project will be apply in the current project. After all the information collected from the previous project, this project starts with the design and development of the IoT based scorecard for golf sport. The design and development stage will be the second milestone for this project. At this stage, the layout and features will be developed, and the application will be produced.

Research activities	**Milestones**
Stage 1: Revisit the previous related project / journal	1
Stage 2: Design and development of the IoT based Scorecard for golf sport	2
Stage 3: Performance testing of the IoT based scorecard system	3
Stage 4: Analysis based on the performance and experimental output	4
Stage 5: Findings Report	5

Table 3.3: Five (5) Major Stage of the Project

After that, the third milestone is to perform a performance testing of the IoT based scorecard against the traditional scorecard. The performance testing will be done by testing out all the features in the scorecard application to check whether have error occur or not. Next, an analysis will be produced based on the data collected by the performance testing and experimental output of the project. This analysis mainly based on the performance data and error that could be occur in the scorecard application during its operation. Further, a findings report will be written after the final analysis being produced. The report will include the results data collected from the performance testing, and the analysis results.

CHAPTER 4

PROJECT IMPLEMENTATION

4.0 Introduction

This chapter will discuss about the hardware used and the implementation of the project. Block diagram and estimated cost of this project will also be provided.

4.1 Block Diagram

Based on Figure 4.1, there will be one input device and two components needed to be used as output devices. The input device to be used is either an Android based smartphone or an Android based tablet which will be used as the scorecard of the system. The device is used to key in the stroke score that golfers obtained in the game. The data include the stroke score of nine holes in the golf course with expected PAR values and the actual PAR values obtained for four players in one device. After that, the output device is computer or a smart TV. The computer or smart TV that logged in to the online database website will display the data stored in the database.

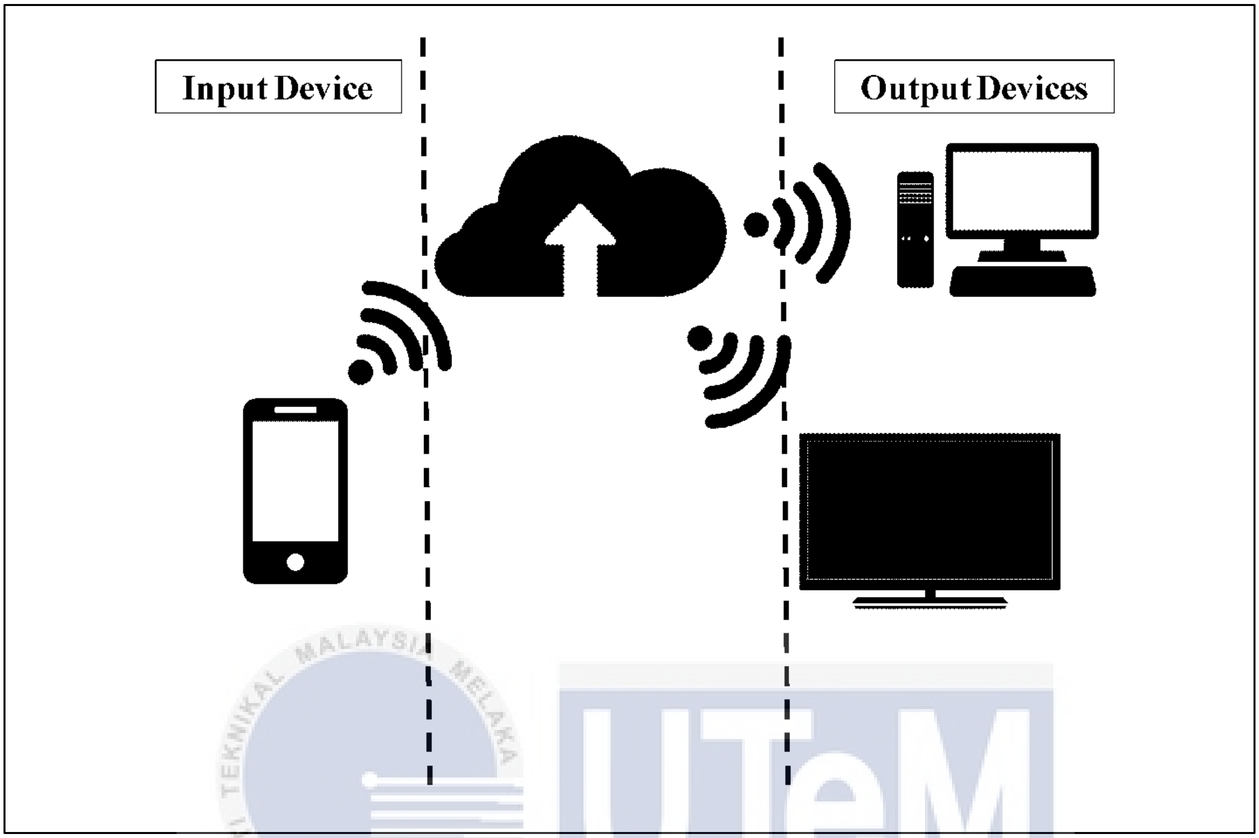


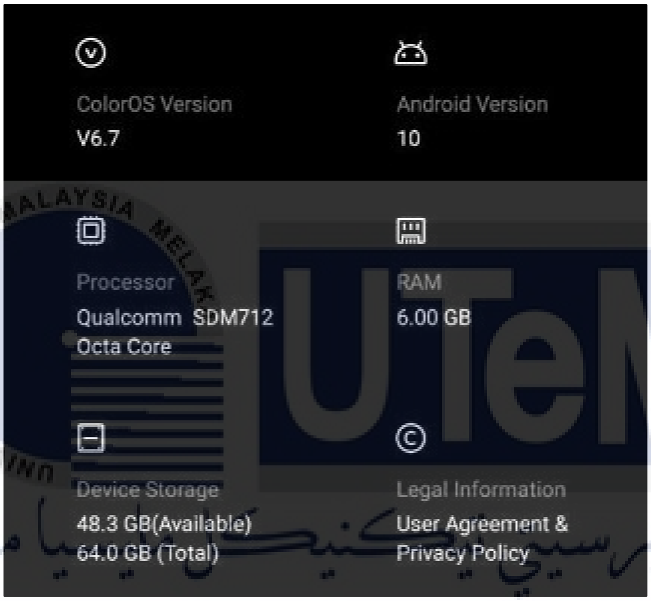
Figure 4.1: Block Diagram of the Project

4.2 Hardware Development

The hardware used in this project includes an Android based smart device, either a smartphone or a tablet is acceptable. Other than that, the online database used is Google Firebase online database system.

4.2.1 Android-based Smart Device

The input device of this project is an Android based smart device. Any smartphone or tablet based on Android version 5.1 (Android Lollipop) and above will compatible with the scorecard application. The device must have at least 100 Megabytes of free space and 1 Gigabytes of RAM to run the application smoothly without any lagging. Other than that, with presence of an octa core processor will further enhance the user experience of the scorecard application. This device is used to input the players details and players' score to the scorecard application. The specification details can be found in settings application, about phone tab for any Android devices. Figure 4.4 below shows a sample specification for the scorecard application.



ColorOS Version
V6.7

Android Version
10

Processor
Qualcomm SDM712
Octa Core

RAM
6.00 GB

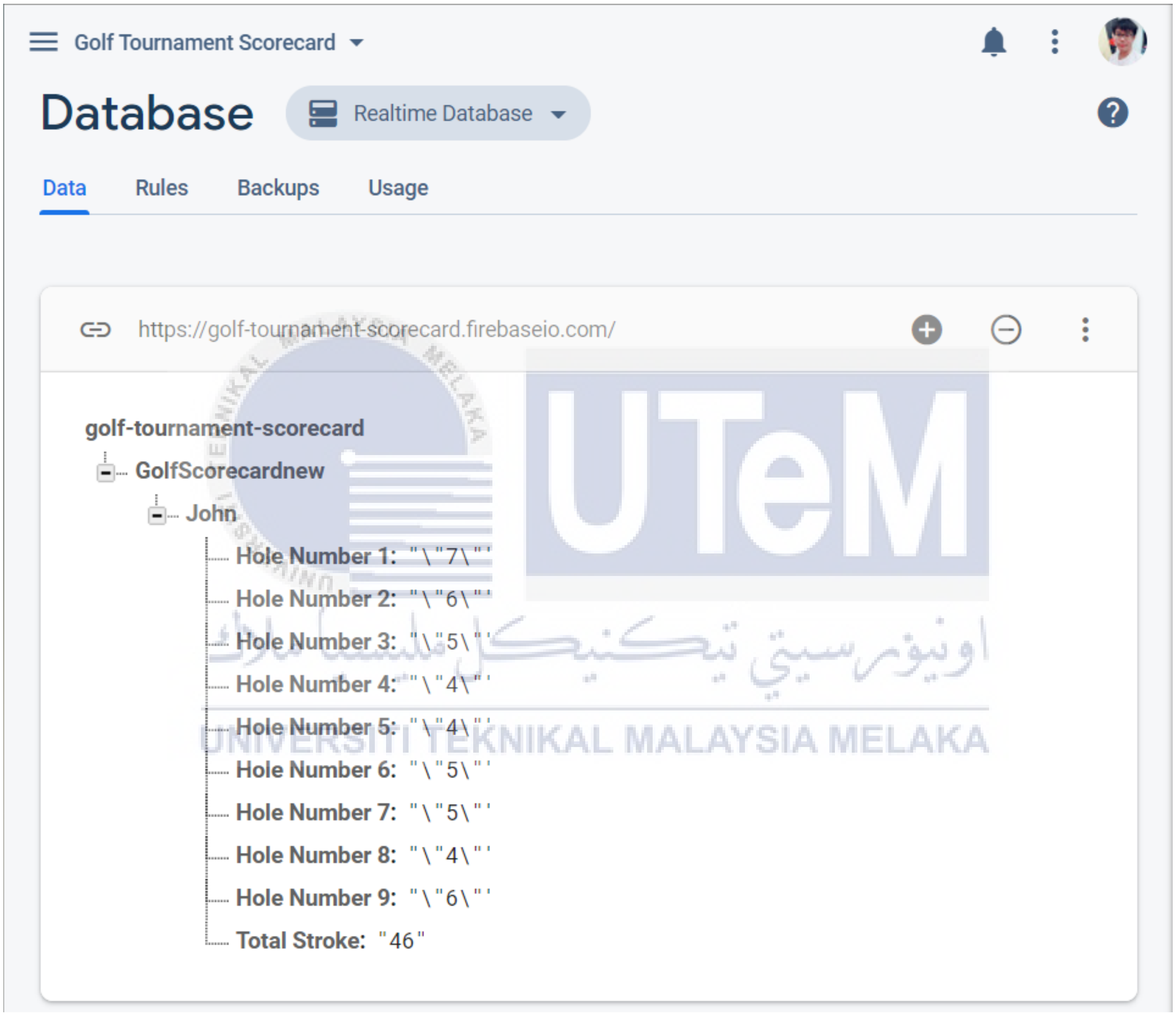
Device Storage
48.3 GB(Available)
64.0 GB (Total)

Legal Information
User Agreement &
Privacy Policy

Figure 4.4: Sample Specification of Android device

4.2.2 Google Firebase Online Database

Google Firebase is a mobile and web application development platform. There are several Firebase services include online database system and cloud storage. Google firebase will be chosen as the online database for this project because it is free to access and it has a user-friendly interface. The Firebase database requires Google account sign in to use the services provided. Then, there is a unique address used to bind the Firebase online database to the scorecard application. So, whenever the scorecard application sent any data to the cloud, the Firebase will display it as a real-time data on the spot. The online database can be access with the same account authentication anywhere and anytime. The sample of the online database layout with data uploaded is illustrated in Figure 4.5 below.



https://golf-tournament-scorecard.firebaseio.com/

```
golf-tournament-scorecard
  GolfScorecardnew
    John
      Hole Number 1: "\"7\""
      Hole Number 2: "\"6\""
      Hole Number 3: "\"5\""
      Hole Number 4: "\"4\""
      Hole Number 5: "\"4\""
      Hole Number 6: "\"5\""
      Hole Number 7: "\"5\""
      Hole Number 8: "\"4\""
      Hole Number 9: "\"6\""
      Total Stroke: "46"
```

Figure 4.5: Google Firebase Layout with Data

4.3 Scorecard Implementation in Golf Tournament

During a golf tournament, golfers will record their score obtained by using the golf scorecard. This section will discuss the implementation of current golf scorecard and IoT based golf scorecard during a tournament with example. Figure 4.7 below shows the sample golf line for golfers group A and group B participated in a 9 holes golf course in a tournament. The same golf line will be applied to both current golf scorecard example and IoT based golf scorecard example.

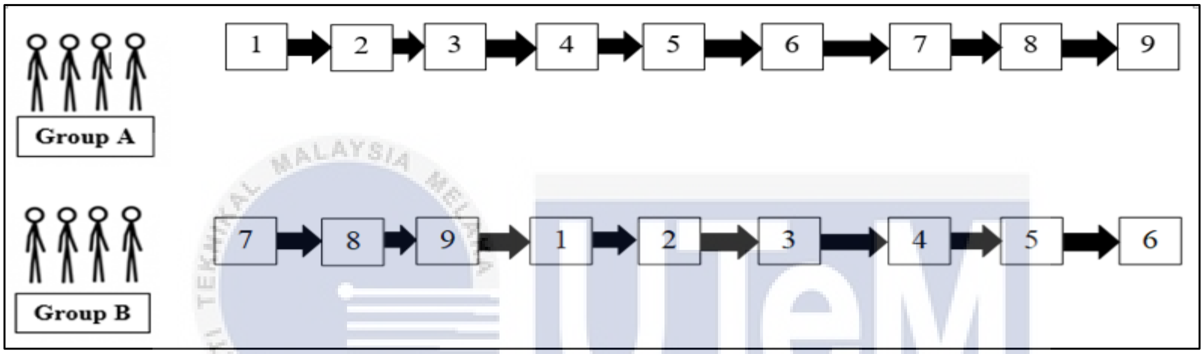


Figure 4.7: Golf Line for Group A and Group B

4.3.1 Implementation of Current Golf Scorecard

The implementation of the current golf scorecard will require a physical golf scorecard used to record the score obtained during a tournament. Figure 4.8 show the tournament starts at 8am in the morning and both group A and B start their game at the respective starting point. The score obtained will be recorded in the physical golf scorecard provided by the golf club.

