Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name: Mark Leonard
Student ID: x14562027
Supervisor: Dominic Carr

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: *Mark Leonard* Date: 13/05/2018

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

Complete the sections above and attach it to the front of one of the copies of your assignment,

What constitutes plagiarism or cheating?

The following is extracted from the college's formal statement on plagiarism as quoted in the Student Handbooks. References to "assignments" should be taken to include any piece of work submitted for assessment.

Paraphrasing refers to taking the ideas, words or work of another, putting it into your own words and crediting the source. This is acceptable academic practice provided you ensure that credit is given to the author. Plagiarism refers to copying the ideas and work of another and misrepresenting it as your own. This is completely unacceptable and is prohibited in all academic institutions. It is a serious offence and may result in a fail grade and/or disciplinary action. All sources that you use in your writing must be acknowledged and included in the reference or bibliography section. If a particular piece of writing proves difficult to paraphrase, or you want to include it in its original form, it must be enclosed in quotation marks

and credit given to the author.

When referring to the work of another author within the text of your project you must give the author's surname and the date the work was published. Full details for each source must then be given in the bibliography at the end of the project

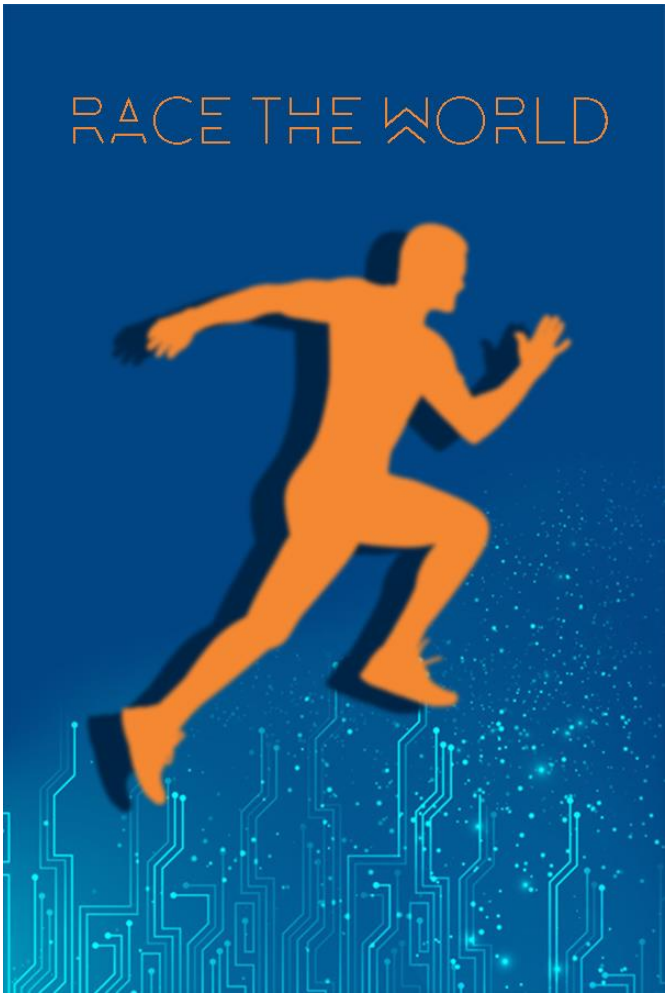
Penalties for Plagiarism

If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the college's Disciplinary Committee. Where the Disciplinary Committee makes a finding that there has been plagiarism, the Disciplinary Committee may recommend

- that a student's marks shall be reduced
- that the student be deemed not to have passed the assignment
- that other forms of assessment undertaken in that academic year by the same student be declared void
- that other examinations sat by the same student at the same sitting be declared void

Further penalties are also possible including

- suspending a student college for a specified time,
- expelling a student from college,
- prohibiting a student from sitting any examination or assessment.,
- the imposition of a fine and
- the requirement that a student to attend additional or other lectures or courses or undertake additional academic work.



PROJECT REPORT

RACE THE WORLD ANDROID APPLICATION

Mark Leonard
x14562027@student.ncirl.ie

Contents

Executive Summary 5
Introduction 6
Background 6
Aims 6
Technologies 6
System 7
Functional Requirements 8
Use Case Diagram 8
Non-Functional Requirements 14
Performance/Response time requirement 14
Availability requirement 14
Robustness requirement 14
Security requirement 14
Reliability requirement 14
Recovery requirement 14
Data Requirements 15
Environmental Requirements 15
Design and Architecture 15
GUI 15
Main Page 16
Log-in Page 17
User Profile Page 18
Menu Page 19
Search Users Page 20
Challenge (Waiting on Opponent) Page 21
Run/Race Page 23
Race Finish/Stop Page 24
System Architecture 25
Flow Diagram 26
Analysis and Design 26
Architecture Design 26
Logical View 26
Hardware Architecture 27
Software Architecture 27
Security Architecture 27

Implementation 27
System Design 28
Testing 33
Conclusion 35
Further Development 35
Bibliography 36
Appendix 37
User Manuel 37
Project Proposal 40
Objectives 41
Background 43
Technical Approach 43
Special resources required 45
Project Plan 45
Technical Details 45
Evaluation 47
Monthly Journals 48
September 2017 48
October 2017 48
November 2017 49
December 2017 50
January 2018 51
February 2018 51
March 2018 51
April 2018 52
May 2018 53

Executive Summary

The purpose of this document is to provide the technical details and requirements for the Race the World Android application. Like many other fitness apps available on the Android market, Race the World allows users to track progress of their runs and receive statistics on their performance both during and after the run. To motivate runners, this application has introduced gamification in the form of a racing feature. The feature allows users to race against either friends or random opponents in real-time which provides runners with a competitive outlet when local races are not available. The idea is that the user can simply log into the application and choose to challenge another user to a race, and if the other users responds a race will begin.

The goal of the application is to give users of fitness applications an alternative compared to the usual statistics driven applications that most people use today. The gamification is the fundamental part of the application.

The application was developed using Android Studio with Java programming language and using Google Firebase to store, retrieve and process user's data, making use of the 'Realtime Database' and 'Authentication' provided by Firebase.

Introduction

Background

The primary objective of this document is to clearly communicate the expected functionality and requirements for the Race the World software application to both the developer and lectures who will be reviewing it. Specifically, it will clearly outline the requirements of the software in terms of functionality, interfaces and performance. Running and fitness apps have become increasingly popular in the last decade as they provide users with an easy way to plan and track fitness goals. Many of these apps try and encourage their users to be active through various methods such as reaching certain goals daily or simply sending push notification reminders on a regular basis.

Race the World aims to add an element of fun and competitiveness that very few of these applications offer. It aims to improve the user's overall wellbeing and fitness levels. Numerous websites and fitness blogs encourage people to use fitness applications and regular reasons they give include things like they "help track your progress", "save time and efficiency" and "boost your motivation". Race the World tries to implement all these features in the application.

Aims

One aspect of encouragement that has rarely been explored is the ability for users to not only compete against themselves, but against other runners in real time. The aim of this application is to allow users to match up with people from anywhere in the world and race in real time. As well as showing race statistics, the app will also allow the user to see other aspects of their runs, such as total distance and total time spent competing. The application aims to be user-friendly to encourage the competitors to use it for its purpose.

Technologies

For the development of this application the technologies being used are Android Studio IDE working with Google Firebase which will be connected to the application within the integrated Firebase features in Android Studio. Android Studio is the official integrated development environment (IDE) for Google's Android operating system, it is developed by JetBrains, the company behind IntelliJ IDEA software and designed specifically for Android development.

The main elements of Firebase being used for this application are Firebase real-time database and authentication. Conveniently, Android Studio provides most of the tools required for the development of the application.

System

In this section, the report discusses both the functional and non-functional requirements of the application respectively.

On the current android market there are endless fitness and running applications, however very few if any have a gamification to them. This is something that many users want. Part of human nature is being competitive, and with the application, competitiveness and fitness are brought in to one.

Functional Requirements

This section looks at the functional requirements for the application.

(Not all these requirements were implemented in the final project due to various reasons discussed in this document, mainly due to working a full time job and poor time management on my behalf).

Use Case Diagram

Requirement 1: User Sign In

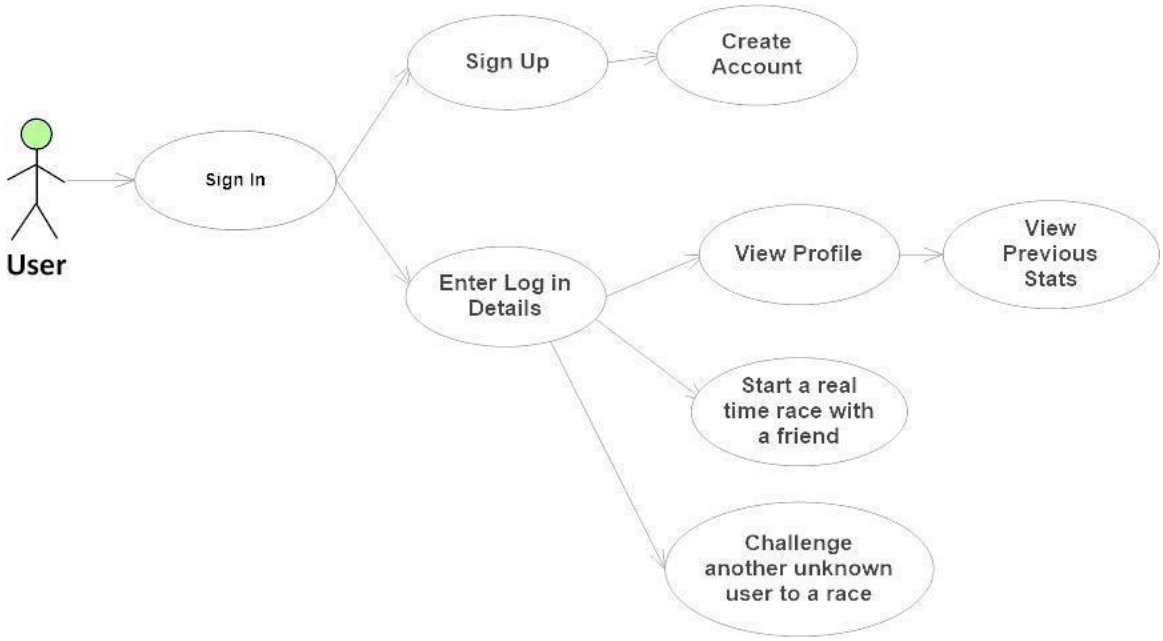
Description & Priority

This use case describes the role of the user in logging in to use the android application.

Scope

The scope of this use case is for the user to sign into the Android application.

Use Case Diagram



Flow Description

Precondition

The system is idle.

Activation

This use case starts when the user clicks ‘Sign in’.

Main flow

1. The user is presented with the options to ‘Google Sign In’
2. The user is prompted to select their chosen Google account.
3. The user can view their profile, race against friends, or challenge other random users to a race.
4. The system will display the user’s stats.
5. The user confirms any actions they have carried out.
6. The system will return the user to their homepage or proceed to begin a race.

Exceptional flow

1. The mobile device freezes or crashes, making it impossible for the user to sign in or register for the application.
2. The battery run out on the mobile device, stopping the user from signing up.

Termination

1. The user logs out.

Post condition

1. The system will return to its default homepage.

Requirement 2: Administration

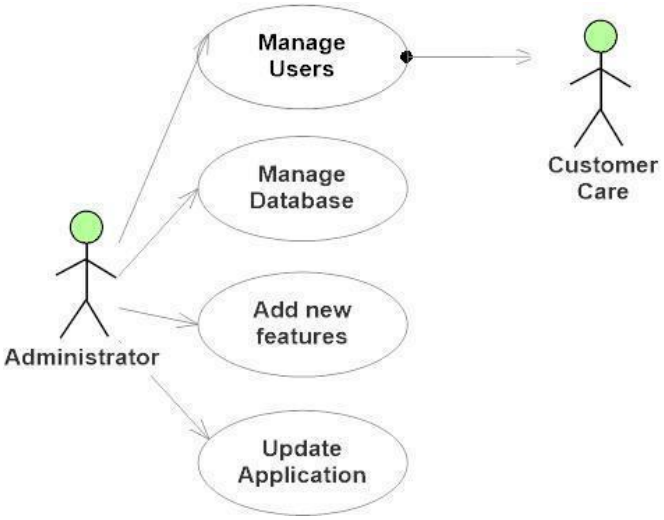
Description & Priority

This use case describes the role of the administrator in managing users and troubleshooting any issues raised by users

Scope

The scope of this use case is for the administrator to manage users, the database and keep the application up to date.

Use Case Diagram



Flow Description

Precondition

The system is idle.

Activation

This use case starts when the administrator logs in.

Main flow

1. The administrator views user issues.
2. The administrator passes these issues to the customer care team.
3. The administrator views issues raised.
4. The system will display the issues that need to be solved by the customer care team.
5. The customer care solves issues as required.
6. The system will display the changes that were made by the administrator.

Exceptional flow

1. The mobile device freezes or crashes, making it impossible for the user to sign in or register for the application.
2. The battery run out on the mobile device, stopping the user from signing up.

Termination

1. The administrator logs out.

Post condition

1. The system will return to its default homepage.

Requirement 3: Performing a run

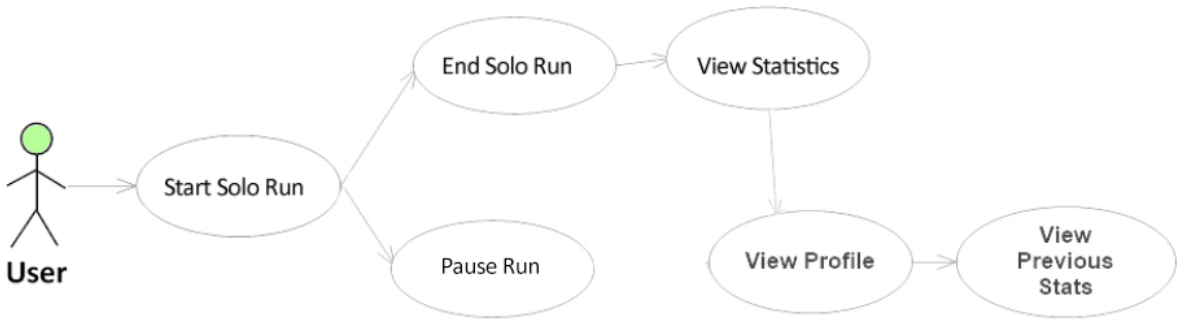
Description & Priority

This use case describes the user running and tracking their own personal run.

Scope

The scope of this use case is to show how the user goes for a run and the application tracks them while they are running.

Use Case Diagram



Flow Description

Precondition

The user is logged into their profile.

Activation

This use case starts when the user chooses to begin a run.

Main flow

1. The user chooses to begin a solo run.
2. The system counts the user into their run.

3. The user begins to run.
4. The system tracks the users distance, location, calories burned etc.
5. The system informs the user of their progress in real time.
6. The user ends their run.
7. The system displays the user's statistics.

Exceptional flow

1. The mobile device freezes or crashes, making it impossible for the user to begin their run.
2. The battery run out on the mobile device, stopping the system from tracking the user.

Termination

1. The user ends their run and closes the application.

Post condition

1. The system will return to the user's profile.

Requirement 4: Racing opponent

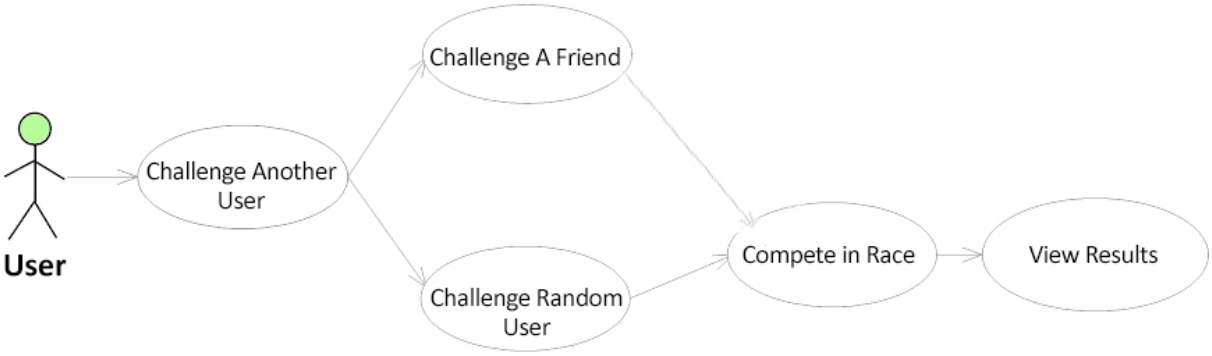
Description & Priority

This use case describes the user racing another user.

Scope

The scope of this use case is to show how the user challenges a friend or random opponent to a race.

Use Case Diagram



Flow Description

Precondition

The user is logged into their profile.

Activation

This use case starts when the user chooses to challenge another user.

Main flow

1. The user chooses to challenge another user.
2. The user chooses to race a friend or random opponent.
3. The user selects the distance of the race.
4. The system counts the runners into their race.
5. The users begin racing each other.
6. The system tracks both users and informs them of their position in the race
7. The user or the opponent reaches the set distance.
8. The system informs the users which of them has won the race.

Exceptional flow

1. The mobile device freezes or crashes, making it impossible for one of the users to compete in the race.
2. The battery run out on the mobile device, stopping the system from tracking the users.

Termination

1. The user or their opponent wins the race#

Post condition

The system will return to the users' profile.

Non-Functional Requirements

Performance/Response time requirement

Users of the application will vary greatly in fitness levels so that will influence the performance/response time as different people will run a specific distance in a varying number of minutes.

Availability requirement

The application should operate at an ongoing basis and should only be made unavailable for maintenance or updates at no more than once a month for a limited time frame.

Robustness requirement

The main functionality of the application is tracking a runner in real-time, therefore it must be robust to stay connected as it is constantly in motion. If the real-time tracking were to fail it would result in the application not performing.

Security requirement

The application must be secure as every user has personal information stored, as well as tracking. This must be secured from any attacks from external sources to prevent any breach of user data. This is the main reason for implementing Firebase's authentication and limiting the application to Google account users only, as the application is an Android application every user would already have a Google account already active on their device.

Reliability requirement

The application should be reliability always. If two users are racing each other in real-time, it must be able to handle to demand for keeping two users up to date on their progress in the race as well as tracking them both live.

Recovery requirement

Shall the application have any sort of system failure the system should be recovered as soon as possible to a fully working condition. While informing users of the expected recovery time.

Data Requirements

The main data requirements for the application are a

- Google Account – which all android device users should already have active, this is for log in purposes, and the users name and email address are stored from this.
- User location – When launching the application for the first time the user is prompted to allow the application to track their location. As this is the key feature of the application it is a requirement.
- User activity records – In order to track the users total distance and time it is a requirement to have access to this data, which is securely stored on the Firebase database.

Environmental Requirements

This section of the report covers the environmental requirements which is the various components that are essential to develop the application.

- Computer: The application is being developed on a Windows computer running Android Studio, so this is a requirement.
- Internet access: Internet access is a requirement for connecting for testing and connecting to Firebase.
- Android device: An android device with the appropriate specifications to run the application is required for testing and running the application during development.
- Photoshop: Photoshop is required for customising the overall styling of the application as opposed to using the Android Studio default themes.