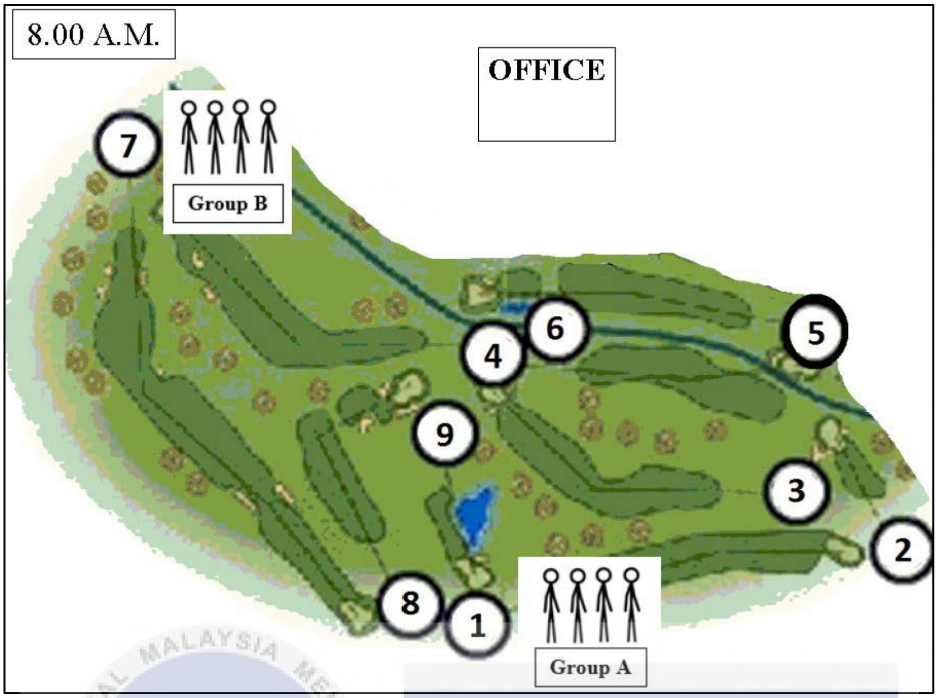


Figure 4.8: The Golf Tournament Starts at 8AM

At 10AM, the tournament progress for group A and group B is illustrated in figure 4.9 below. Group A is currently at hole number 5 whereas group B currently at hole number 2 of the golf course.

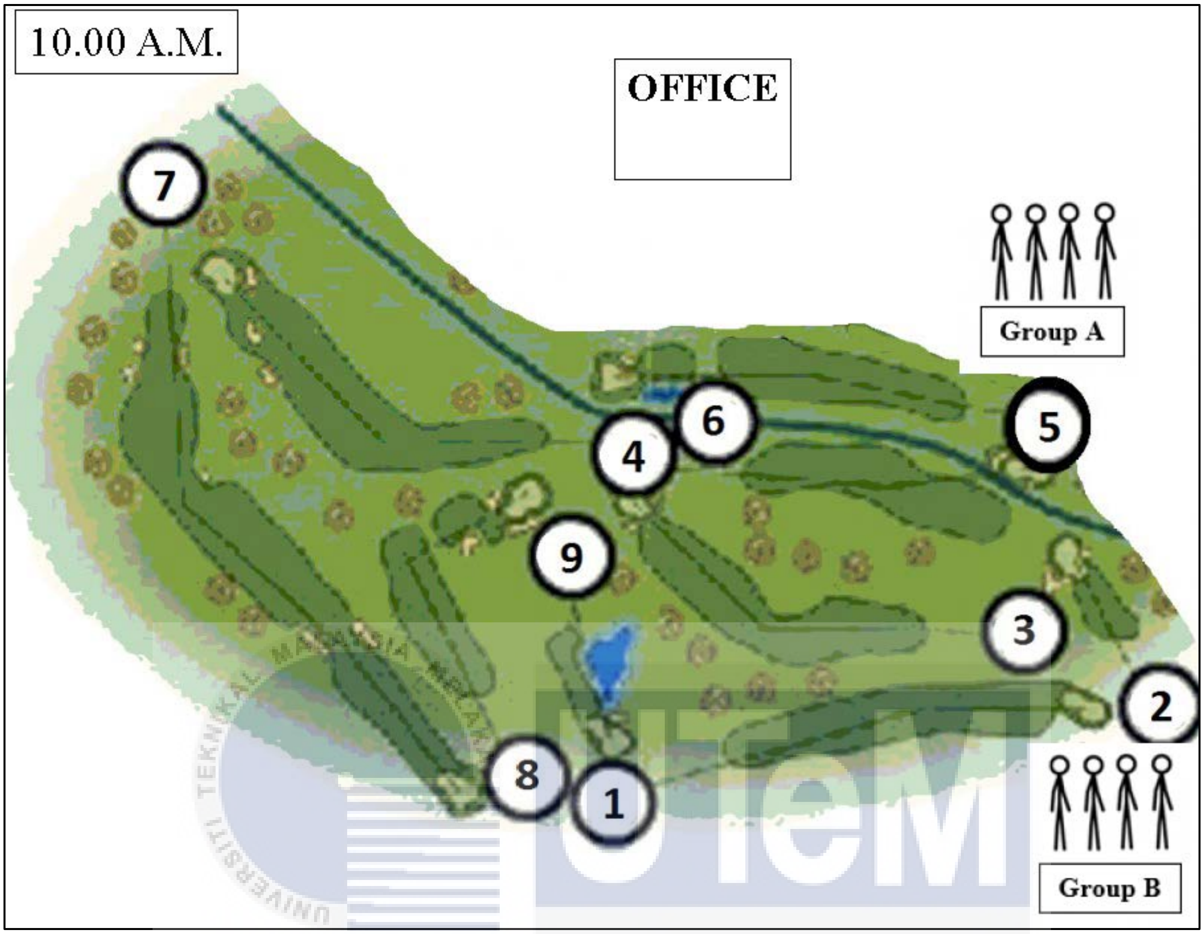


Figure 4.9: Progress of Group A and Group B at 10AM

After that, both group A and B are playing at the final hole at 12PM. Figure 4.10 shows the final progress of players in group A and group B at 12PM. Group A is currently at hole number 9 and group B is current at hole number 6.

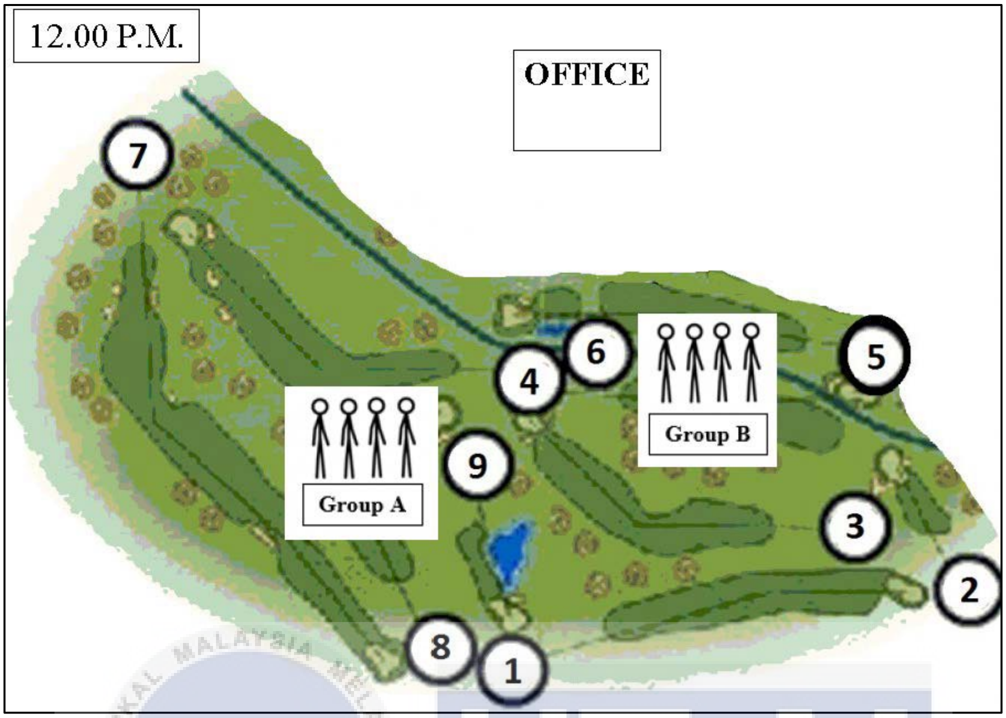


Figure 4.10: Final Progress for Group A and Group B

After group A and group B finished their game at 12PM, the scorecard with score obtained by them will submitted to the office of the golf club for the winner analysis. Figure 4.11 below shows the winner of the tournament announced at 1.30PM.

Figure 4.11: Winner Announcement at 1.30PM

The golf tournament game ended at 12PM but the winner analysis takes time of an hour and a half to finish. In this case, the golf tournament ended at 1.30PM.

4.3.2 Implementation of IoT based Golf Scorecard

The IoT based scorecard allow golfer to submit their score obtained right after they finish their games in every golf holes that they played by using the Android based scorecard application. Figure 4.12 shows the golfers group A and group B started their games at 8AM and submit their score by the scorecard application.

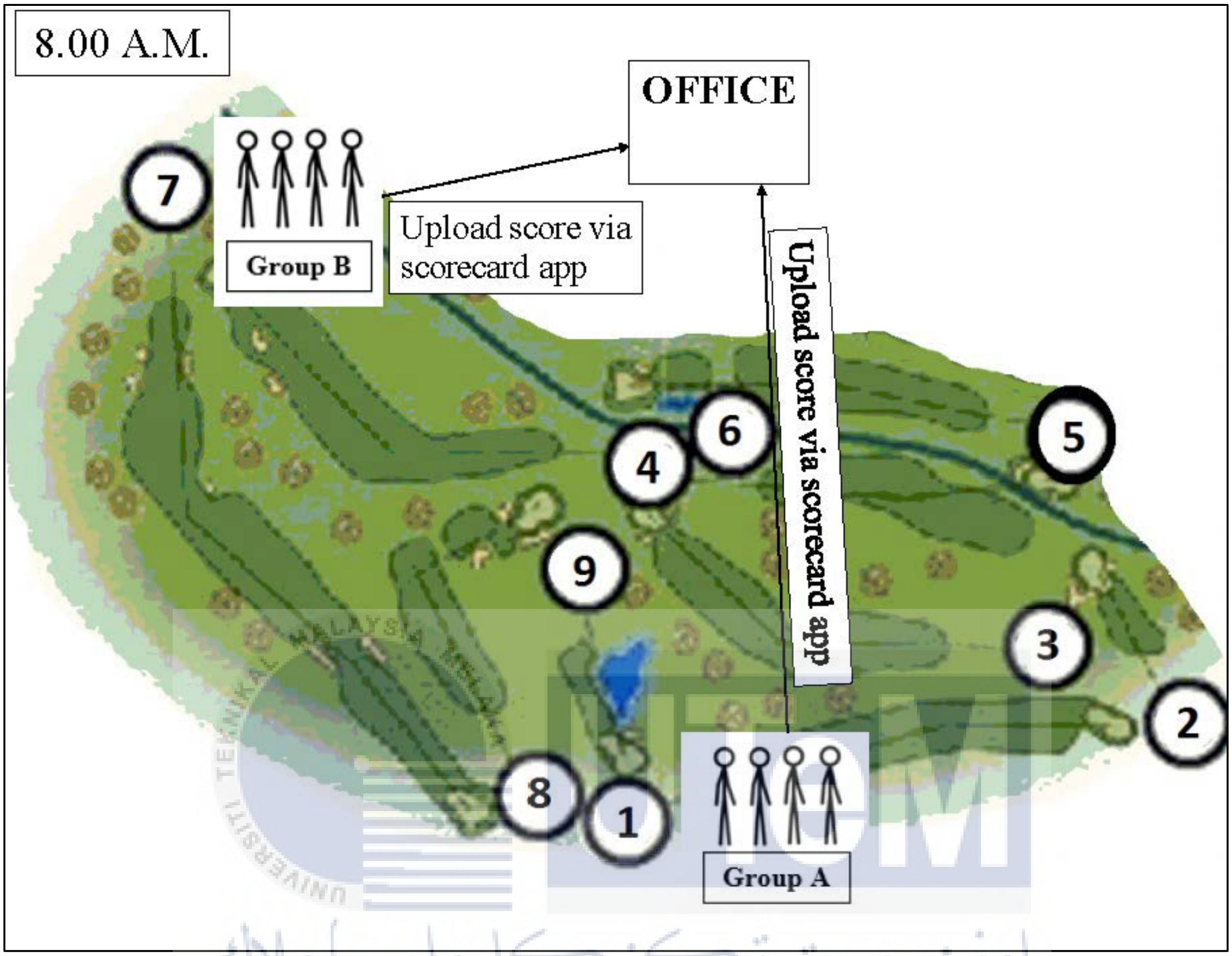

8.00 A.M.
OFFICE
Group B
Upload score via scorecard app
Upload score via scorecard app
Group A
7
6
4
5
9
3
2
8
1

Figure 4.12: Golfers Submit the Score Obtained via Scorecard Application at 8AM

After that, the progress of both group A and group B current until golf course hole number5 and hole number 2 respectively. Figure 4.13 below illustrated the progress of group A and group B at 10AM during the tournament.

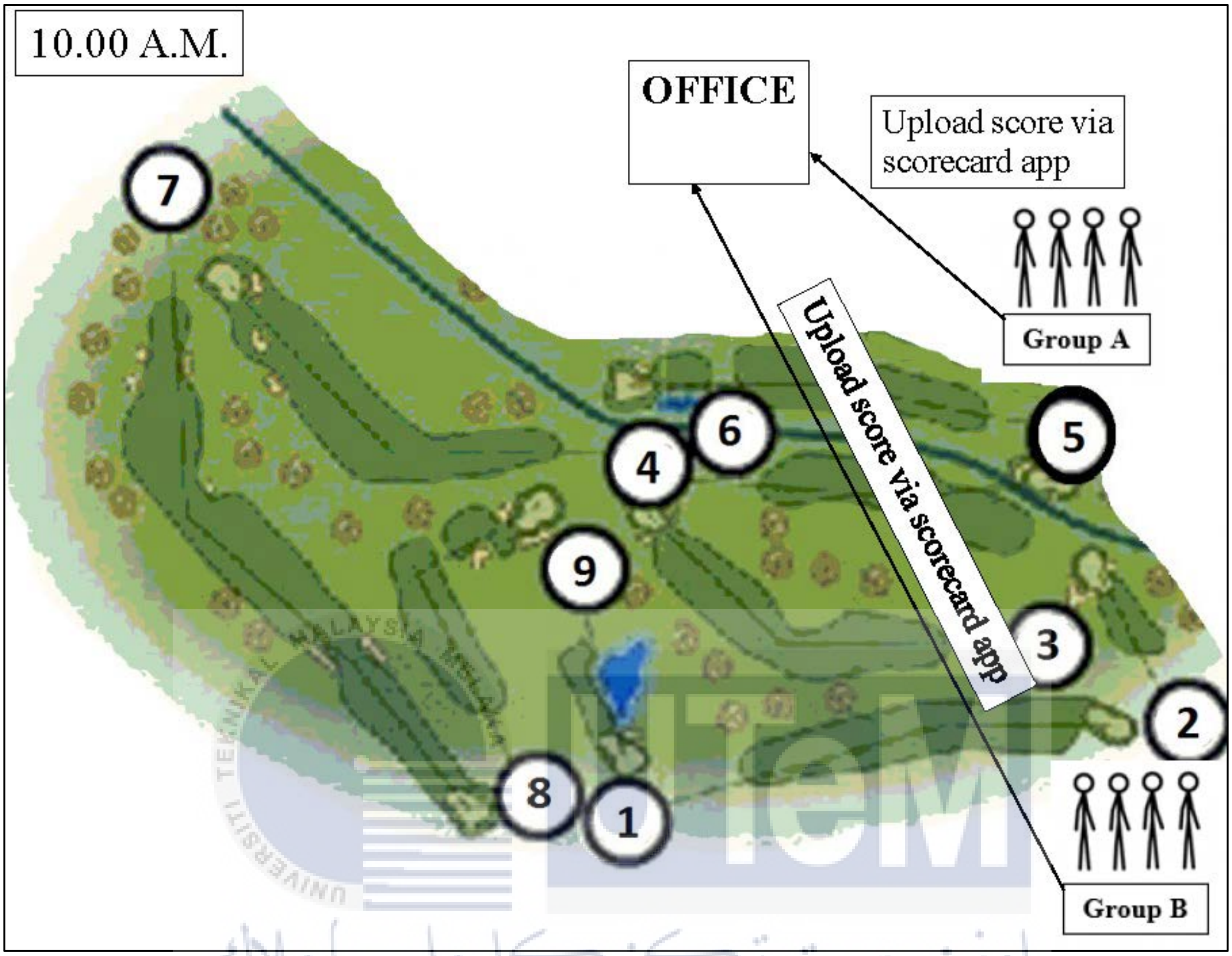

10.00 A.M.
OFFICE
Upload score via scorecard app
Group A
Upload score via scorecard app
Group B
7
6
4
5
9
3
2
8
1

Figure 4.13: Progress of Group A and Group B at 10AM

At 12PM sharp, both groups proceeded to their final checkpoint which is hole number 9 for group A and hole number 6 for group B. Figure 4.14 below shows the final progress for group A and group B. All the score obtained during the tournament is uploaded to the online database of the IoT base golf scorecard in real-time manner. So, the winner can be analysed right after the game ended. The winner announcement of the tournament at 12PM is shown in figure 4.15 below.