Design and Architecture

GUI

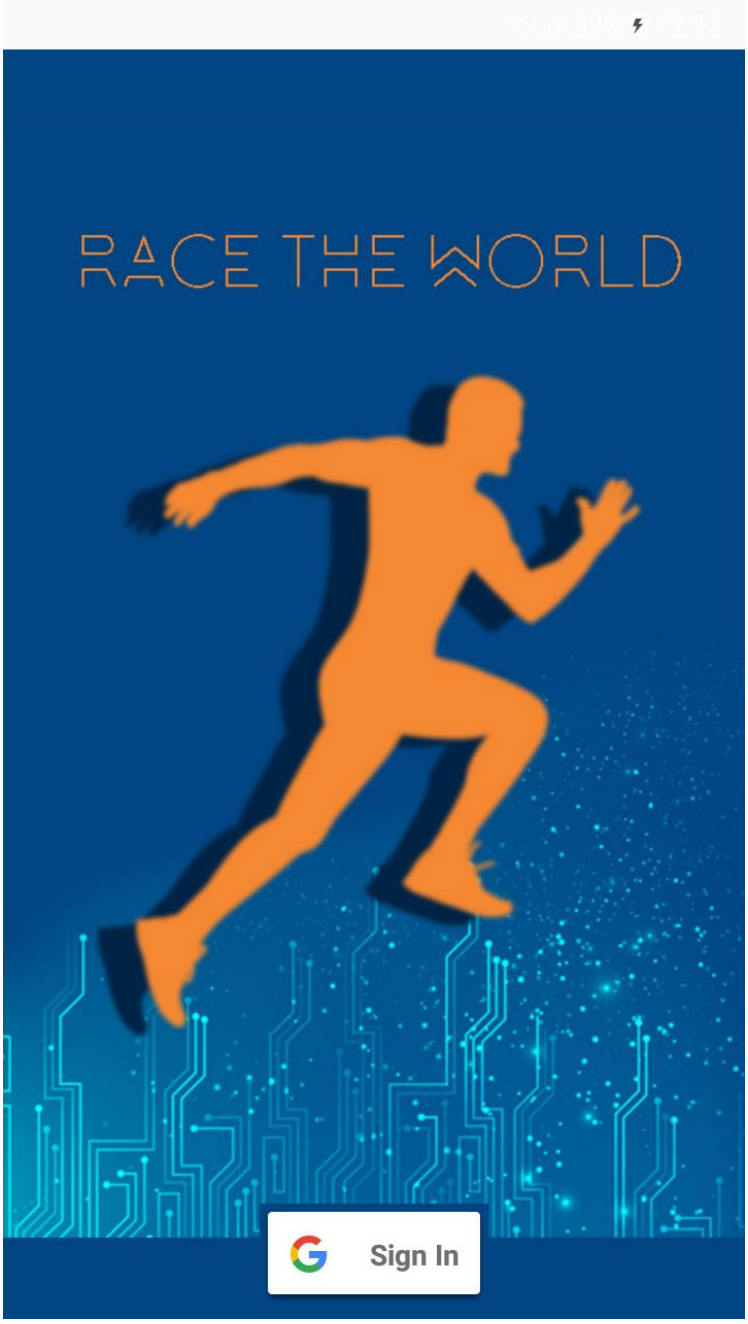
Race the World application is made up of the following graphical user interfaces:

- Main Page
- Login Page
- User Profile Page
- Search Users Page
- Challenge Users Page
- Run Page

Main Page

On the main page when the user launches the application or logs out, the user is presented with a permission request to allow the application to track their location (this occurs only the first time the application is installed), then they are shown a splash screen showing a graphic designed for the application as well as a google sign in button. The sign in button is their only option when choosing to use the application.

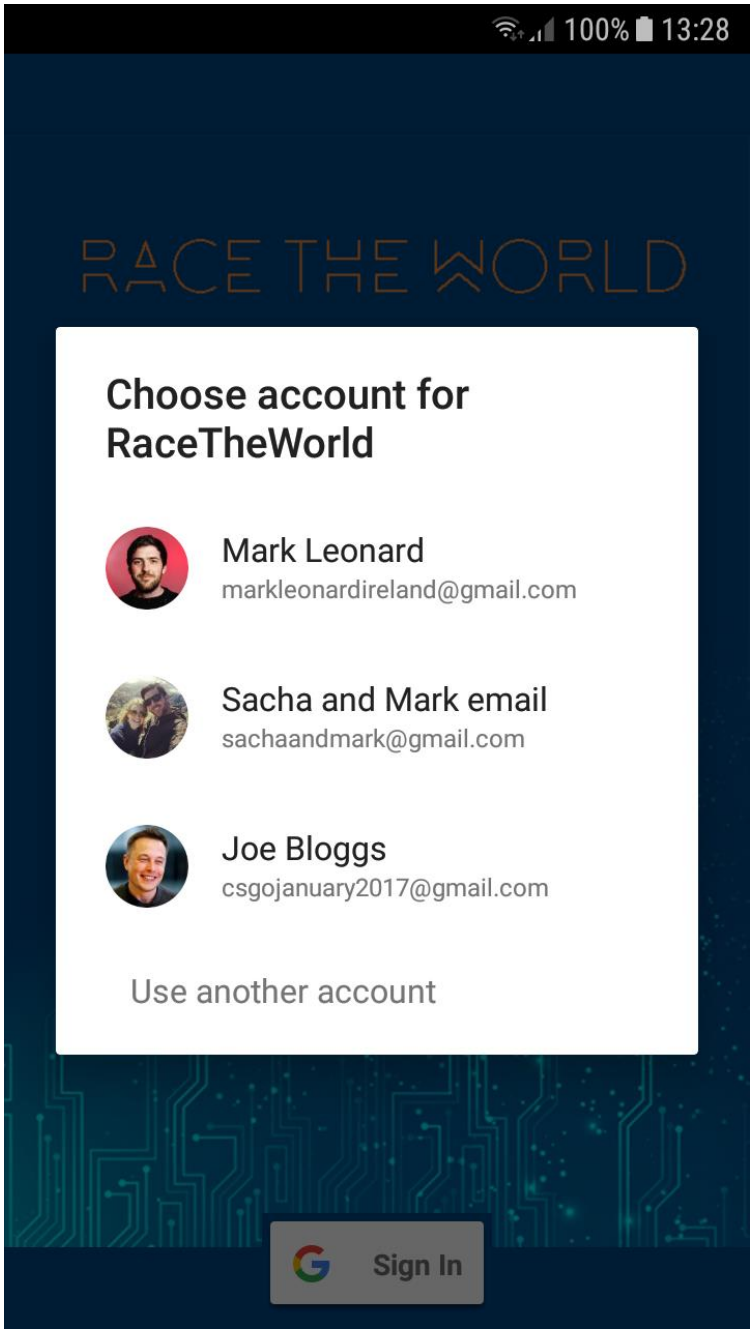
Main Page:



Log-in Page

The log-in page will be presented to the user when they choose the google sign-in button from the main page. A pop up will make the user choose which Google account they want to use if they are signed into multiple accounts at one time on their device.

Log-In Page:



User Profile Page

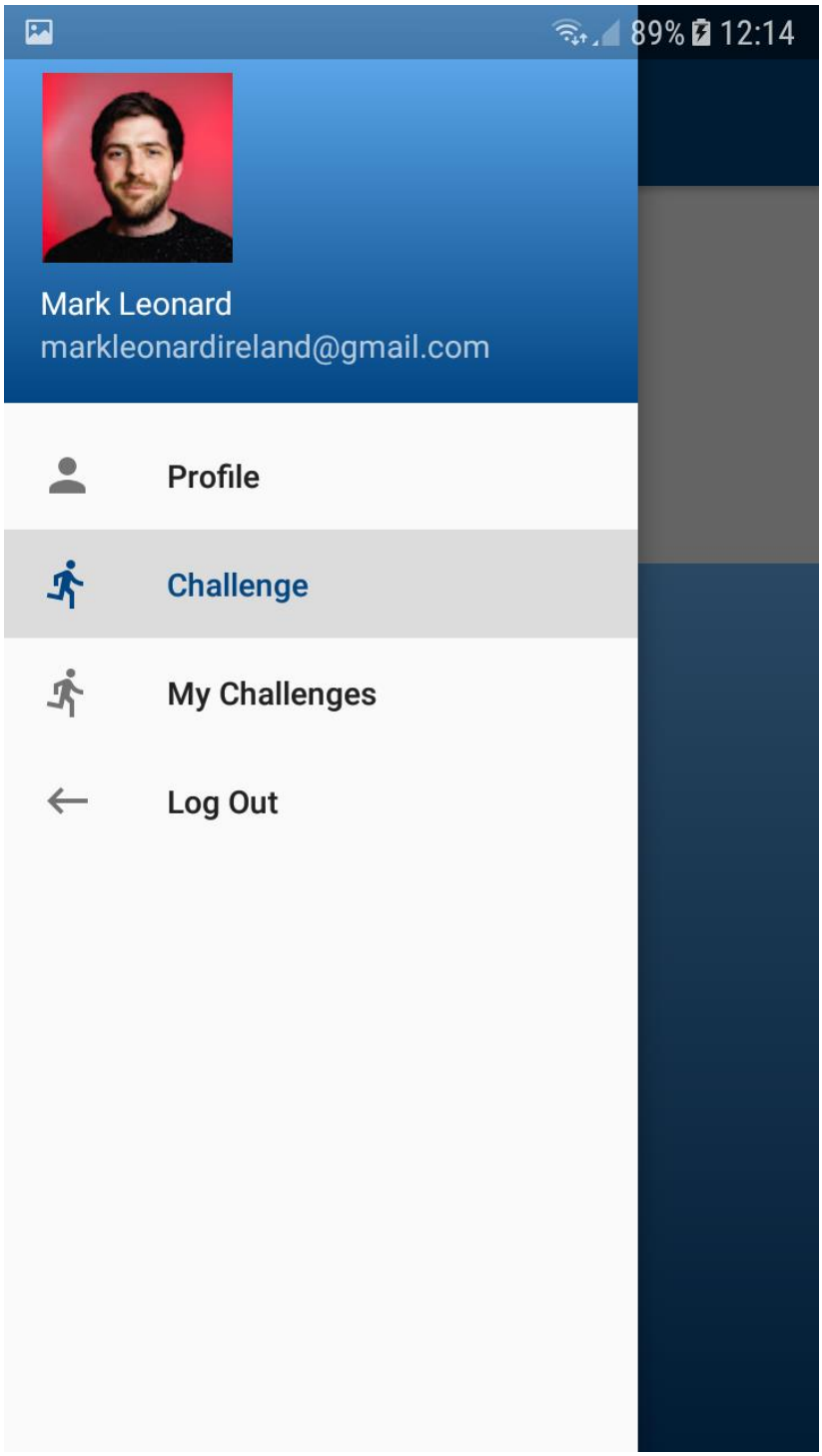
The user profile page shows the signed in user their Google profile image as well as the total wins they have in the application races and their total distance.

User Profile Page:

Menu Page

The main menu appears as a slide in navigation view from swiping in from the left side of the user's device. The header of the menu shows the current signed in user profile image as well as their name and email address.