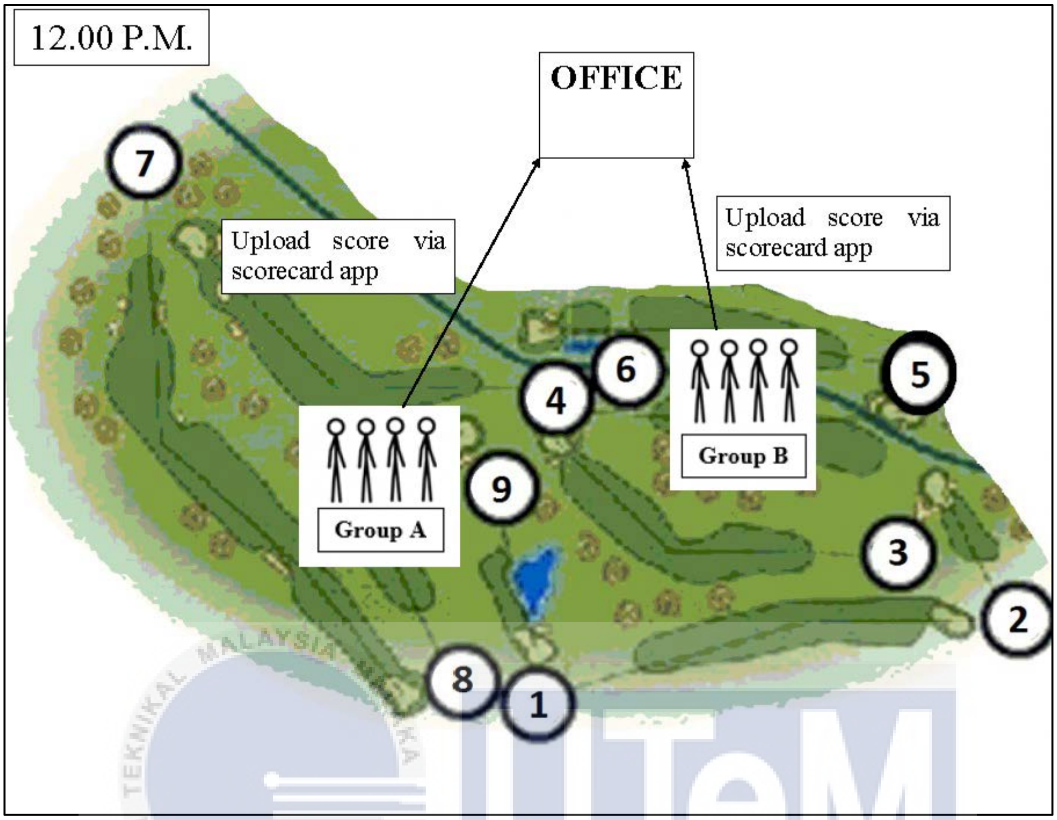

12.00 P.M.
OFFICE
Upload score via scorecard app
Upload score via scorecard app
Group A
Group B
7
6
4
5
9
3
2
8
1

Figure 4.14: Final Progress for Group A and Group B

Figure 4.15: Winner Announcement at 12PM

CHAPTER 5

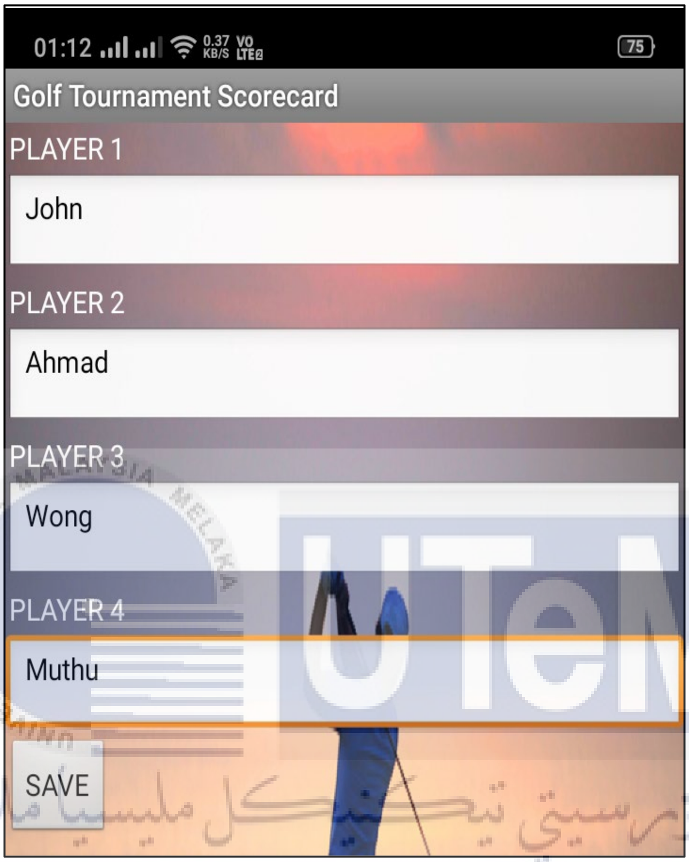
RESULTS AND ANALYSIS

5.0 Introduction

This section will discuss about the data that collected from the performance testing of the product to analyse the accuracy and stability of the IoT based golf scorecard.

5.1 Results

The application developed is tested on an Android based smartphone with 4 Gigabytes of RAM and more than 100 Megabytes of free space running Android version 9.0 (Android Pie). The application can be run smoothly on this device without any error. The application consists of 6 different pages. Figure 5.4 shown the main page of the scorecard application. The main page used to input the players' details for 4 players include their names. When all the names are successfully key in, press the “SAVE” button to store the players’ details and the application will go to the next page automatically.



01:12 0.37 KB/S VO LTE 75

Golf Tournament Scorecard

PLAYER 1

John

PLAYER 2

Ahmad

PLAYER 3

Wong

PLAYER 4

Muthu

Figure 5.1: Main Page of the Scorecard Application

After that, Figure 5.2 illustrate the second page of the application. The second page is used to select player’s name to record their score obtained and the overall score ranking option. The 4 players name can be selected in order to record the score for them. Figure 5.3 shown the score record page for player 1 which is named John for example. Figure 5.4 shown the player 1 score recording page with sample score keyed in. In this page, the score recording area consists of score for 9 different holes and expected PAR values for each hole. After the score being enter, press “SEND” button to send the keyed in data to the online database and the total stroke beside the “SEND” button will display the total stroke obtained for that particular player by that time. The score recording page also available for player 2, player 3 and player 4 by clicking their names in the second page. Furthermore, the Firebase layout is illustrated in Figure 5.5 and Figure 5.6 below for the scorecard application and computer website respectively with the sample score of player 1, named John for demonstration purpose. The Firebase layout will display the score obtained for respective holes and also the total stroke score obtained for the player.

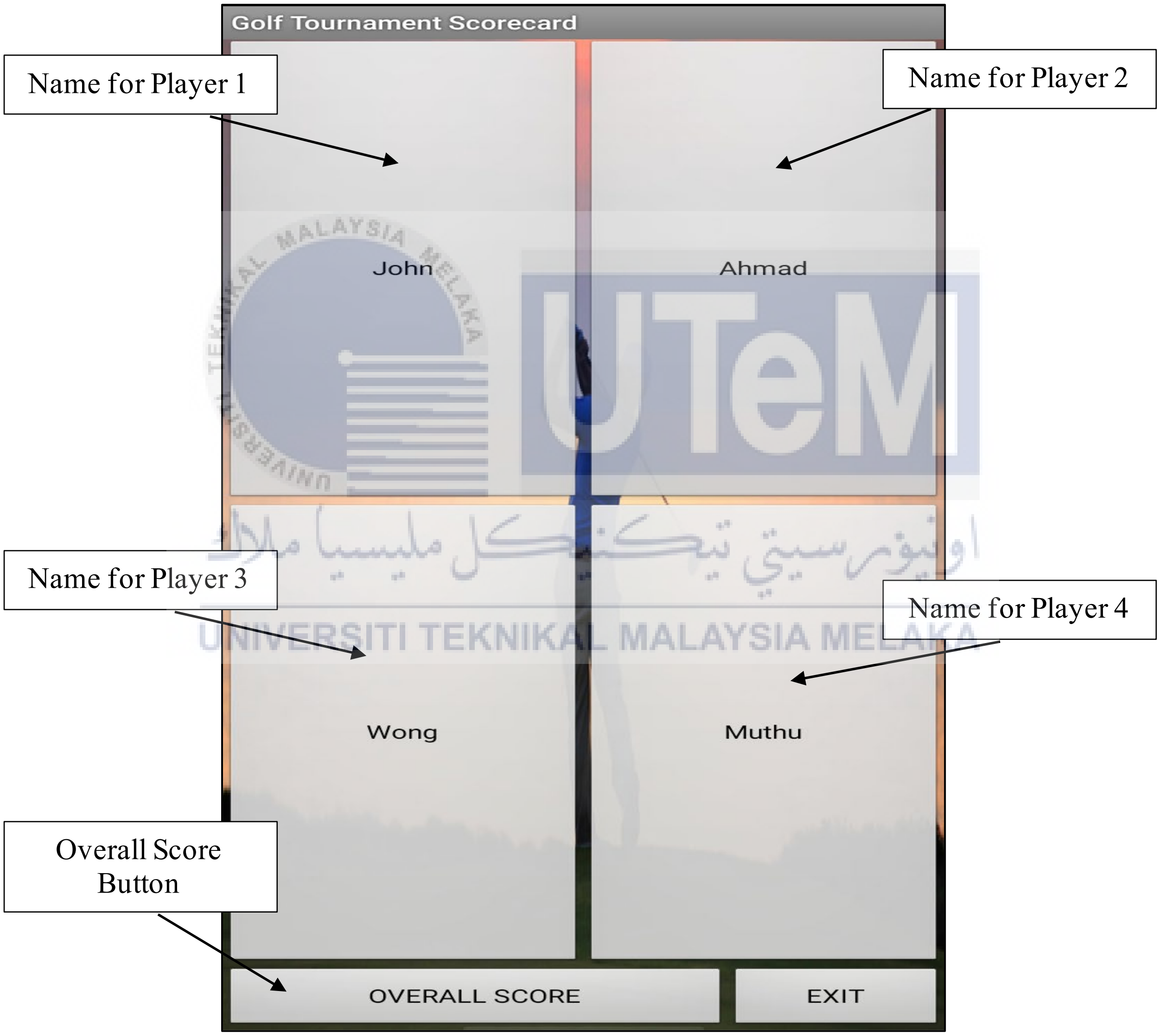
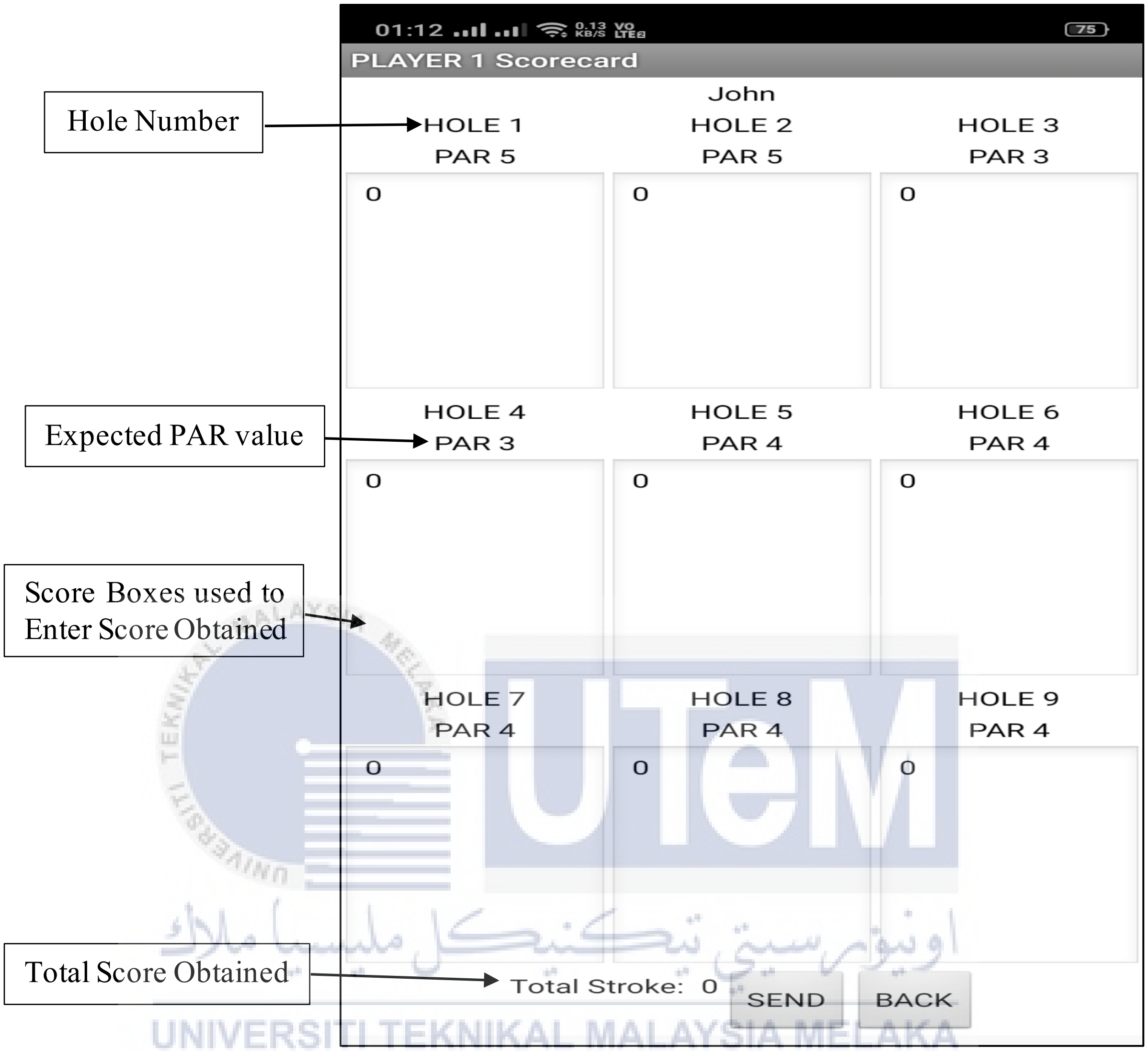


Figure 5.2: Players Selection Page



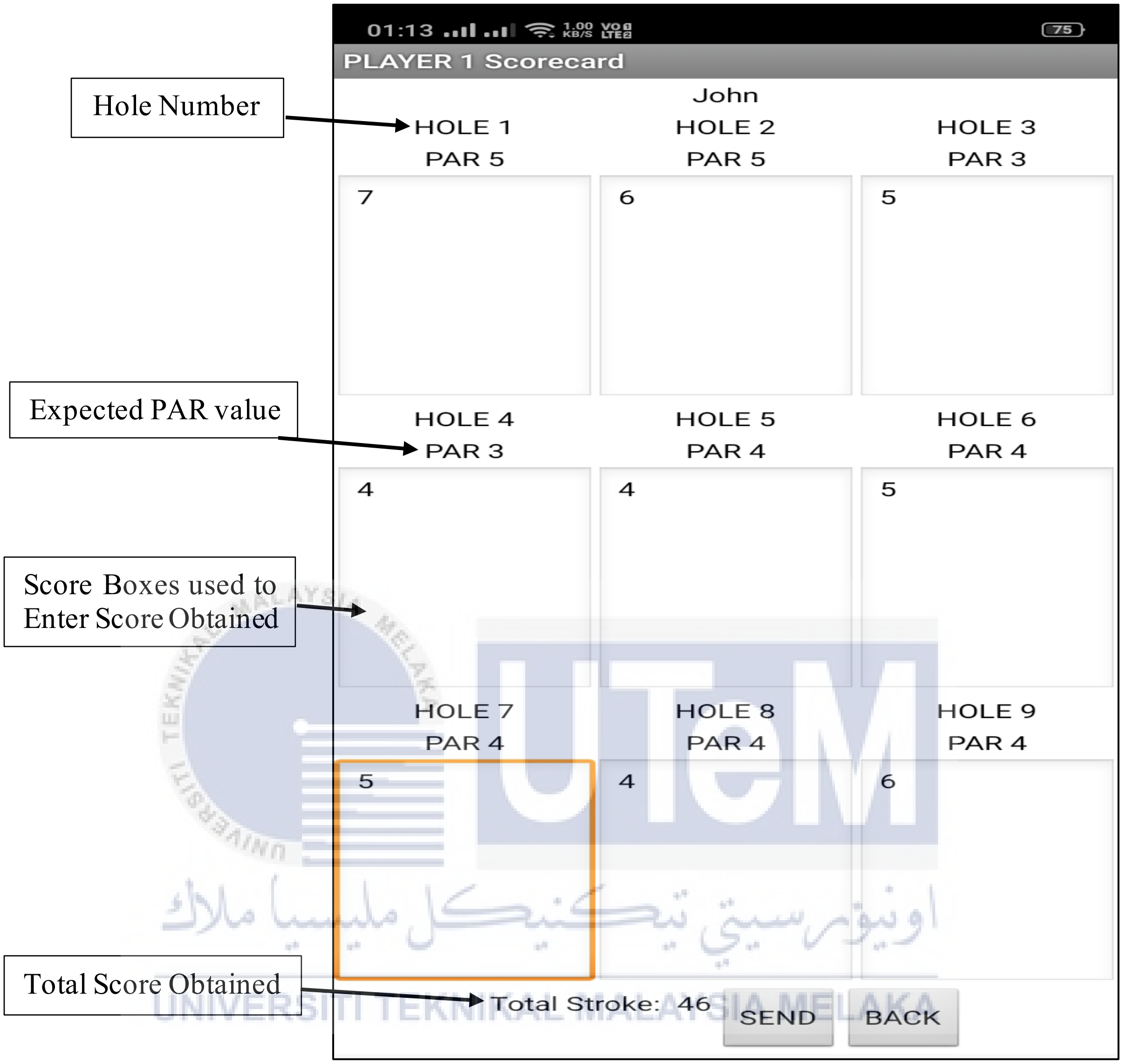
PLAYER 1 Scorecard

John

HOLE 1 PAR 5	HOLE 2 PAR 5	HOLE 3 PAR 3
0	0	0
HOLE 4 PAR 3	HOLE 5 PAR 4	HOLE 6 PAR 4
0	0	0
HOLE 7 PAR 4	HOLE 8 PAR 4	HOLE 9 PAR 4
0	0	0

Total Stroke: 0

Figure 5.3: Score Recording Page for Player 1



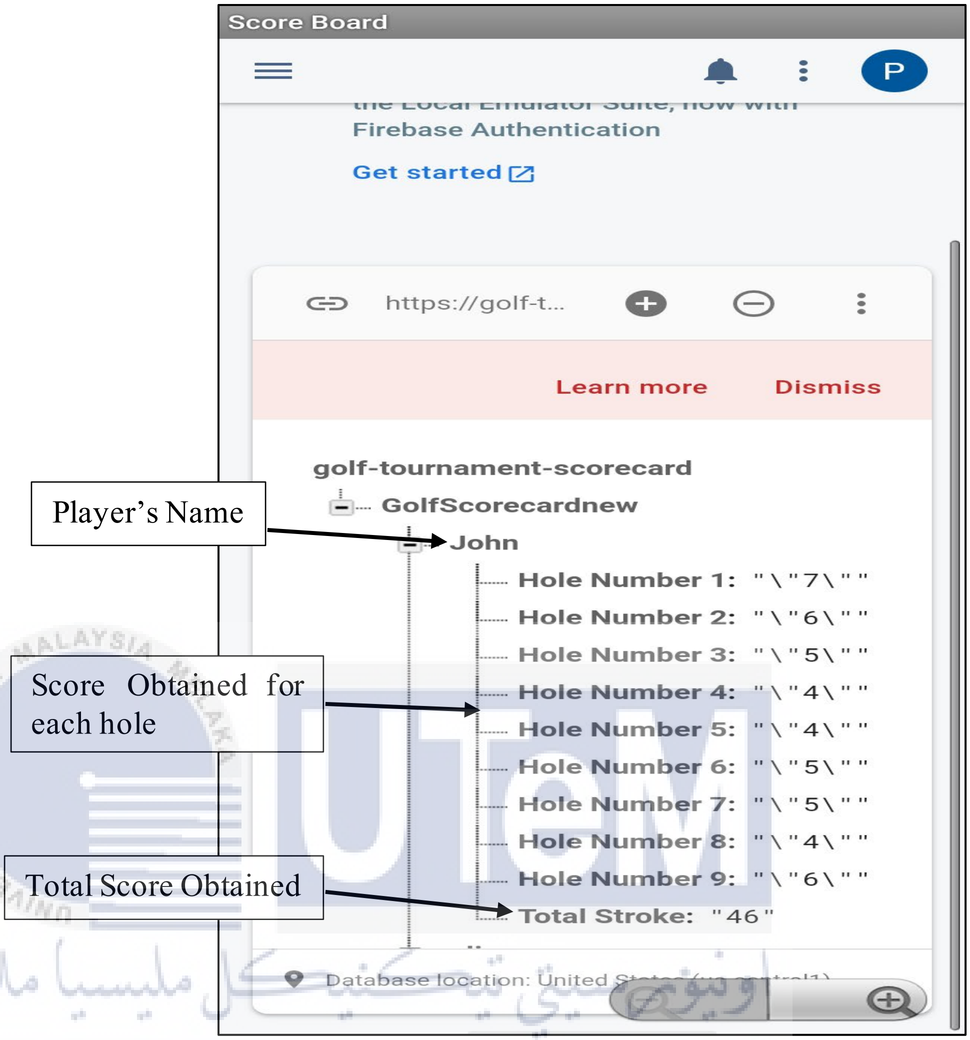
PLAYER 1 Scorecard

John

HOLE 1 PAR 5	HOLE 2 PAR 5	HOLE 3 PAR 3
7	6	5
HOLE 4 PAR 3	HOLE 5 PAR 4	HOLE 6 PAR 4
4	4	5
HOLE 7 PAR 4	HOLE 8 PAR 4	HOLE 9 PAR 4
5	4	6

Total Stroke: 46

Figure 5.4: Score Recording Page for Player 1 with Sample Score



Score Board

the Local Emulator Suite, now with
Firebase Authentication

Get started

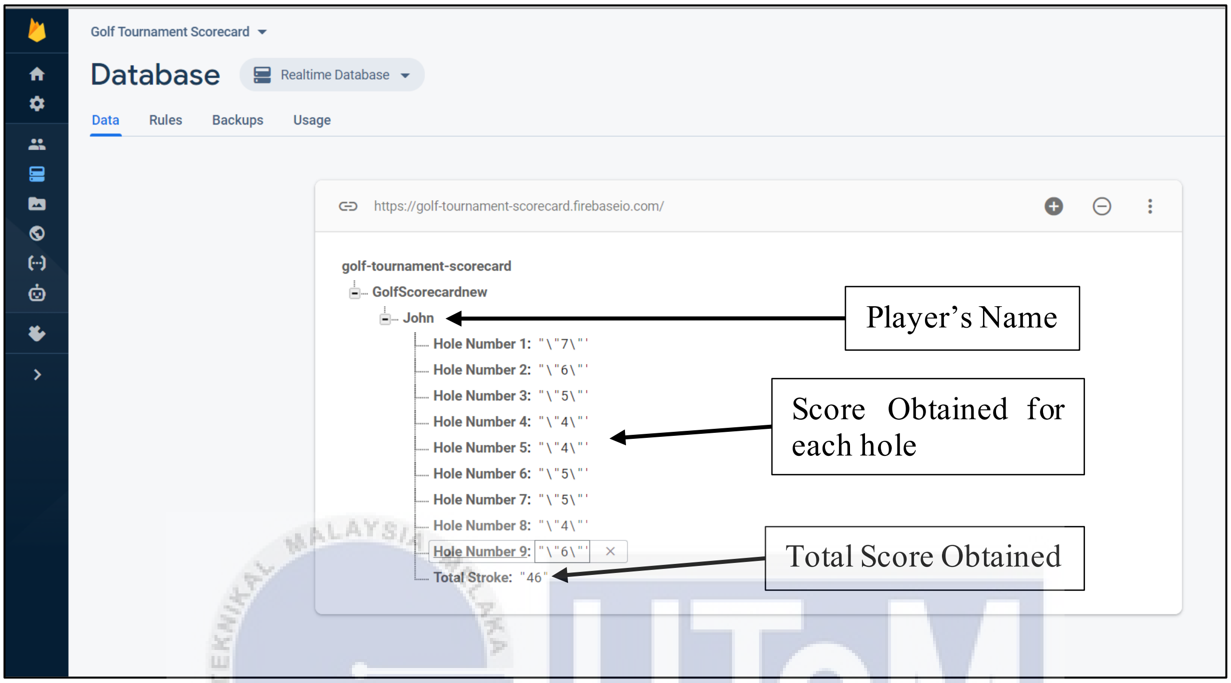
https://golf-t...

Learn more Dismiss

```
golf-tournament-scorecard
  GolfScorecardnew
    John
      Hole Number 1: "\"7\""
      Hole Number 2: "\"6\""
      Hole Number 3: "\"5\""
      Hole Number 4: "\"4\""
      Hole Number 5: "\"4\""
      Hole Number 6: "\"5\""
      Hole Number 7: "\"5\""
      Hole Number 8: "\"4\""
      Hole Number 9: "\"6\""
      Total Stroke: "46"
```

Database location: United States (us-central1)

Figure 5.5: Firebase Layout in the Scorecard App



https://golf-tournament-scorecard.firebaseio.com/

```
golf-tournament-scorecard
  GolfScorecardnew
    John
      Hole Number 1: "\"7\""
      Hole Number 2: "\"6\""
      Hole Number 3: "\"5\""
      Hole Number 4: "\"4\""
      Hole Number 5: "\"4\""
      Hole Number 6: "\"5\""
      Hole Number 7: "\"5\""
      Hole Number 8: "\"4\""
      Hole Number 9: "\"6\""
      Total Stroke: "46"
```

Player’s Name

Score Obtained for each hole

Total Score Obtained

Figure 5.6: Firebase Layout with Sample Score of Player 1

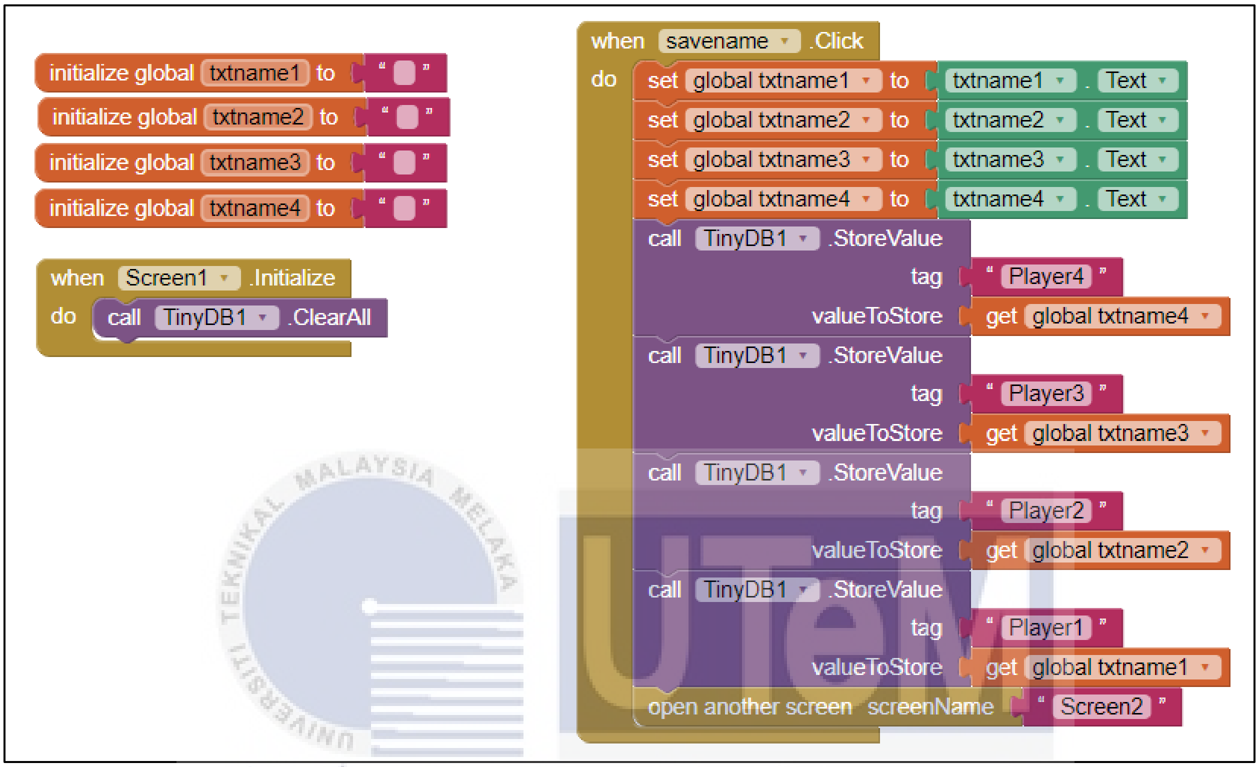
5.2 Analysis

The result for this project is to produce an Android based scorecard application with no error communicate with the online database. The block language coding is the key to ensure there are no error. There are several types of error can be analysed from the block language followed by the page of the application. There are total of 7 pages in the application, where screen 1 is the main page of the application, screen 2 is the player selection page, screen 3 to screen 6 is the score recording page for the 4 players respectively. Then, screen 7 is the overall score display page. There are several types of blocks used in this project which is illustrated in the table 5.7.