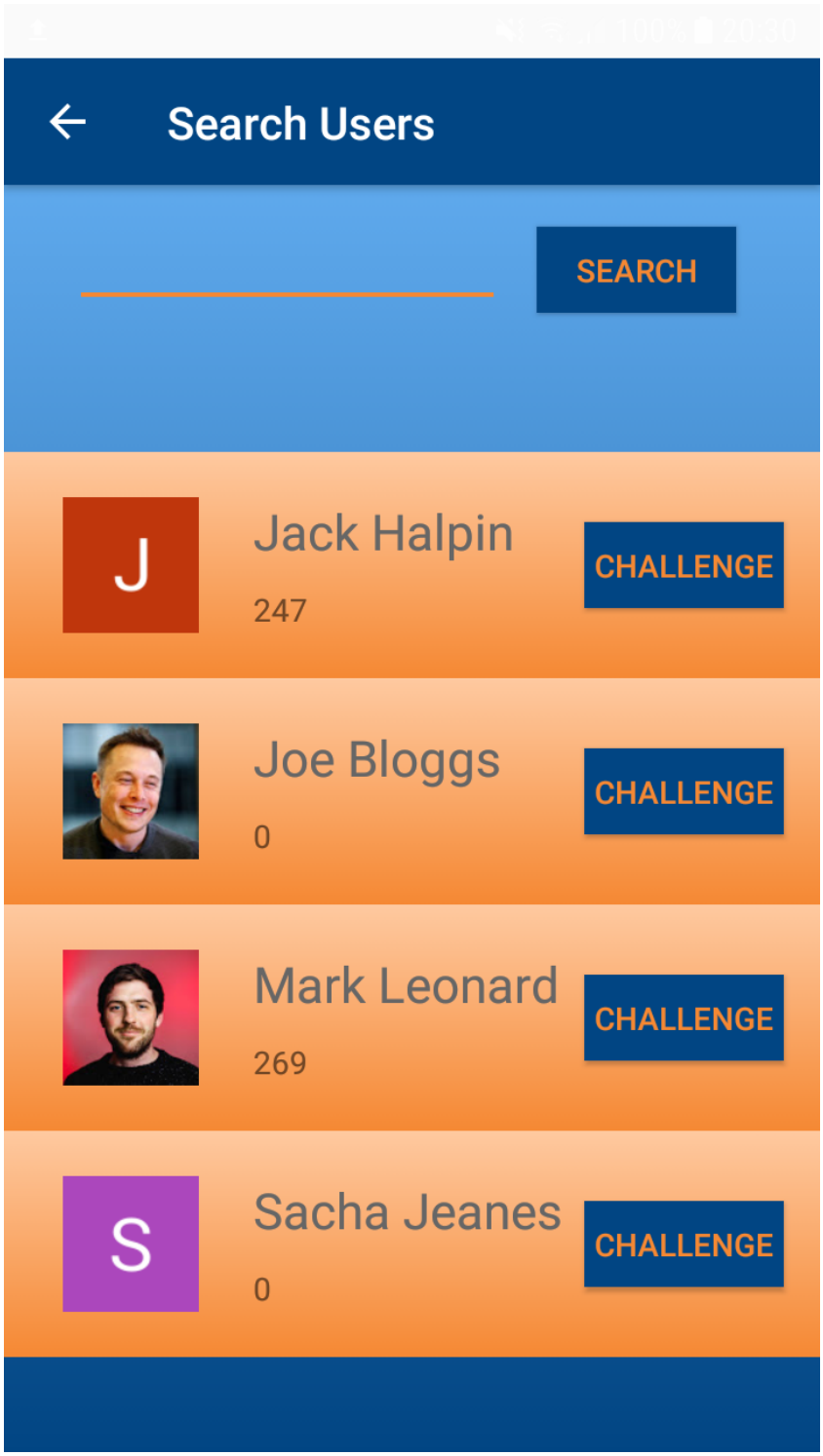
Menu Page:



Search Users Page

The search users page displays a search bar and button to the user, they can search for a specific opponent here and when the desired opponent is found they can click the ‘running person’ icon beside the opponent’s name to challenge that user to a race.

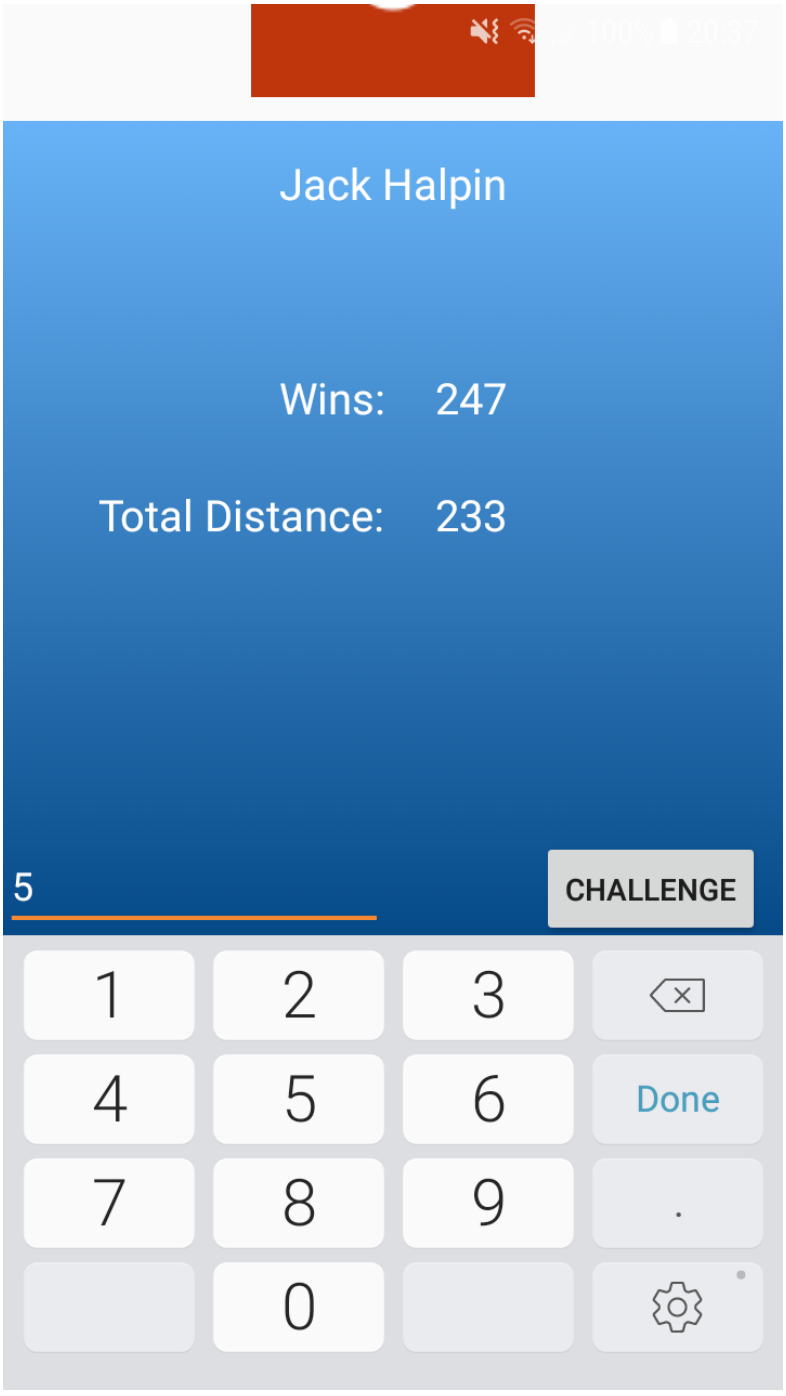
Search Users Page:



Challenge (Waiting on Opponent) Page

After the user has challenged another user on the Search Users page they will be brought to a ‘Waiting for Opponent…’ page showing the image and details of the opponent they have challenged, the user will wait on this page until their opponent has accepted and challenged them in return.

Challenge (Waiting on Opponent) Page: Setting the distance in km.

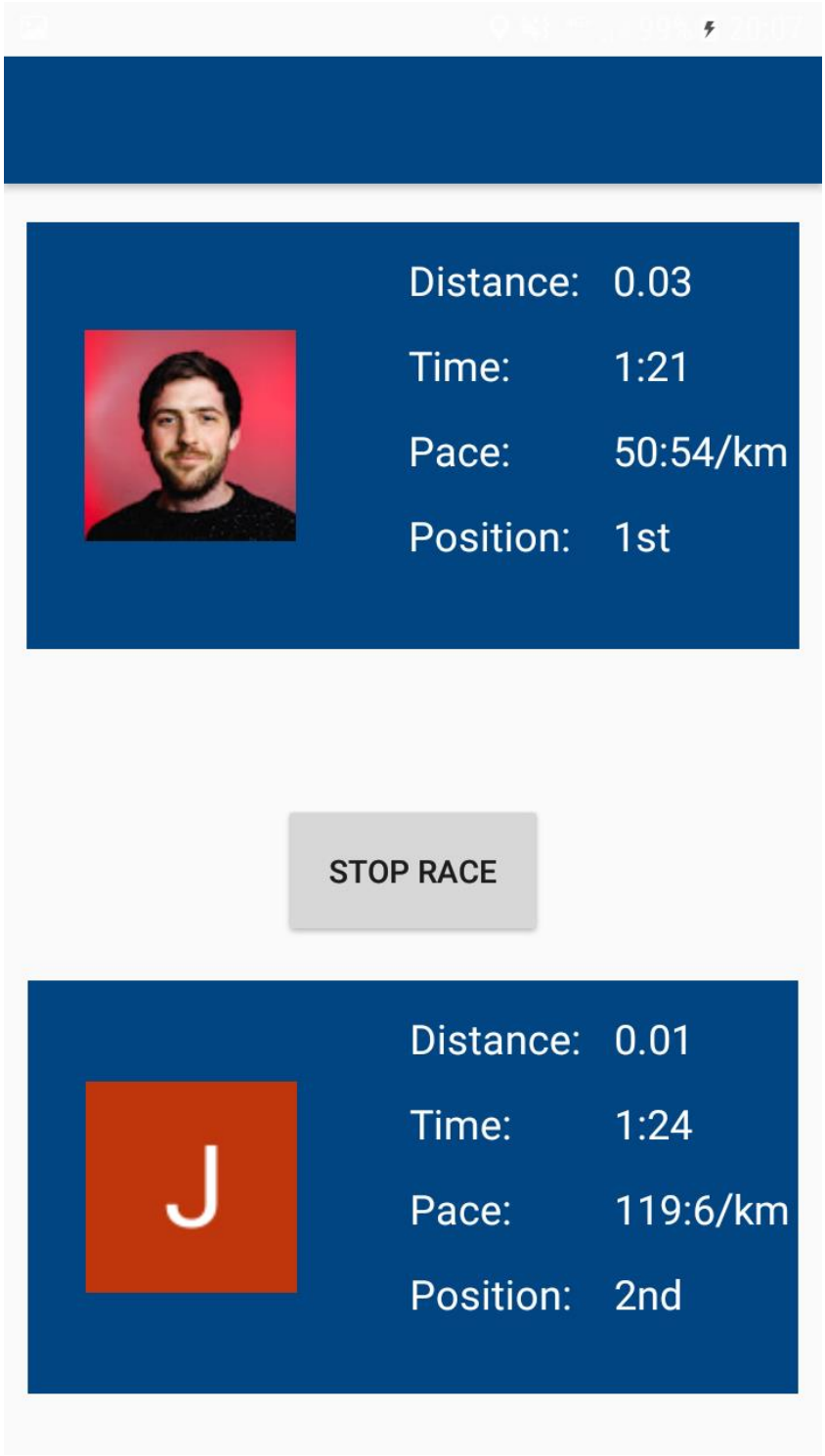


Challenge (Waiting on Opponent) Page: Waiting for opponent's response

Run/Race Page

The run/race activity page shows in a split screen format, with the top half showing the signed in user's statistics against their opponent's statistics in the bottom half the screen. There is also a 'stop race' button that ends tracking for that race and ends it instantly.