Colour	Function	Description
Orange 🟧	Declaration blocks	Used to declare global variables
Pink 🟥	Variable blocks	Used to store and call global variables
Gold 🟨	Action blocks	To perform an action when specific button or screen initialized
Purple 🟪	Database blocks (FirebaseDB/TinyDB)	Used to store and call data stored in the database
Light/Dark Green 🟩 🟩	Text blocks	Assign text string to specific declaration blocks
Blue 🟦	Mathematics blocks	Used to perform mathematical operation

Table 5.7: Types of Blocks

Figure 5.6 shows the block language for the main page of the scorecard application. There are several types of error can be caused by the absence of some specific blocks. First, if there is an absence of the declaration blocks, the mobile application will not be able to declare the variable enter by users. This will cause the mobile application unable to perform declaration and cannot save the data enter by user. Then, absence of the variable blocks will carry a similar error which the variable enter by user cannot be store. After that, if there is an absence of the TinyDB database blocks, the scorecard application will not be able to save the names enter by the user in the offline database of the mobile application. For the absence of the action blocks, the scorecard application will not perform any action after the screen initialized or the button being clicked. Table X.X shows the effects for the absence of specify blocks.



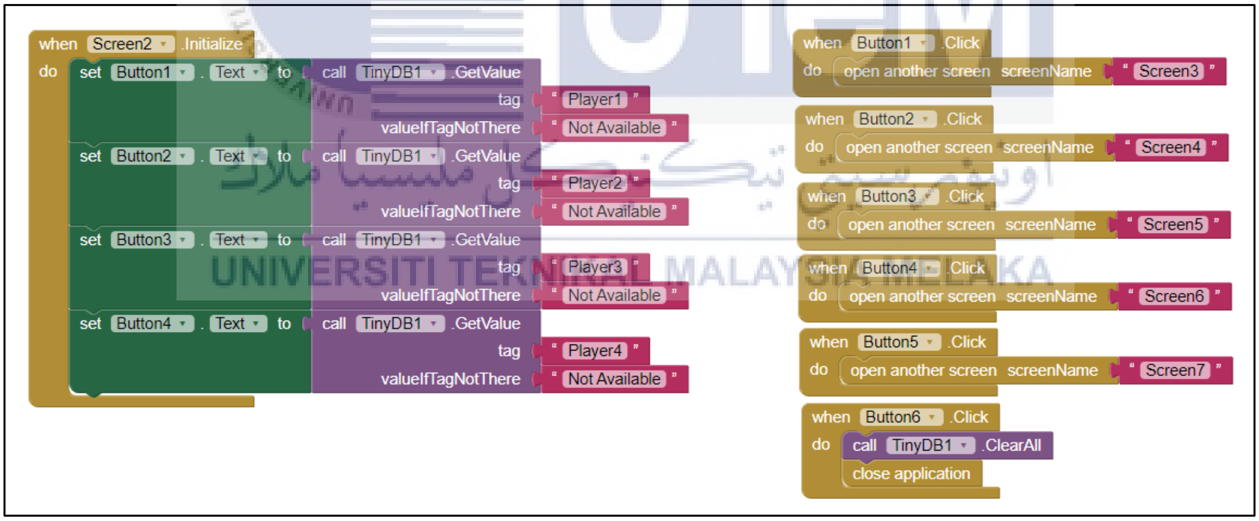
```
initialize global txtname1 to " "
initialize global txtname2 to " "
initialize global txtname3 to " "
initialize global txtname4 to " "

when Screen1 .Initialize
do   call TinyDB1 .ClearAll

when savename .Click
do   set global txtname1 to txtname1 . Text
     set global txtname2 to txtname2 . Text
     set global txtname3 to txtname3 . Text
     set global txtname4 to txtname4 . Text
     call TinyDB1 .StoreValue
                       tag " Player4 "
              valueToStore get global txtname4
     call TinyDB1 .StoreValue
                       tag " Player3 "
              valueToStore get global txtname3
     call TinyDB1 .StoreValue
                       tag " Player2 "
              valueToStore get global txtname2
     call TinyDB1 .StoreValue
                       tag " Player1 "
              valueToStore get global txtname1
     open another screen  screenName " Screen2 "
```

Figure 5.8: Block Language for Main Page of the Scorecard Application

Blocks/Effects **Case**	**Declaration Blocks**	**Variable Blocks**	**TinyDB Blocks**	**Action Blocks**	**Effects**
Case 1	**Absent**	Present	Present	Present	Application not able to declare variables.
Case 2	Present	**Absent**	Present	Present	Application not able to store variables.
Case 3	Present	Present	**Absent**	Present	Application not able to store the name enter by users.
Case 4	Present	Present	Present	**Absent**	Application not able to perform actions.

Table 5.9: Effects for Absence of Specific Blocks for Screen 1

Figure 5.7 illustrate the block language programming for screen 2 of the scorecard application, which is the player selection page. This page consists of fewer blocks which most of it are action blocks because this page mainly used to direct users to the desired player's score recording page and the overall score board. The error can be occurred here will be the lack of TinyDB blocks because the button used to select players will not be able to display as the player's names since the TinyDB blocks are used to store the name enter by users earlier. Other than that, the action blocks are necessary due to it is the action to direct users to desired player's score recording page. If there are no action blocks here, users are not able to be directed to the score recording page. Furthermore, the text blocks are used to set the specific text box to certain text. In this case, the text blocks are used to set the text of the button to the players' name that stored in TinyDB.



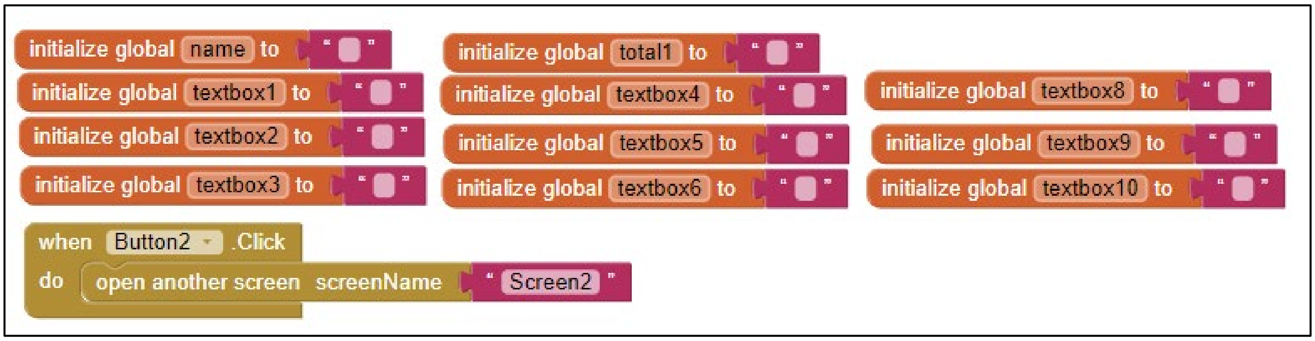
```
when Screen2 .Initialize
do  set Button1 . Text to  call TinyDB1 .GetValue
                                               tag  " Player1 "
                                valueIfTagNotThere  " Not Available "
    set Button2 . Text to  call TinyDB1 .GetValue
                                               tag  " Player2 "
                                valueIfTagNotThere  " Not Available "
    set Button3 . Text to  call TinyDB1 .GetValue
                                               tag  " Player3 "
                                valueIfTagNotThere  " Not Available "
    set Button4 . Text to  call TinyDB1 .GetValue
                                               tag  " Player4 "
                                valueIfTagNotThere  " Not Available "

when Button1 .Click
do  open another screen  screenName  " Screen3 "

when Button2 .Click
do  open another screen  screenName  " Screen4 "

when Button3 .Click
do  open another screen  screenName  " Screen5 "

when Button4 .Click
do  open another screen  screenName  " Screen6 "

when Button5 .Click
do  open another screen  screenName  " Screen7 "

when Button6 .Click
do  call TinyDB1 .ClearAll
    close application
```

Figure 5.10: Block Language for Screen 2

Blocks/Effects Case	Tiny DB Blocks	Action Blocks	Text Blocks	Effects
Case 1	**Absent**	Present	Present	Application not able to call out the names stored in TinyDB
Case 2	Present	**Absent**	Present	Application not able to direct users to the desired player's score recording page.
Case 3	Present	Present	**Absent**	Application not able to display the players' names on the button or text box.

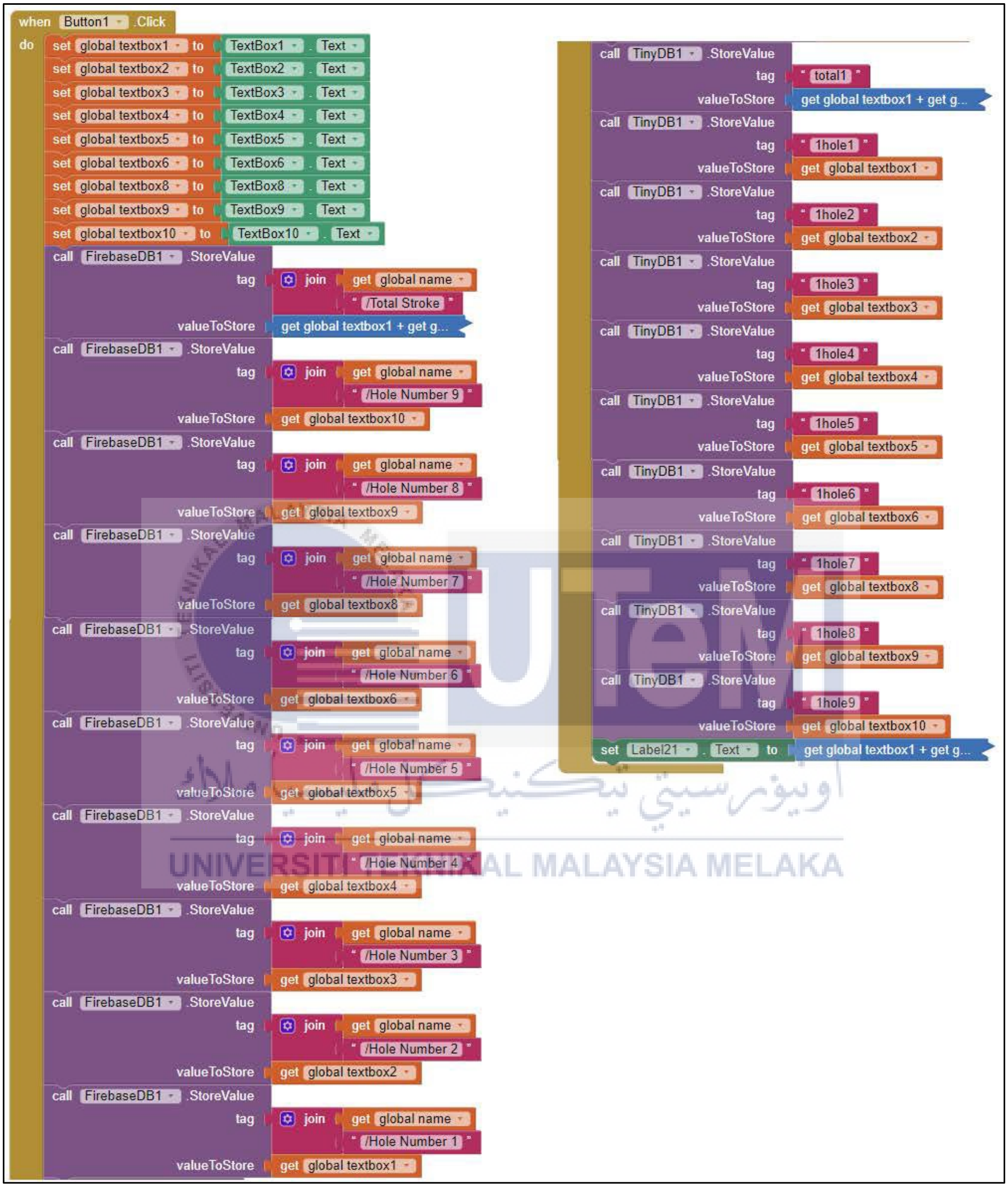
Table 5.11: Effects for Absence of Specific Blocks for Screen 2

The block language programming for screen 3 which is the player's score recording page is illustrate in figure 5.8, 5.9 and 5.10. The block language programming for screen 4, screen 5 and screen 6 are similar to screen 3. The declaration blocks are important because if declaration problems occur, the whole page will be error. Other than that, the Firebase blocks are used to communicate and send data to the online database. Absence of Firebase blocks will cause the scorecard application unable to communicate and send the real-time score to the Firebase online database. Further, the TinyDB blocks are used to save the players' score in the scorecard application and process the score with a mathematics block to analyse the total score obtained by each player. The text blocks are used to display the players' names stored in TinyDB, the names will not be display without the text blocks.



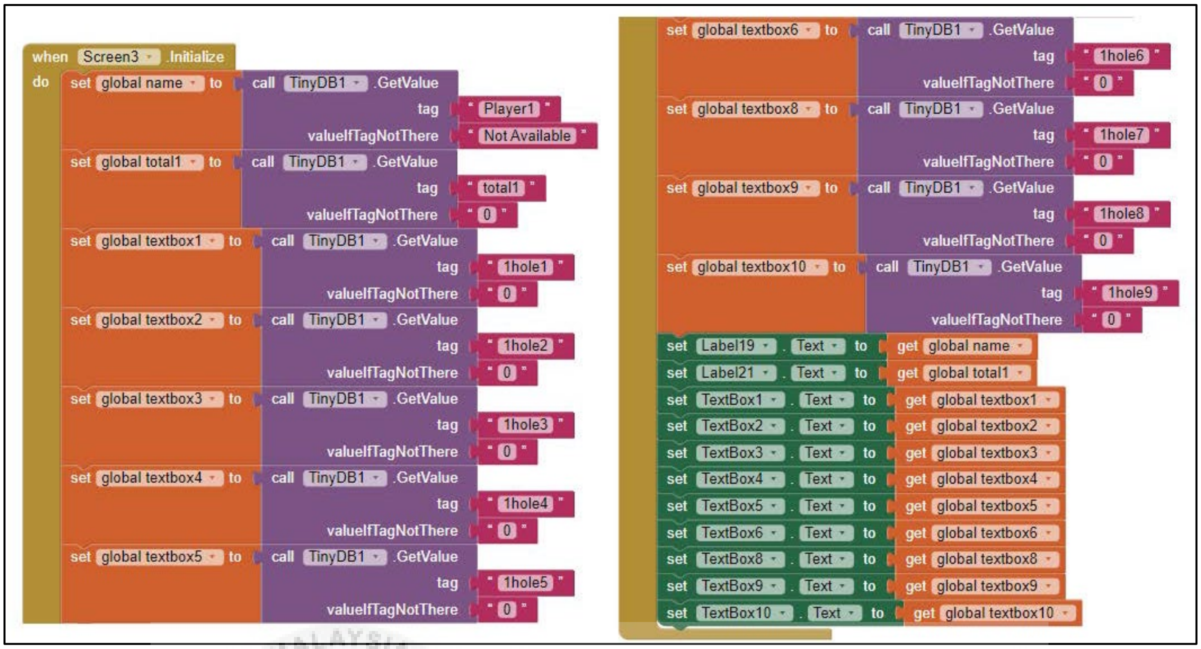
```
initialize global name to " "
initialize global textbox1 to " "
initialize global textbox2 to " "
initialize global textbox3 to " "
initialize global total1 to " "
initialize global textbox4 to " "
initialize global textbox5 to " "
initialize global textbox6 to " "
initialize global textbox8 to " "
initialize global textbox9 to " "
initialize global textbox10 to " "
when Button2 .Click
do open another screen screenName " Screen2 "
```

Figure 5.12: Declaration Blocks for Players' Score Recording Page



```
when Button1 .Click
do  set global textbox1 to TextBox1 . Text
    set global textbox2 to TextBox2 . Text
    set global textbox3 to TextBox3 . Text
    set global textbox4 to TextBox4 . Text
    set global textbox5 to TextBox5 . Text
    set global textbox6 to TextBox6 . Text
    set global textbox8 to TextBox8 . Text
    set global textbox9 to TextBox9 . Text
    set global textbox10 to TextBox10 . Text
    call FirebaseDB1 .StoreValue
                tag  join  get global name
                           " /Total Stroke "
       valueToStore  get global textbox1 + get g...
    call FirebaseDB1 .StoreValue
                tag  join  get global name
                           " /Hole Number 9 "
       valueToStore  get global textbox10
    call FirebaseDB1 .StoreValue
                tag  join  get global name
                           " /Hole Number 8 "
       valueToStore  get global textbox9
    call FirebaseDB1 .StoreValue
                tag  join  get global name
                           " /Hole Number 7 "
       valueToStore  get global textbox8
    call FirebaseDB1 .StoreValue
                tag  join  get global name
                           " /Hole Number 6 "
       valueToStore  get global textbox6
    call FirebaseDB1 .StoreValue
                tag  join  get global name
                           " /Hole Number 5 "
       valueToStore  get global textbox5
    call FirebaseDB1 .StoreValue
                tag  join  get global name
                           " /Hole Number 4 "
       valueToStore  get global textbox4
    call FirebaseDB1 .StoreValue
                tag  join  get global name
                           " /Hole Number 3 "
       valueToStore  get global textbox3
    call FirebaseDB1 .StoreValue
                tag  join  get global name
                           " /Hole Number 2 "
       valueToStore  get global textbox2
    call FirebaseDB1 .StoreValue
                tag  join  get global name
                           " /Hole Number 1 "
       valueToStore  get global textbox1

    call TinyDB1 .StoreValue
                tag  " total1 "
       valueToStore  get global textbox1 + get g...
    call TinyDB1 .StoreValue
                tag  " 1hole1 "
       valueToStore  get global textbox1
    call TinyDB1 .StoreValue
                tag  " 1hole2 "
       valueToStore  get global textbox2
    call TinyDB1 .StoreValue
                tag  " 1hole3 "
       valueToStore  get global textbox3
    call TinyDB1 .StoreValue
                tag  " 1hole4 "
       valueToStore  get global textbox4
    call TinyDB1 .StoreValue
                tag  " 1hole5 "
       valueToStore  get global textbox5
    call TinyDB1 .StoreValue
                tag  " 1hole6 "
       valueToStore  get global textbox6
    call TinyDB1 .StoreValue
                tag  " 1hole7 "
       valueToStore  get global textbox8
    call TinyDB1 .StoreValue
                tag  " 1hole8 "
       valueToStore  get global textbox9
    call TinyDB1 .StoreValue
                tag  " 1hole9 "
       valueToStore  get global textbox10
    set Label21 . Text to  get global textbox1 + get g...
```

Figure 5.13: Button Action Blocks:



```
when Screen3 .Initialize
do  set global name to      call TinyDB1 .GetValue
                                          tag  "Player1"
                            valueIfTagNotThere  "Not Available"
    set global total1 to    call TinyDB1 .GetValue
                                          tag  "total1"
                            valueIfTagNotThere  "0"
    set global textbox1 to  call TinyDB1 .GetValue
                                          tag  "1hole1"
                            valueIfTagNotThere  "0"
    set global textbox2 to  call TinyDB1 .GetValue
                                          tag  "1hole2"
                            valueIfTagNotThere  "0"
    set global textbox3 to  call TinyDB1 .GetValue
                                          tag  "1hole3"
                            valueIfTagNotThere  "0"
    set global textbox4 to  call TinyDB1 .GetValue
                                          tag  "1hole4"
                            valueIfTagNotThere  "0"
    set global textbox5 to  call TinyDB1 .GetValue
                                          tag  "1hole5"
                            valueIfTagNotThere  "0"
    set global textbox6 to  call TinyDB1 .GetValue
                                          tag  "1hole6"
                            valueIfTagNotThere  "0"
    set global textbox8 to  call TinyDB1 .GetValue
                                          tag  "1hole7"
                            valueIfTagNotThere  "0"
    set global textbox9 to  call TinyDB1 .GetValue
                                          tag  "1hole8"
                            valueIfTagNotThere  "0"
    set global textbox10 to call TinyDB1 .GetValue
                                          tag  "1hole9"
                            valueIfTagNotThere  "0"
    set Label19 . Text to get global name
    set Label21 . Text to get global total1
    set TextBox1 . Text to get global textbox1
    set TextBox2 . Text to get global textbox2
    set TextBox3 . Text to get global textbox3
    set TextBox4 . Text to get global textbox4
    set TextBox5 . Text to get global textbox5
    set TextBox6 . Text to get global textbox6
    set TextBox8 . Text to get global textbox8
    set TextBox9 . Text to get global textbox9
    set TextBox10 . Text to get global textbox10
```

Figure 5.14: Screen 3 Action Blocks

Blocks/Effects Case	Declaration Blocks	Firebase Blocks	TinyDB Blocks	Text Blocks	Action Blocks	Effects
Case 1	**Absent**	Present	Present	Present	Present	Application not able to declare variables.
Case 2	Present	**Absent**	Present	Present	Present	Application not able to communicate and send data to the online database.
Case 3	Present	Present	**Absent**	Present	Present	Application not able to store and call out the data stored in the application.
Case 4	Present	Present	Present	**Absent**	Present	Players' names unable to display in the application.
Case 5	Present	Present	Present	Present	**Absent**	The data cannot be sent to Firebase after the button was pressed.

Table 5.15: Effects for Absence of Specific Blocks for Screen 3

CHAPTER 6

CONCLUSION

6.0 Introduction

This chapter will discuss about the conclusion and findings for the IoT based scorecard for golf sport.

6.1 Conclusion

As mentioned in the chapter before, there are three objectives that needed to be achieved by completing this project. All the three objectives should be achieved by design and develop an IoT based golf scorecard. All the objectives were achieved since the development of the scorecard application and the online database are able to communicate with each other, send and receive data without error. The project had been completed according to the flow chart. The project is aimed to develop an IoT based scorecard for golf sport by featuring an Android based application and an online database.

Before the development of the scorecard, all the related previous project is been studied in order to improve the method used in the previous project and imply the advantage of the previous project in this current IoT based golf scorecard. Based on the findings from previous project, this project featured an Android based scorecard application and Google Firebase as the online database of this project. The Android based scorecard application will be the input device of this project and the data will be sent to the online database through the scorecard application. Other than that, the Firebase online database act as the online storage for all the data collected by the scorecard application in real-time manner. This project presents a IoT based scorecard that allow golfers to record, display and analyse their score obtained in a tournament wirelessly.

6.2 Recommendation

This project proposed to have improvements to increase its practicality and functionality in the future. As this project is first time developed and not an ideal system that can be apply on all golf courses or even other sports, the improvements are proposed as follows:

1. Change the online database to a more precise and suitable database such as MySQL Database Service.

   MySQL Database Services provide fully managed database service to deploy cloud-native applications.

   MySQL provides more comprehensive set of advanced features, management tools and technical support to achieve the highest levels of MySQL scalability, security, reliability, and uptime.
2. Develop the scorecard application by using Android Studio 2.0.

Android Studio 2.0 provides cloud test lab that enables developers to test the software through various devices and platform versions.

Android Studio 2.0 also provides app indexing which will allows users to conveniently locate the application online.

REFERENCE

Chatterjee, N., Chakraborty, S., Decosta, A., & Nath, A. (2018). Real-time Communication Application Based on Android Using Google Firebase. *International Journal of Advance Research in Computer Science and Management Studies*, 6(4), 74 – 79.

Khawas, C., & Shah, P. (2018). Application of firebase in android app development-a study. *International Journal of Computer Applications*, *179*(46), 49-53.

Laakso, J. K. (1994). GOLF SCORECARD. *U.S. Patent No. 5,284,340*. Washington, DC: U.S. Patent and Trademark Office.

Lambourne, G. T. (1996). ELECTRONIC GOLF SCORECARD. *U.S. Patent No. 5,536,010*. Washington, DC: U.S. Patent and Trademark Office.

Li, W. J., Yen, C., Lin, Y. S., Tung, S. C., & Huang, S. (2018, February). JustIoT Internet of Things based on the Firebase real-time database. In *2018 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE)* (pp. 43-47). IEEE.

Moroney, L. (2017). The firebase realtime database. In *The Definitive Guide to Firebase* (pp. 51-71). Apress, Berkeley, CA.

Ramadhan, A. F., & Suryana, T. LIVE SCORE APPLICATION DEVELOPMENT FOR FLAG FOOTBALL BASED ON ANDROID.

Remedio, J. W., & Appleton, M. R. (1990). GOLF SCORE RECODING SYSTEM AND NETWORK. *U.S. Patent No. 4,910,677*. Washington, DC: U.S. Patent and Trademark Office.

Sharma, D., & Dand, H. (2019). Firebase as BaaS for College Android Application. *International Journal of Computer Applications*, *178*(20), 1-6.

Stonehem, B. (2016). *Google Android Firebase: Learning the Basics* (Vol. 1). First Rank Publishing.

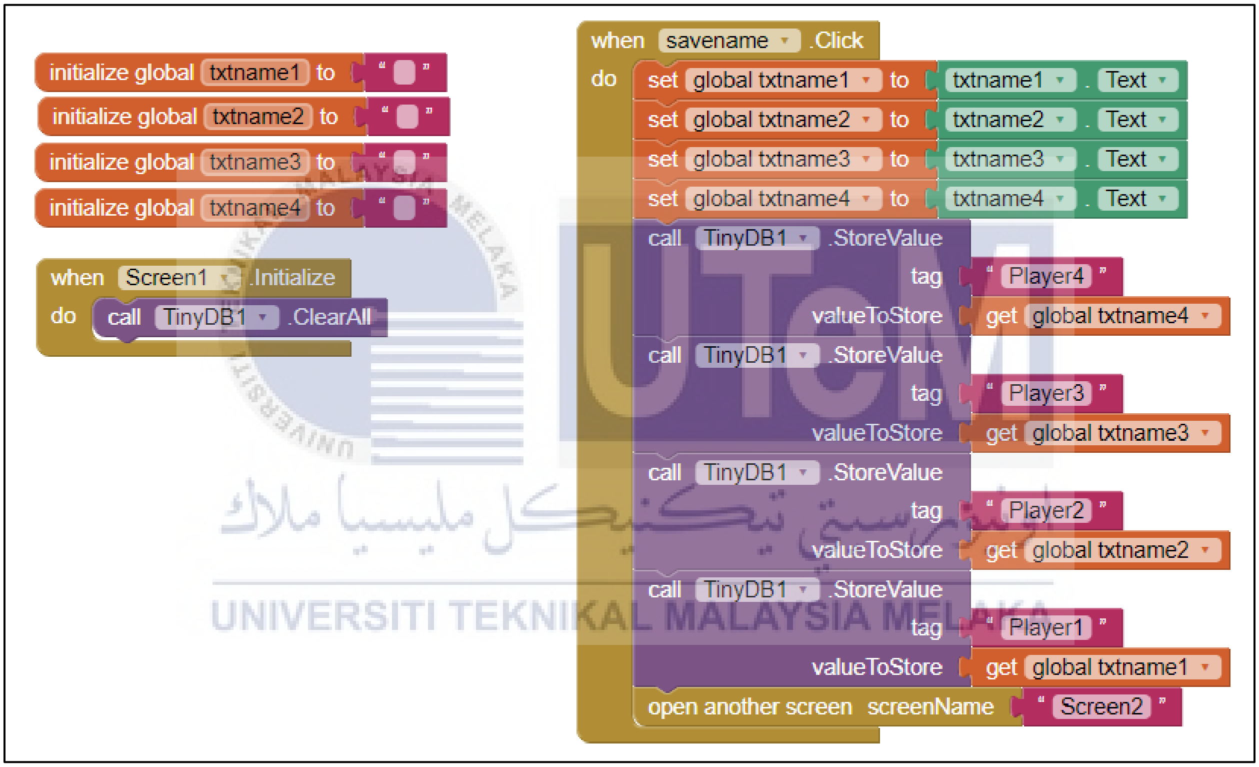
Sudiartha, I. K. G., Indrayana, I. N. E., Suasnawa, I. W., Asri, S. A., & Sunu, P. W. (2020, July). Data Structure Comparison Between MySql Relational Database and Firebase Database NoSql on Mobile Based Tourist Tracking Application. In *Journal of Physics: Conference Series* (Vol. 1569, No. 3, p. 032092). IOP Publishing.

Wiratno, A. R., & Hastuti, K. (2017). Implementation of Firebase Realtime Database to track BRT Trans Semarang. *Scientific Journal of Informatics*, *4*(2), 95-103.