Run/Race Page:

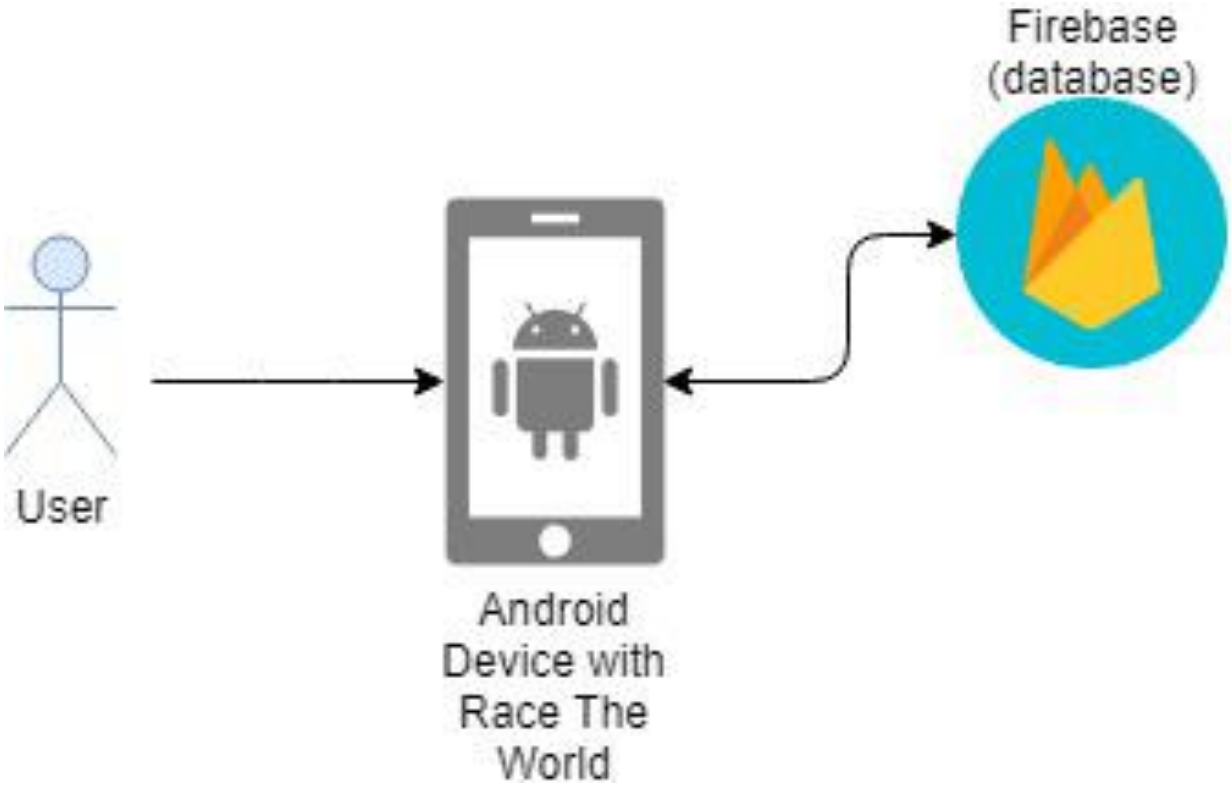


Race Finish/Stop Page

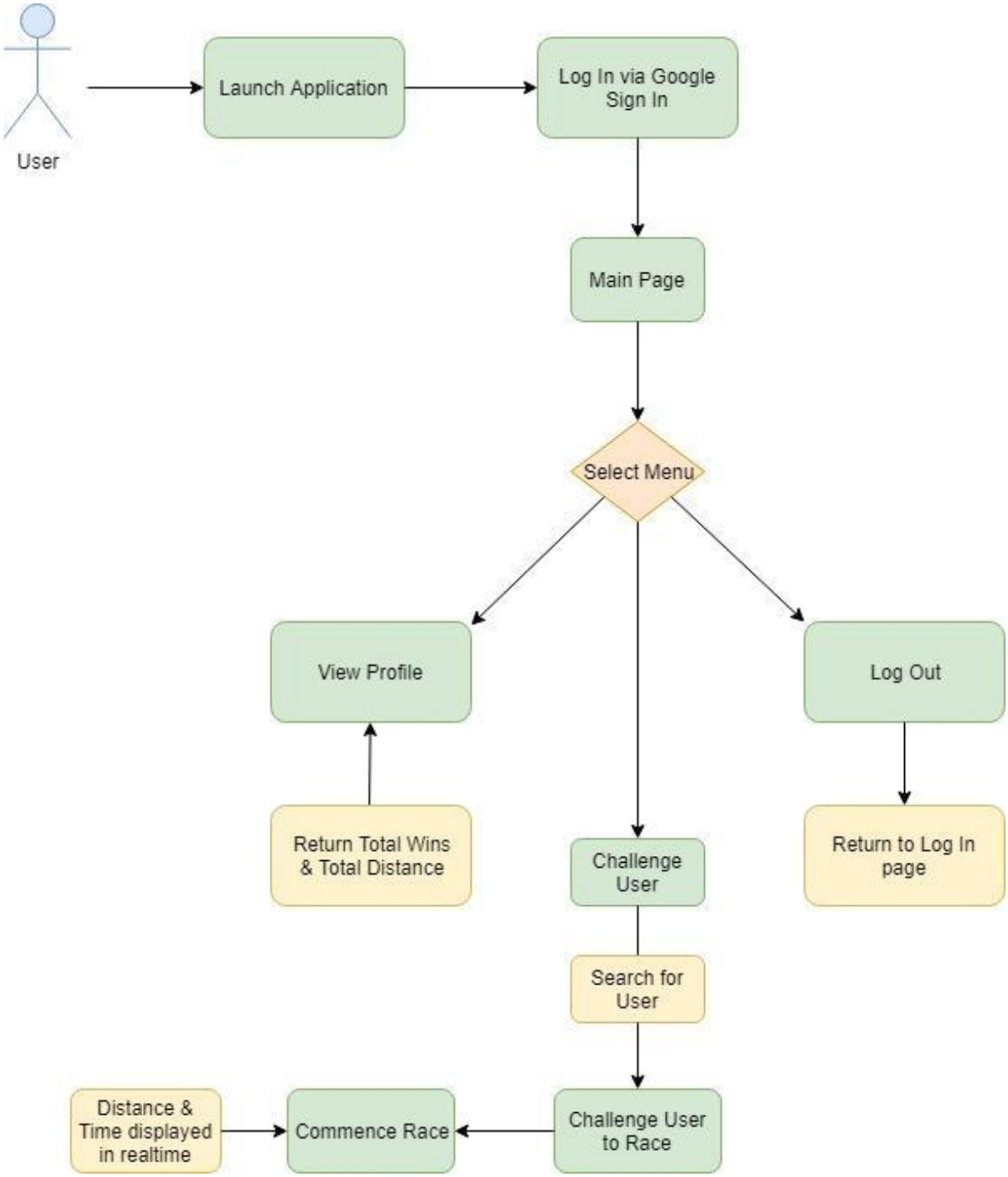
When one of the two users reach the set distance for the race or if a user chooses to press the “Stop Race” button during the race the leader at that time will be given the win for that race.

Race Finish/Stop Page:

System Architecture



Flow Diagram



Analysis and Design

Architecture Design

Logical View

Race the world logical view is made up three different parts, the user interface is build using xml, android activities and fragments then give functionality and

logic to the xml files. Google Firebase is then the back-end system that stores, retrieves and returns all the information given in the application.

Hardware Architecture

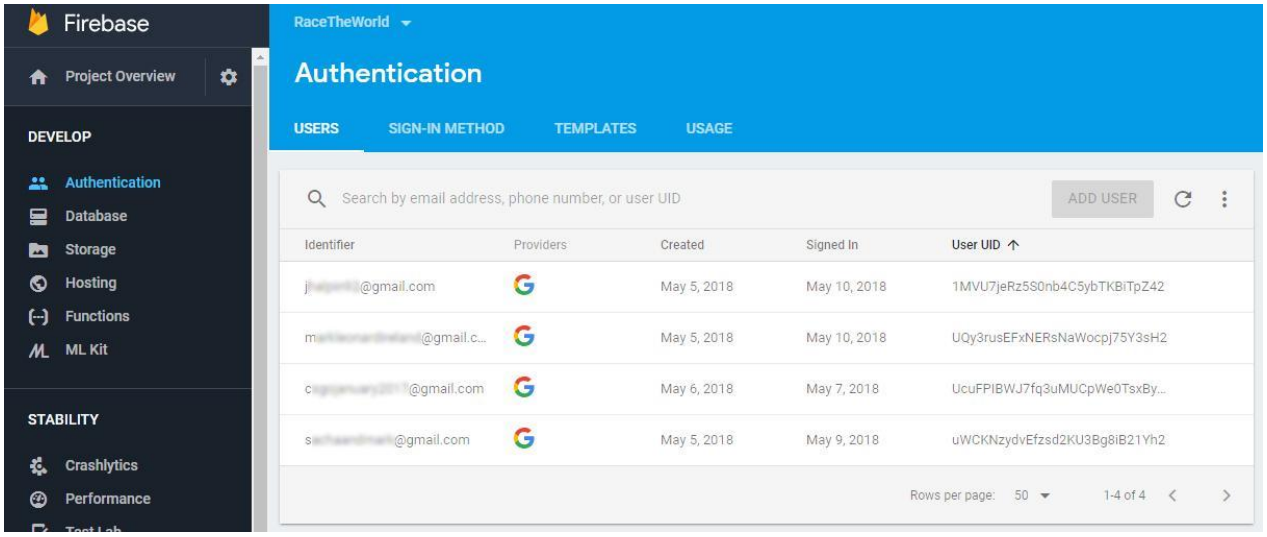
The hardware architecture required for running the application is an android device with a up to date software and operating system.

Software Architecture

The project is built using Android Studio and Google Firebase as the back-end system.

Security Architecture

Security for the application is done through the Google sign in with the Firebase authentication system.



Implementation

For the implementation of this project, it was decided to use the Java Programming language with Android Studio IDE as this has become the standard for developing Android applications worldwide. It has proven to be a reliable IDE for developing android applications. Also, the use of Google's Firebase was decided on because it is integrated into Android Studio making them work very well together. Firebase's authentication and real time database were also very convenient throughout this project.

System Design

This section of the document will look at a few different code snippets showing how the various functionalities outlined in this document worked.

Sign in with Google / Firebase Authentication

```
package com.example.mark.racetheworld.Activities;

import ...

public class GoogleSignIn extends AppCompatActivity {
    protected static final int MY_PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION = 1;
    SignInButton button;
    FirebaseAuth mAuth;
    private final static int RC_SIGN_IN = 2;

    GoogleApiClient mGoogleApiClient;
    FirebaseAuth.AuthStateListener mAuthListener;

    @Override
    protected void onStart() {
        super.onStart();

        mAuth.addAuthStateListener(mAuthListener);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_google_sign_in);

        int permissionCheck = ContextCompat.checkSelfPermission( context: this, android.Manifest.permission.ACCESS_FINE_LOCATION);


        //TODO: Hande the event that the user denies access to GPS
        //If the permission is denied we have to request it.
        if (permissionCheck == PackageManager.PERMISSION_DENIED){
            ActivityCompat.requestPermissions( activity: this,
                    new String[]{android.Manifest.permission.ACCESS_FINE_LOCATION},
                    MY_PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION);
        }


        button = (SignInButton) findViewById(R.id.googleBtn);
        mAuth = FirebaseAuth.getInstance();

        button.setOnClickListener((view) -> { signIn(); });


        mAuthListener = (AuthStateListener) (firebaseAuth) -> {
                if (firebaseAuth.getCurrentUser() != null){
                    startActivity(new Intent( packageContext: GoogleSignIn.this, Menu.class));
                    finish();
                }
        };

        GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
```

```
84          GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
85                  .requestIdToken(getString(R.string.default_web_client_id))
86                  .requestEmail()
87                  .build();
88
89          mGoogleApiClient = new GoogleApiClient.Builder( context: this)
90                  .enableAutoManage( fragmentActivity: this /* FragmentActivity */, (connectionResult) -> {
93                          Toast.makeText( context: GoogleSignIn.this,  text: "Something Went Wrong", Toast.LENGTH_SHORT).show();
94                  })
96                  .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
97                  .build();
98      }
99
100     private void signIn() {
101
102         Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
103         startActivityForResult(signInIntent, RC_SIGN_IN);
104     }
105
106     @Override
107     public void onActivityResult(int requestCode, int resultCode, Intent data) {
108         super.onActivityResult(requestCode, resultCode, data);
109
110         // Result returned from launching the Intent from GoogleSignInApi.getSignInIntent(...);
111         if (requestCode == RC_SIGN_IN) {
112             GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);
113             if (result.isSuccess()) {
114                 // Google Sign In was successful, authenticate with Firebase
115                 GoogleSignInAccount account = result.getSignInAccount();
116                 firebaseAuthWithGoogle(account);
117             } else {
118                 //Google Sign in failed.
119                 Toast.makeText( context: GoogleSignIn.this,  text: "Auth Went Wrong", Toast.LENGTH_SHORT).show();
120             }
121         }
122     }
123
124     @   private void firebaseAuthWithGoogle(GoogleSignInAccount account) {
125         AuthCredential credential = GoogleAuthProvider.getCredential(account.getIdToken(),  s1: null);
126         mAuth.signInWithCredential(credential)
127                 .addOnCompleteListener( activity: this, (task) -> {
130                     if (task.isSuccessful()) {
131                         // Sign in success, update UI with the signed-in user's information
132                         Log.d( tag: "TAG",  msg: "signInWithCredential:success");
133                         FirebaseUser user = mAuth.getCurrentUser();
134                         //updateUI(user);
135                     } else {
136                         // If sign in fails, display a message to the user.
137                         Log.w( tag: "TAG",  msg: "signInWithCredential:failure", task.getException());
138                         Toast.makeText( context: GoogleSignIn.this,  text: "Authentication Failed.",
139                                 Toast.LENGTH_SHORT).show();
```

Menu Navigation

```java
package com.example.mark.racetheworld.Activities;

import ...

public class Menu extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener {
    private DrawerLayout drawer;
    private FirebaseAuth mAuth;
    private GoogleApiClient mGoogleApiClient;
    private View mNavHeaderView;
    private FirebaseDBHelper mHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menu);


        mHelper = new FirebaseDBHelper();
        GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
                .requestIdToken(getString(R.string.default_web_client_id))
                .requestEmail()
                .build();

        mGoogleApiClient = new GoogleApiClient.Builder( context: this)
                .enableAutoManage( fragmentActivity: this /* FragmentActivity */, (connectionResult) -> {
                        Toast.makeText( context: Menu.this,  text: "Something Went Wrong", Toast.LENGTH_SHORT).show();
                })
                .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
                .build();



        mGoogleApiClient.connect();

        mAuth = FirebaseAuth.getInstance();
        FirebaseDBHelper helper = new FirebaseDBHelper();
        helper.createUserIfNotExists(mAuth.getCurrentUser().getUid());
        //helper.resetReadyState(mAuth.getCurrentUser().getUid());

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        NavigationView navigationView = findViewById(R.id.nav_view);

        drawer = findViewById(R.id.drawer_layout);
        mNavHeaderView = navigationView.getHeaderView( index: 0);

        String uid = FirebaseAuth.getInstance().getCurrentUser().getUid();
        DatabaseReference userRef = FirebaseDatabase.getInstance().getReference().child("Users").child(uid);
        Log.e( tag: "onCreateView",  msg: "Creating View");
        ValueEventListener eventListener = new ValueEventListener() {
```

```
84          String uid = FirebaseAuth.getInstance().getCurrentUser().getUid();
85          DatabaseReference userRef = FirebaseDatabase.getInstance().getReference().child("Users").child(uid);
86          Log.e( tag: "onCreateView",  msg: "Creating View");
87          ValueEventListener eventListener = new ValueEventListener() {
88              @Override
89              public void onDataChange(DataSnapshot dataSnapshot) {
90                  System.out.println(dataSnapshot.toString());
91                  Log.e( tag: "OnDataChange",  msg: "Creating View");
92                  if(dataSnapshot.exists()) {
93                      //create new user
94                      Log.e( tag: "OnDataChange",  msg: "Creating View");
95
96                      User currUser = dataSnapshot.getValue(User.class);
97                      TextView name = mNavHeaderView.findViewById(R.id.user_name);
98                      TextView email = mNavHeaderView.findViewById(R.id.user_email);
99                      ImageView image = mNavHeaderView.findViewById(R.id.user_profile);
100
101                     mHelper.setImageFromUrl(image, currUser.photoURL);
102                     name.setText(currUser.name);
103                     email.setText(currUser.email);
104                 }
105             }
106
107             @Override
108             public void onCancelled(DatabaseError databaseError) {}
109         };
110         userRef.addListenerForSingleValueEvent(eventListener);
111
112
113
114         // Update the elements in the navigation view
115
116         navigationView.setNavigationItemSelectedListener(this);
117
118         ActionBarDrawerToggle toggle = new ActionBarDrawerToggle( activity: this, drawer, toolbar,
119                 "Open navigation drawer", "Open navigation drawer");
120         drawer.addDrawerListener(toggle);
121         toggle.syncState();
122
123         if (savedInstanceState == null) {
124         getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
125                 new ProfileFragment()).commit();
126         navigationView.setCheckedItem(R.id.nav_profile);}
127     }
128
129     @Override
130     public boolean onNavigationItemSelected(@NonNull MenuItem item) {
131         switch (item.getItemId()) {
132             case R.id.nav_profile:
133                 getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
134                         new ProfileFragment()).commit();
```

```
126             navigationView.setCheckedItem(R.id.nav_profile);}
127         }
128
129         @Override
130         public boolean onNavigationItemSelected(@NonNull MenuItem item) {
131             switch (item.getItemId()) {
132                 case R.id.nav_profile:
133                     getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,
134                             new ProfileFragment()).commit();
135                     break;
136                  case R.id.nav_challenge:
137                      Intent intent = new Intent( packageContext: Menu.this, SearchUser.class);
138                      startActivity(intent);
139                     break;
140                 case R.id.nav_my_challenges:
141                     Intent intent3 = new Intent( packageContext: Menu.this, MyChallengesActivity.class);
142                     startActivity(intent3);
143                     break;
144                 case R.id.nav_logout:
145                     mAuth = FirebaseAuth.getInstance();
146                     mAuth.signOut();
147                     // Google sign out
148                     Auth.GoogleSignInApi.signOut(mGoogleApiClient).setResultCallback(
149                             (ResultCallback) (status) → {
152                                 Intent intent = new Intent( packageContext: Menu.this, GoogleSignIn.class);
153                                 startActivity(intent);
154                             });
156             }
157
158             drawer.closeDrawer(Gravity.LEFT);
159
160             return true;
161         }
162
163         @Override
164         public void onBackPressed() {
165             if (drawer.isDrawerOpen(GravityCompat.START)) {
166                 drawer.closeDrawer(GravityCompat.START);
167             } else {
168                 super.onBackPressed();
169             }
170
171             super.onBackPressed();
172         }
173     }
174
```