APPENDIX

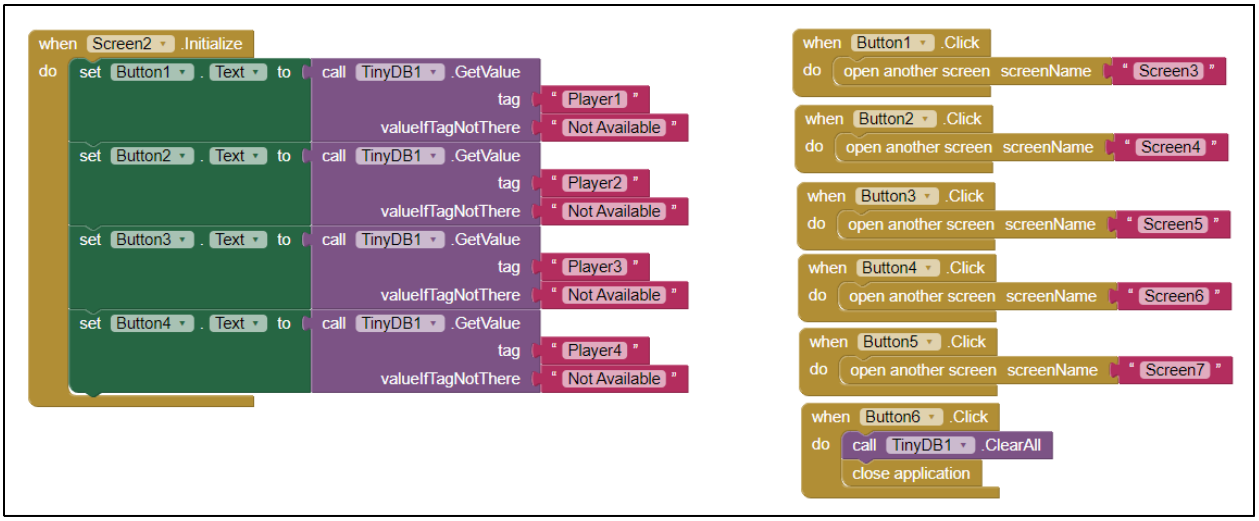
Appendix I

Block Program Language used in the development of the Scorecard Application.
Main Page



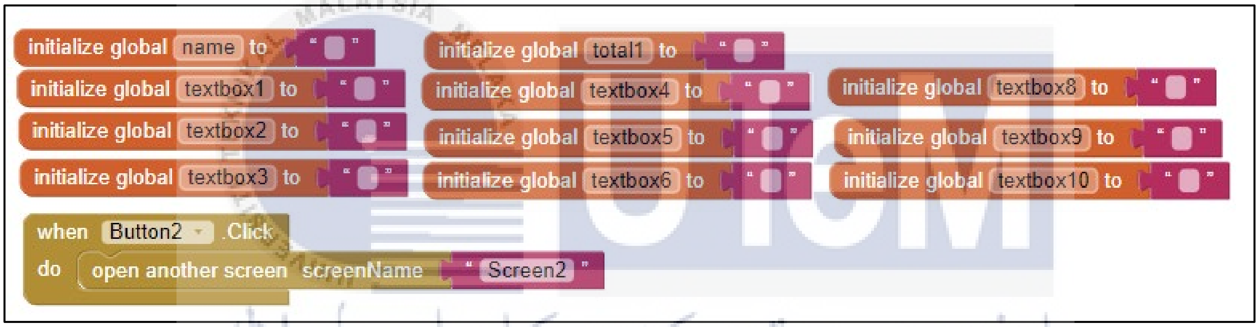
```
initialize global txtname1 to “ ”
initialize global txtname2 to “ ”
initialize global txtname3 to “ ”
initialize global txtname4 to “ ”

when Screen1 .Initialize
do  call TinyDB1 .ClearAll

when savename .Click
do  set global txtname1 to txtname1 . Text
    set global txtname2 to txtname2 . Text
    set global txtname3 to txtname3 . Text
    set global txtname4 to txtname4 . Text
    call TinyDB1 .StoreValue
                         tag  “ Player4 ”
                valueToStore  get global txtname4
    call TinyDB1 .StoreValue
                         tag  “ Player3 ”
                valueToStore  get global txtname3
    call TinyDB1 .StoreValue
                         tag  “ Player2 ”
                valueToStore  get global txtname2
    call TinyDB1 .StoreValue
                         tag  “ Player1 ”
                valueToStore  get global txtname1
    open another screen  screenName  “ Screen2 ”
```

Players Selection Page

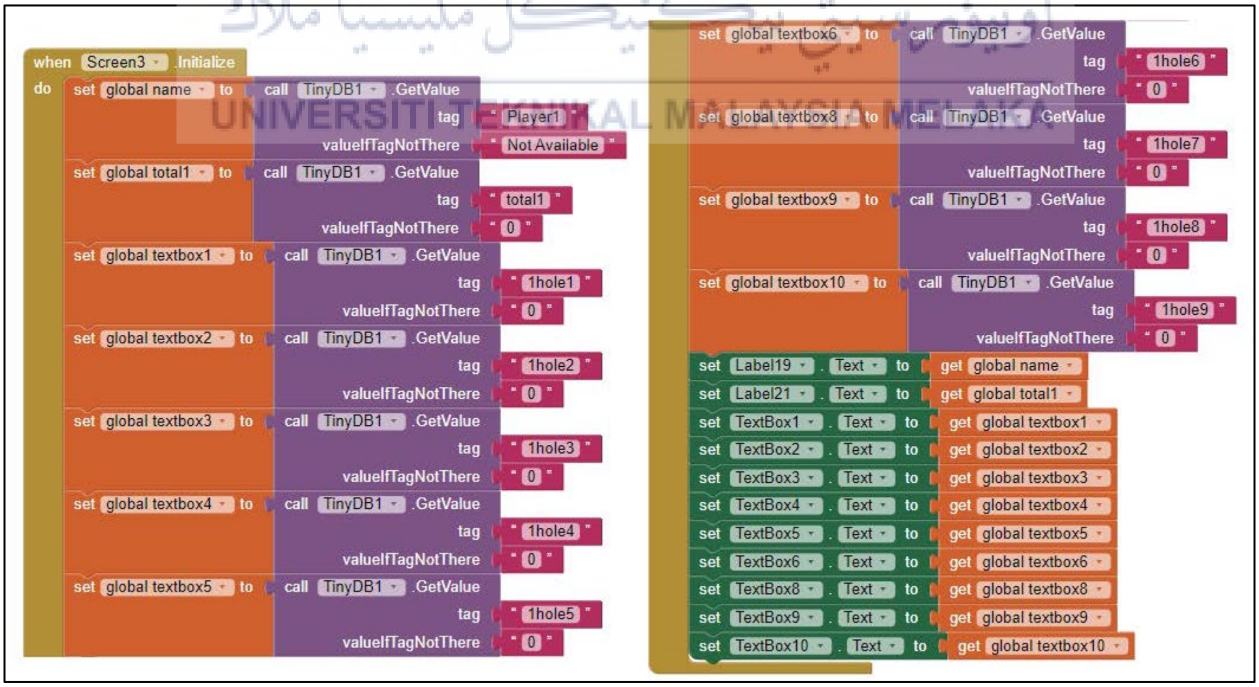


```
when Screen2 .Initialize
do  set Button1 . Text to  call TinyDB1 .GetValue
                                          tag  "Player1"
                            valueIfTagNotThere  "Not Available"
    set Button2 . Text to  call TinyDB1 .GetValue
                                          tag  "Player2"
                            valueIfTagNotThere  "Not Available"
    set Button3 . Text to  call TinyDB1 .GetValue
                                          tag  "Player3"
                            valueIfTagNotThere  "Not Available"
    set Button4 . Text to  call TinyDB1 .GetValue
                                          tag  "Player4"
                            valueIfTagNotThere  "Not Available"

when Button1 .Click
do  open another screen  screenName  "Screen3"

when Button2 .Click
do  open another screen  screenName  "Screen4"

when Button3 .Click
do  open another screen  screenName  "Screen5"

when Button4 .Click
do  open another screen  screenName  "Screen6"

when Button5 .Click
do  open another screen  screenName  "Screen7"

when Button6 .Click
do  call TinyDB1 .ClearAll
    close application
```

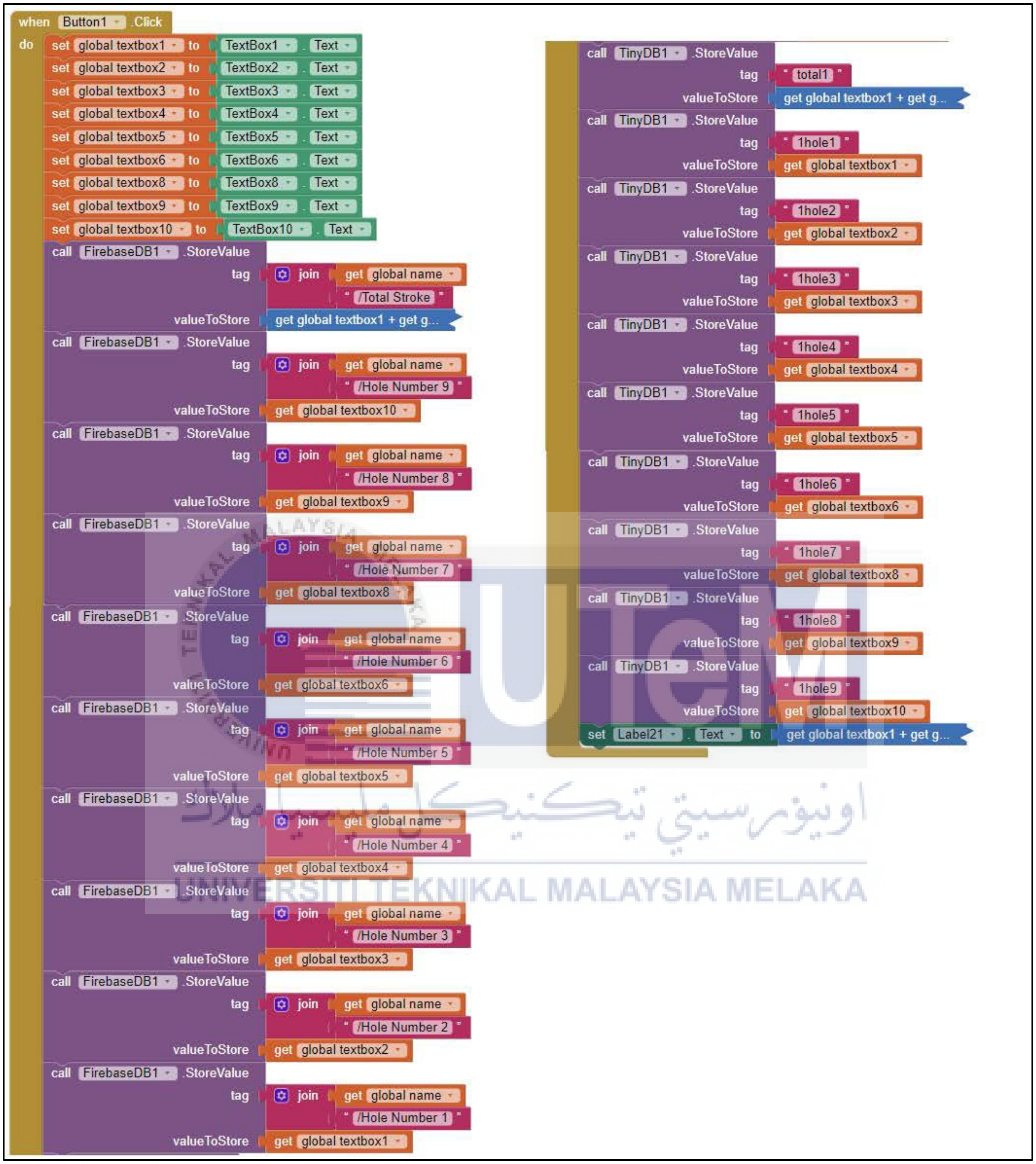
Player’s Score Recording Page



```
initialize global name to " "
initialize global textbox1 to " "
initialize global textbox2 to " "
initialize global textbox3 to " "
initialize global total1 to " "
initialize global textbox4 to " "
initialize global textbox5 to " "
initialize global textbox6 to " "
initialize global textbox8 to " "
initialize global textbox9 to " "
initialize global textbox10 to " "

when Button2 .Click
do  open another screen  screenName  "Screen2"
```

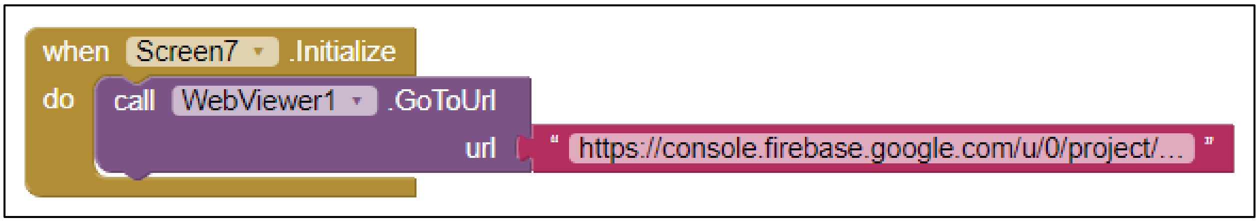


```
when Screen3 .Initialize
do  set global name to  call TinyDB1 .GetValue
                                       tag  "Player1"
                         valueIfTagNotThere  "Not Available"
    set global total1 to  call TinyDB1 .GetValue
                                       tag  "total1"
                         valueIfTagNotThere  "0"
    set global textbox1 to  call TinyDB1 .GetValue
                                       tag  "1hole1"
                         valueIfTagNotThere  "0"
    set global textbox2 to  call TinyDB1 .GetValue
                                       tag  "1hole2"
                         valueIfTagNotThere  "0"
    set global textbox3 to  call TinyDB1 .GetValue
                                       tag  "1hole3"
                         valueIfTagNotThere  "0"
    set global textbox4 to  call TinyDB1 .GetValue
                                       tag  "1hole4"
                         valueIfTagNotThere  "0"
    set global textbox5 to  call TinyDB1 .GetValue
                                       tag  "1hole5"
                         valueIfTagNotThere  "0"
    set global textbox6 to  call TinyDB1 .GetValue
                                       tag  "1hole6"
                         valueIfTagNotThere  "0"
    set global textbox8 to  call TinyDB1 .GetValue
                                       tag  "1hole7"
                         valueIfTagNotThere  "0"
    set global textbox9 to  call TinyDB1 .GetValue
                                       tag  "1hole8"
                         valueIfTagNotThere  "0"
    set global textbox10 to  call TinyDB1 .GetValue
                                       tag  "1hole9"
                         valueIfTagNotThere  "0"
    set Label19 . Text to  get global name
    set Label21 . Text to  get global total1
    set TextBox1 . Text to  get global textbox1
    set TextBox2 . Text to  get global textbox2
    set TextBox3 . Text to  get global textbox3
    set TextBox4 . Text to  get global textbox4
    set TextBox5 . Text to  get global textbox5
    set TextBox6 . Text to  get global textbox6
    set TextBox8 . Text to  get global textbox8
    set TextBox9 . Text to  get global textbox9
    set TextBox10 . Text to  get global textbox10
```



```
when Button1 .Click
do  set global textbox1 to TextBox1 . Text
    set global textbox2 to TextBox2 . Text
    set global textbox3 to TextBox3 . Text
    set global textbox4 to TextBox4 . Text
    set global textbox5 to TextBox5 . Text
    set global textbox6 to TextBox6 . Text
    set global textbox8 to TextBox8 . Text
    set global textbox9 to TextBox9 . Text
    set global textbox10 to TextBox10 . Text
    call FirebaseDB1 .StoreValue
                       tag  join  get global name
                                  " /Total Stroke "
              valueToStore  get global textbox1 + get g...
    call FirebaseDB1 .StoreValue
                       tag  join  get global name
                                  " /Hole Number 9 "
              valueToStore  get global textbox10
    call FirebaseDB1 .StoreValue
                       tag  join  get global name
                                  " /Hole Number 8 "
              valueToStore  get global textbox9
    call FirebaseDB1 .StoreValue
                       tag  join  get global name
                                  " /Hole Number 7 "
              valueToStore  get global textbox8
    call FirebaseDB1 .StoreValue
                       tag  join  get global name
                                  " /Hole Number 6 "
              valueToStore  get global textbox6
    call FirebaseDB1 .StoreValue
                       tag  join  get global name
                                  " /Hole Number 5 "
              valueToStore  get global textbox5
    call FirebaseDB1 .StoreValue
                       tag  join  get global name
                                  " /Hole Number 4 "
              valueToStore  get global textbox4
    call FirebaseDB1 .StoreValue
                       tag  join  get global name
                                  " /Hole Number 3 "
              valueToStore  get global textbox3
    call FirebaseDB1 .StoreValue
                       tag  join  get global name
                                  " /Hole Number 2 "
              valueToStore  get global textbox2
    call FirebaseDB1 .StoreValue
                       tag  join  get global name
                                  " /Hole Number 1 "
              valueToStore  get global textbox1

    call TinyDB1 .StoreValue
                       tag  " total1 "
              valueToStore  get global textbox1 + get g...
    call TinyDB1 .StoreValue
                       tag  " 1hole1 "
              valueToStore  get global textbox1
    call TinyDB1 .StoreValue
                       tag  " 1hole2 "
              valueToStore  get global textbox2
    call TinyDB1 .StoreValue
                       tag  " 1hole3 "
              valueToStore  get global textbox3
    call TinyDB1 .StoreValue
                       tag  " 1hole4 "
              valueToStore  get global textbox4
    call TinyDB1 .StoreValue
                       tag  " 1hole5 "
              valueToStore  get global textbox5
    call TinyDB1 .StoreValue
                       tag  " 1hole6 "
              valueToStore  get global textbox6
    call TinyDB1 .StoreValue
                       tag  " 1hole7 "
              valueToStore  get global textbox8
    call TinyDB1 .StoreValue
                       tag  " 1hole8 "
              valueToStore  get global textbox9
    call TinyDB1 .StoreValue
                       tag  " 1hole9 "
              valueToStore  get global textbox10
    set Label21 . Text to  get global textbox1 + get g...
```

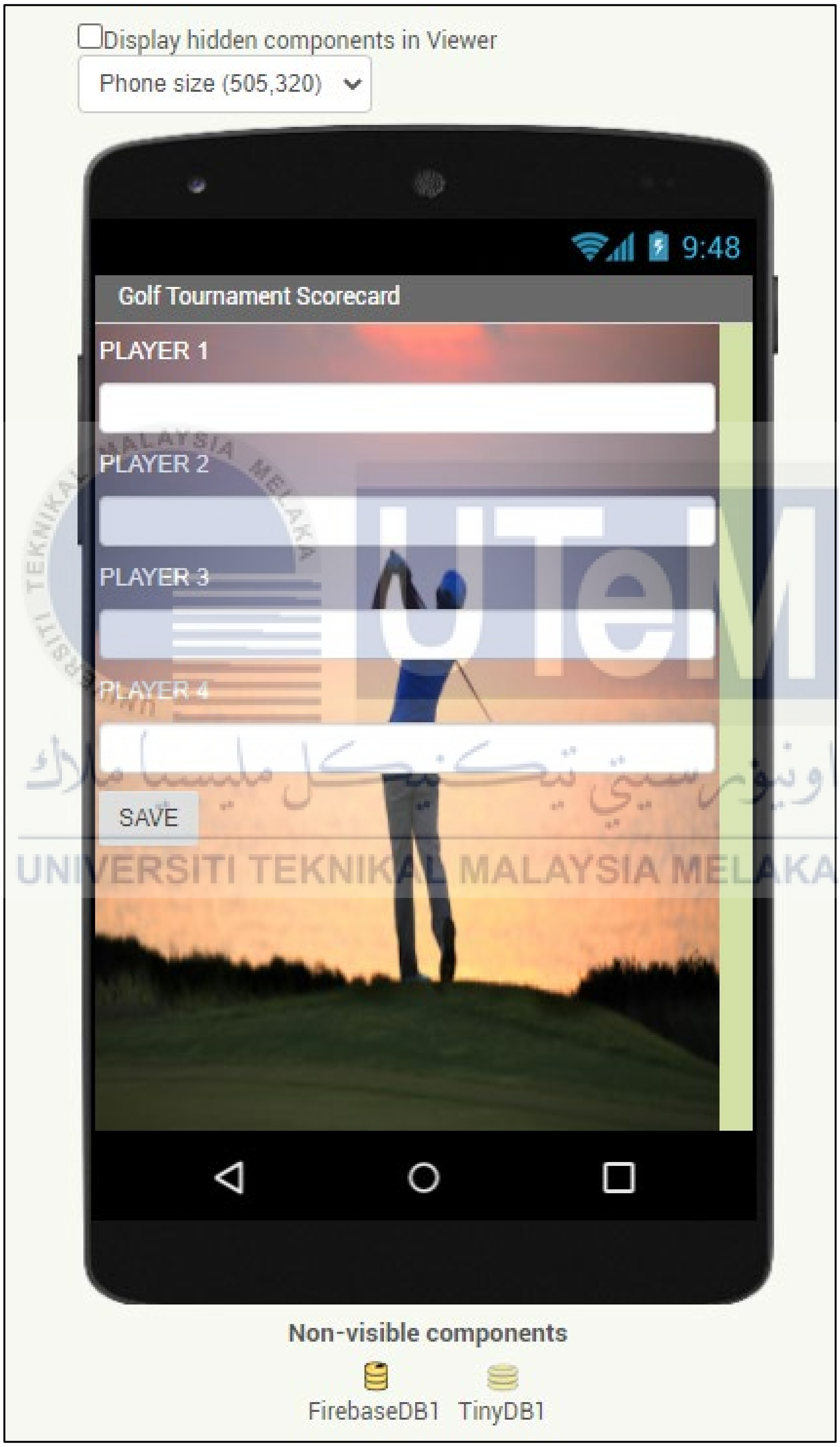
Overall Score Page



```
when Screen7 .Initialize
do  call WebViewer1 .GoToUrl
                        url  " https://console.firebase.google.com/u/0/project/... "
```

Appendix II

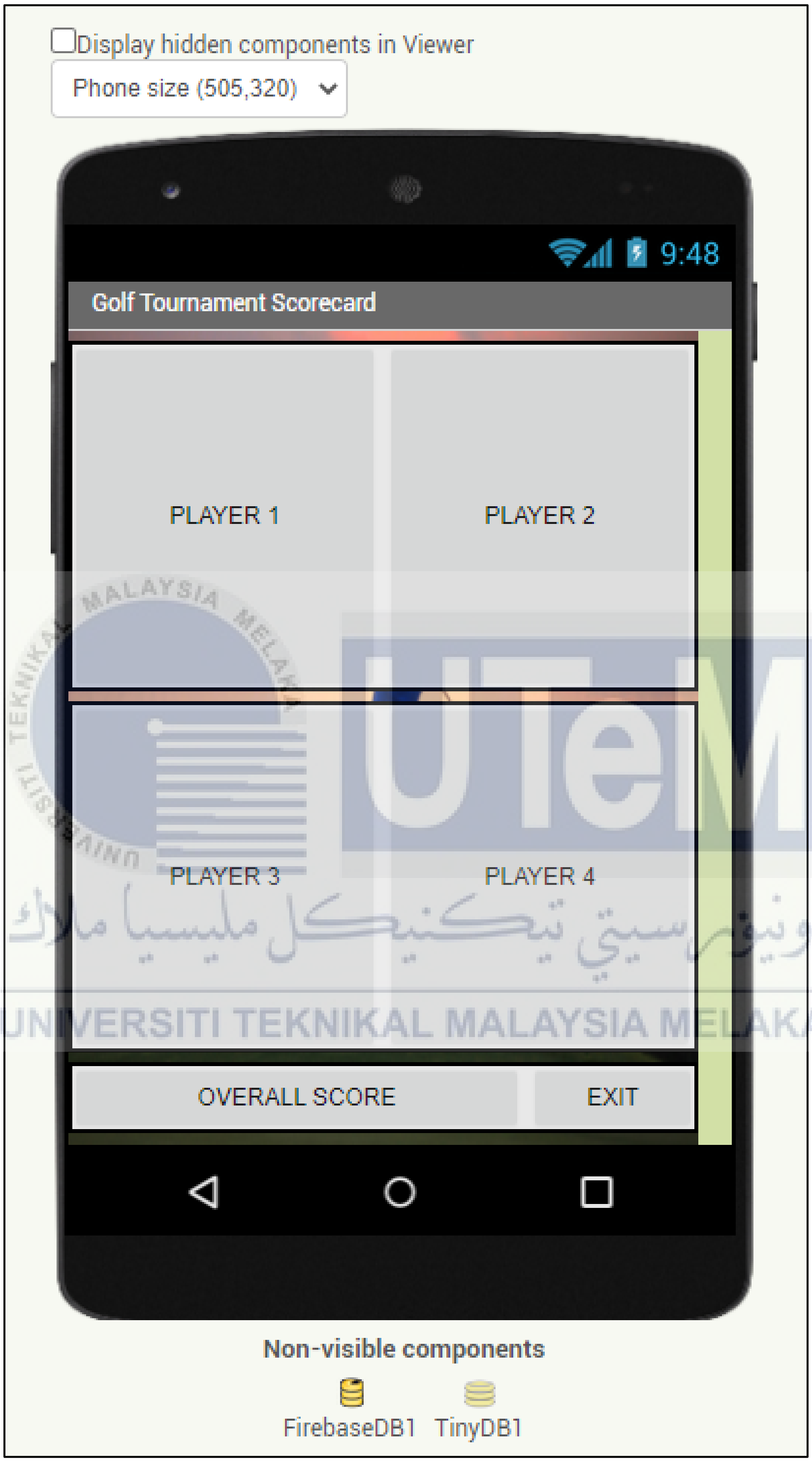
MIT App Inventor II Project Layout
Main Page



☐ Display hidden components in Viewer

Phone size (505,320)

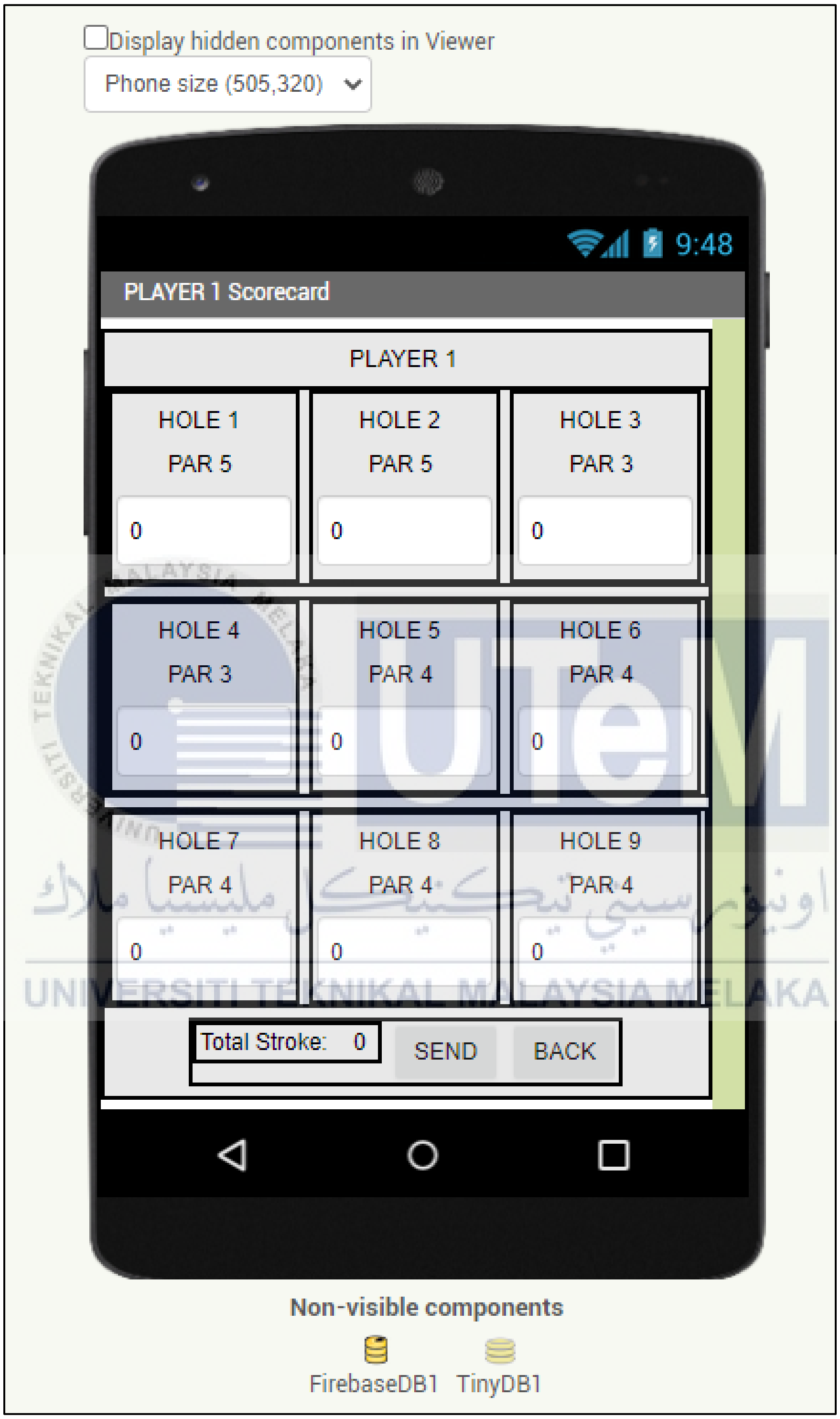
Players Selection Page



☐Display hidden components in Viewer

Phone size (505,320)

Player’s Score Recording Page



□Display hidden components in Viewer

Phone size (505,320)

9:48

PLAYER 1 Scorecard

PLAYER 1

HOLE 1 PAR 5 0	HOLE 2 PAR 5 0	HOLE 3 PAR 3 0
HOLE 4 PAR 3 0	HOLE 5 PAR 4 0	HOLE 6 PAR 4 0
HOLE 7 PAR 4 0	HOLE 8 PAR 4 0	HOLE 9 PAR 4 0

Total Stroke: 0

Non-visible components

FirebaseDB1 TinyDB1