Search User through Firebase

```
90
91      private void firebaseUserSearch(String searchText) {
92
93          Log.e( tag: "seaarching for ", searchText);
94          Toast.makeText( context: SearchUser.this,  text: "Started Search", Toast.LENGTH_LONG).show();
95
96          Query firebaseSearchQuery = mUserDatabase.orderByChild("name").startAt(searchText).endAt(searchText + "\uf8ff");
97
98          FirebaseRecyclerOptions<User> options =
99                  new FirebaseRecyclerOptions.Builder<User>()
100                         .setQuery(firebaseSearchQuery, User.class)
101                         .build();
102
103         mAdapter = new FirebaseRecyclerAdapter<User, UsersViewHolder>(options)
104         {
105             @Override
106             public UsersViewHolder onCreateViewHolder(ViewGroup parent, int viewType)
107             {
108                 Log.e( tag: "onCreateViewHolder: ",  msg: "Creating View Holder");
109                 View view = LayoutInflater.from(parent.getContext())
110                         .inflate(R.layout.List_Layout, parent,  attachToRoot: false);
111
112                 return new UsersViewHolder(view);
113             }
114             @Override
115             protected void onBindViewHolder(UsersViewHolder viewHolder, int position, User model) {
116                 Log.e( tag: "onBindViewHolder", model.name);
117                 viewHolder.setDetails(getApplicationContext(), model.name, model.racesWon, model.email, model.photoURL);
118             }
119         };
120
121         Log.e( tag: "firebaseUserSearch: ",  msg: "Attaching adapter");
122         mAdapter.startListening();
123         mResultList.setAdapter(mAdapter);
124
125     }
126
```

Testing

This section will describe the testing methods that have been carried out to test the quality of the application and ensure everything is functioning as it should be. The benefits of testing include guaranteeing the quality of the application, increased consumer satisfaction and ensuring the user experience is a positive one.

Manual testing will also be performed in the form of live runs and races to determine that the application is functioning correctly. To achieve maximum accuracy, the results will be compared to other well-established running applications on the Android market to gauge the app's overall performance. Manual testing will also be used in testing simple aspects of the application like signing in through Google sign in and searching the application for other users.

The following is a test carried out to see if the ability to sign in and challenge another user to a race is functioning correctly.

Steps	Task	Expected Result	Result	Status
1	User must sign in on the main page.	The user will successfully sign in through Google sign in.	The user has successfully signed in with their Google account.	Pass
2	User must navigate to 'Challenge' option on the menu	The user will click the challenge option	User has chosen the challenge option	Pass
3	User must search and find then challenge the user they wish to race	The user finds the opponent they wish to challenge.	The user finds and challenges their opponent to a 5km race.	Pass
4	User must wait on opponent to accept challenge.	The user is left waiting until challenge accepted.	The user is left waiting until challenge accepted.	Pass
5	The Opponent must accept the challenge and race will begin.	The opponent accepts, and the race begins instantly.	Race begins as the opponent has accepted the challenge.	Pass
6	Both users' appropriate data is tracked, and the race begins.	Race begins	Race begins	Pass

The test has been a success showing that the application can manage a user signing in and competing against another user in a real time race.

Conclusion

Even though I have achieved what I had set out to achieve building application that allows two different users to compete in a race in real time with each other I am slightly disappointed that I could not add additional features that I had originally planned to implement. Such as keeping count of the calories that the user is burning, taking into consideration the elevation they are ascending or descending or counting how many steps the user has taken in a run. But given the circumstances of my full-time work that could be anything up to 50 hours in one week and trying to study a part-time computing course I am proud of what I have achieved with the application.

The gamification aspect of this application is the unique selling point of the app, by introducing gamification I have introduced something that is rarely found in fitness apps, so this alone would be a niche in the market. However, the fitness application market is completely saturated, as result it would be difficult to market this application as there is literally thousands of fitness applications.

Further Development

Over time the system could also expand out to help users track and race across many activities such as cycling or swimming which would utilize much of the same technology that has already been developed for the core application. This should make the application more accessible and help increase user number in later stages of development.

Other such features that could go into further development would be introducing leagues that would be based on users' performance. Overtime they could be tracked, and they would be split into different levels of leagues where they could be promoted or relegated at the end of each month adding an even more competitive edge to the application, users could then only compete with fellow runners within their same league.

Finally, the app could expand out to other mobile platforms, specifically iOS and Windows Mobile and wearables which would be useful for people trying to track their activity.

Bibliography

Bryant, K., Zabonick, R. and Zabonick, R. (2018). Health Benefits to Using a Fitness App. [online] Club Solutions Magazine. Available at: http://clubsolutionsmagazine.com/2015/10/health-benefits-to-using-a-fitness-app/ [Accessed 5 May 2018].

Virtuagym Blog. 2018. The Health Benefits of Fitness Apps. [ONLINE] Available at: https://virtuagym.com/blog/lifestyle/3-reasons-why-you-should-use-fitness-apps/. [Accessed 6 May 2018].

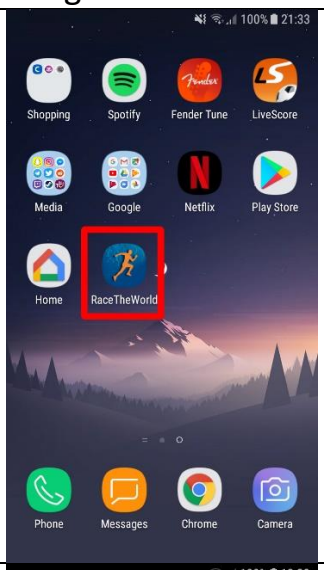
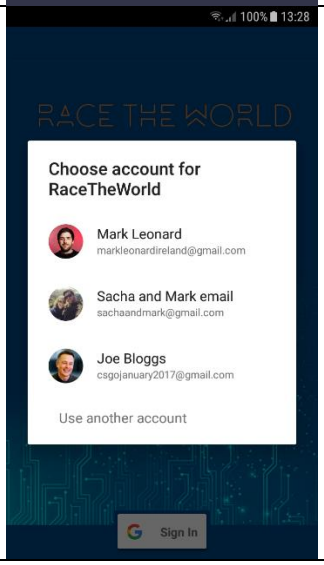
Android Studio - Wikipedia. 2018. Android Studio - Wikipedia. [ONLINE] Available at: https://en.wikipedia.org/wiki/Android_Studio. [Accessed 11 May 2018].

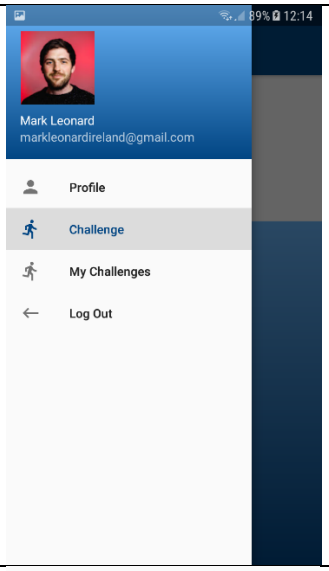
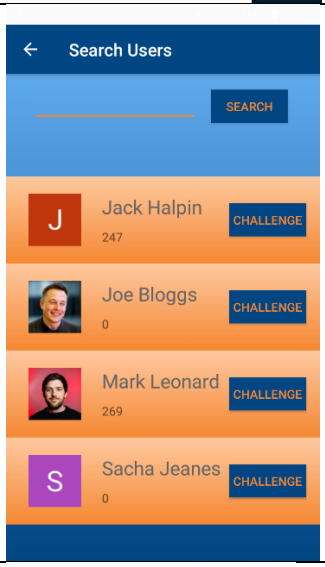
Manual Testing Process Life Cycle in Software Testing. 2018. Manual Testing Process Life Cycle in Software Testing. [ONLINE] Available at: http://toolsqa.com/software-testing/manual-testing/. [Accessed 9 May 2018].

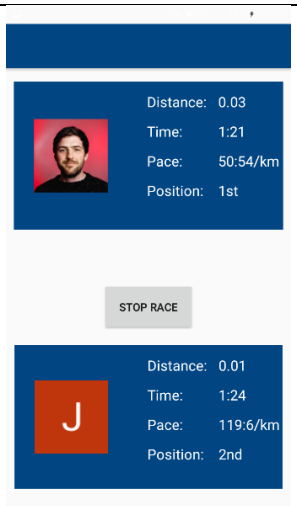
Gauss Development. 2018. Benefits of Software Testing. [ONLINE] Available at: https://gauss-development.com/benefits-software-testing/. [Accessed 11 May 2018].

Appendix

User Manuel

Steps	Task	Image
1	Select the application on your device.	
2	Sign in through Google	

3	Navigate to ‘Challenge’ on the Menu.	
4	Search for the user you wish to race and challenge them to a distance of your choosing.	
5	Wait on their acceptance of the challenge.	

6	On acceptance start running as the race has begun.	
7	First user to the set distance target is the winner.	

Project Proposal

‘RACE THE WORLD’

Mark Leonard

x14562027@student.ncirl.ie

BSc (Hons) in Computing

Cloud Computing

27/10/2017

Objectives

The goal of developing this application is to create a platform where by users can challenge friends or other users to a race anywhere in the world in real time, while also keeping track of their physical wellbeing.

The application will allow runners to send another user a challenge to run an agreed distance, 5km for example, pick a set time for both runners to start and then countdown both runners into the race. For example, somebody in Ireland could be wake on a Sunday morning and challenge a stranger in Japan to a race at 9am Irish time and the person in Japan could be ready for their race at 5pm Japanese time, when the challenge is accepted both runners will be counted in to start running in real time. Expectantly the application will be able to inform the user as they are racing how far they have ran or how far their opponent is ahead or behind them. This could be done potentially through text to speech.

Additionally, the application will track the user using GPS and show where they have travelled and what distance. There already are numerous existing applications on the market for runners/joggers that monitor and track such things as durations, distance, average speed, calories burned, heart rate, altitude climbed or descended as well as other features. However, from personal experience I have never found one single application that offers all features, some only offer one, others maybe two or three features, this usually results in having to run two or three applications at once if you are hoping to track and monitor all these features. With the application I am developing I hope to implement as many of these features as possible.

Furthermore, the application will store user's average speeds and average times and collect the data so that when it comes to two random users challenging each other from anywhere in the world, two similarly skilled people will be paired together, as opposed to a new runner being paired with a marathon runner. Some sort of a league could be implemented with a relegation and promotion system that would be an incentive for people to improve themselves.

Visualisation will be another key feature to the application, many people enjoy looking at simple graphs and charts that show the progression they have made, I plan to represent the user's progress in graph form in the application. Viewing the graphs and charts would be further motivation to constantly improve.

Background

From experience of using multiple fitness applications on the Google Play store, designed for runners and joggers, I have never really found an application that offers all the features I have mentioned previously in my 'objectives' section. For this reason, I have desired to develop an application that has numerous features in an all-in-one type application. Having one single application makes tracking and monitoring much easier for the end user.

However, I don't wish to simply just develop an application that is essentially half a dozen existing applications merged into one, I want to add something new, something that I have yet to find exists in the marketplace, this is where the real-time racing idea comes into play, and I want this to be the main feature of the application.

Originally the idea for the real-time racing came about when my cousin and myself were chatting online while playing some games. We both decided to take a break from gaming and coincidently we were both going for a run, we both agreed we'd run 5km and the first person back online would be the winner. Obviously, this is not a reliable means of tracking and timing a run, we didn't even know if the other ran 5km at all. But from this the idea of having a mobile application that could do this for us and monitor our opponent came to mind.

Technical Approach

For this software project I will be taking an agile management approach, I think breaking the project down into smaller incremental builds will work best for this application.

Initial research for this project will involve downloading and testing similar fitness applications, I will analysis what works best, which

features are more popular, and particularly focus on the highest rated applications on the market and research what it is that makes these the best available.

I will also have to research and experiment with new software, languages and frameworks and other new technologies I plan on using, that I haven't previously used in my life. Such as android application development, Firebase, location GPS tracking in applications.

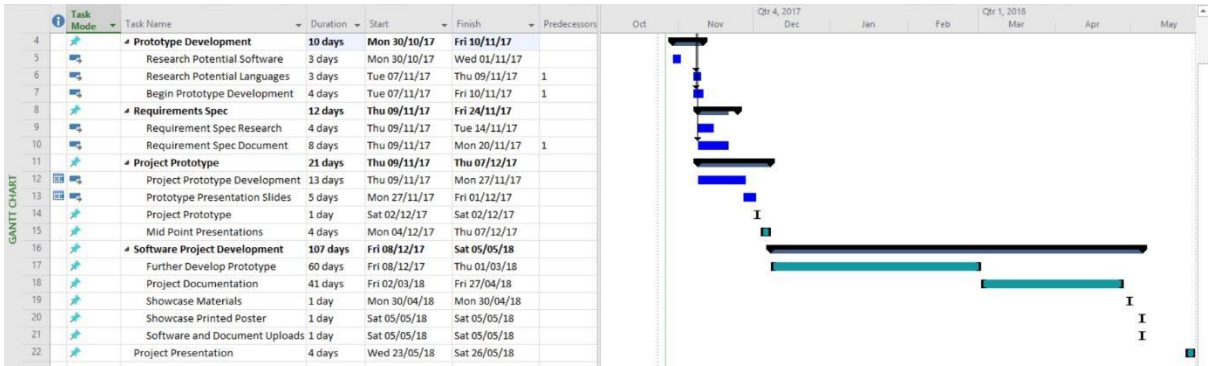
Special resources required

For the development of the application I will using various resources. I have already invested in ‘Android App Development for Dummies’ from the ‘for Dummies’ series, which I have so far found useful as I have never previously used Android Studio.

Other resources I have being using and will be using are online tutorials, from various websites but mainly from youtube.com and pluralsight.com, the latter I have signed up for $29 a month, having previously used pluralsight.com in the past I have found this method of learning works well for me personally.

In terms of hardware for this application I will need an android enabled mobile device to run it on, also further down the line I will probably need a ‘running armband’ for some of the features I plan to implement to be monitored. A ‘running armband’ is a pocket for your mobile device that can be strapped to your upper arm while you are running.

Project Plan



	Task Mode	Task Name	Duration	Start	Finish	Predecessors
4		**Prototype Development**	**10 days**	**Mon 30/10/17**	**Fri 10/11/17**	
5		Research Potential Software	3 days	Mon 30/10/17	Wed 01/11/17	
6		Research Potential Languages	3 days	Tue 07/11/17	Thu 09/11/17	1
7		Begin Prototype Development	4 days	Tue 07/11/17	Fri 10/11/17	1
8		**Requirements Spec**	**12 days**	**Thu 09/11/17**	**Fri 24/11/17**	
9		Requirement Spec Research	4 days	Thu 09/11/17	Tue 14/11/17	
10		Requirement Spec Document	8 days	Thu 09/11/17	Mon 20/11/17	1
11		**Project Prototype**	**21 days**	**Thu 09/11/17**	**Thu 07/12/17**	
12		Project Prototype Development	13 days	Thu 09/11/17	Mon 27/11/17	
13		Prototype Presentation Slides	5 days	Mon 27/11/17	Fri 01/12/17	
14		Project Prototype	1 day	Sat 02/12/17	Sat 02/12/17	
15		Mid Point Presentations	4 days	Mon 04/12/17	Thu 07/12/17	
16		**Software Project Development**	**107 days**	**Fri 08/12/17**	**Sat 05/05/18**	
17		Further Develop Prototype	60 days	Fri 08/12/17	Thu 01/03/18	
18		Project Documentation	41 days	Fri 02/03/18	Fri 27/04/18	
19		Showcase Materials	1 day	Mon 30/04/18	Mon 30/04/18	
20		Showcase Printed Poster	1 day	Sat 05/05/18	Sat 05/05/18	
21		Software and Document Uploads	1 day	Sat 05/05/18	Sat 05/05/18	
22		Project Presentation	4 days	Wed 23/05/18	Sat 26/05/18	

Technical Details

For the development of this application I have looked at various approaches I could take to building it. For the database I have considered MySQL as I have experience using it from second and third year modules in NCI, for the backend of the application I am

planning on using Python, more so as a learning experience more than anything as I haven't previously used it. For the framework I have considered potentially using Flask or Django. I will be using Android Studio for the building of the application using Java. It will be hosted on Amazon Web Services. On recommendation I have also considered potentially using Firebase, this is something that is new to me, but it looks like it could potentially be an option when developing the application.

Evaluation

The application will be tested throughout its development. During the initial development stages I will be constantly testing code as I am writing to ensure any bugs and errors are identified as I go. I will seek friends and family that use android devices to test the application during its development and seek feedback on how it performs. I will incorporate java unit testing throughout the development cycle.

Monthly Journals

September 2017

19th September 2017

Back to college this evening, the start of the final year, first class back and we were briefed on our Software Project, and what is expected for the coming weeks and months ahead. We have project proposals coming up on the 3rd,5th and 6th of October, so it's best to get finalising some of the ideas I've had for my software project soon.

30th September 2017

Spent this evening discussing my software project ideas with some family and friends, narrowing it down to two final options, both which are android application ideas. The reason for picking these were I felt I would enjoy developing these applications and they would be challenging but also achievable in the time frame to complete them by next May. I also have never previously developed an application using Android Studio, so this is a technology I want to experience in my time in college.

October 2017

4th October 2017

Finalised my idea for my software project ahead of my project proposal tomorrow. Planning to develop an android fitness/running application that tracks users and allows them to race other users in real-time anywhere else in the world.

5th October 2017

Had my project proposal today with Dominic Carr, Thibaut Lust and Eugene McLaughlin. Fortunately, all three approved of my idea and even suggested some other ideas I could implement in to the project.

16th October 2017

My supervisor for the software project will be Dominic Carr, I'm looking forward to working with him over the coming months, and it helps that I have him for class every Tuesday evening.

19th October 2017

Met with Dominic for the first time today for a few minutes, showed him a list I had made up of technologies I plan on using for the project, he approved of some and suggested others to me, such as 'Firebase'. To be honest this was very helpful as some of the technologies he suggested I had never used or heard of before, but for the most part they will only help improve what I hope to achieve with my application.

25th October 2017

Been working away on my project proposal which is due on the 27th, have sent the work I've done so far to Dominic for feedback, he says I'm on the right track with it, but suggested I discuss in more detail the competitors of the application. I will consider this when finalising my proposal over the next 2 days.

27th October 2017

Uploaded my Project Proposal Document today, got as much into as I could.

November 2017

17th November 2017

Been busy with all my other modules I haven't got the dedicate the time I was hoping I could to the Software Project, made a start on the requirement specification today however. It's due on the 24th so I should be on course to have it done by then.

23rd November 2017

Continued working away on my requirement specification this evening, but began to feel unwell, and I'm not getting the work done I had hoped. Also missed my opportunity to send my document to Dominic for feedback, as I was not happy with the work load I had done to send it to him.

24th November 2017

Requirement specification due today and unfortunately a bad flu has got a hold of me, and I don't think I will have an opportunity to finish the document as I hoped.

27th November 2017

Been bed bound all weekend and must submit my requirement specification tonight at the latest to present at the mid-point. Unfortunately, the flu has not

helped my work whatsoever and I've uploaded a document that I am not happy with at all. This document is probably the most disappointing work I have uploaded in my time here in NCI, really feel I have fallen behind because of this set back.

December 2017

15th December 2017

I got to attempt some Android tutorials on YouTube. I have been trying to familiarise myself better with Android studio as I have not used it in the past too often. I done some basic tutorials on user sign-ins and tracking user location.

16th December 2017

Started looking at some Firebase tutorials today. Having never previously used this it looks like it will be a very useful tool for this project. It is very convenient that it is synced up with Android studio making it easier to use.

21th December 2017

It is a very busy time in work at this time of year so unfortunately, I have not got an opportunity to have much of a look at my Software Project.

January 2018

2nd January 2018

Exams are coming up over the next few weeks, so I do not have much of a chance to look at myself a project as I have to focus on my studies and projects for other modules.

February 2018

9th February 2018

I have been very busy at work and as a result I've been struggling to keep up with my modules let alone my software project.

20th February 2018

Project is beginning to take shape in Android studio as of today I have successfully linked up my Firebase Project with my Android Studio Project and created a sign in page using Google sign in with Firebase Authentication.

March 2018

3rd March 2018

The bad weather storm has resulted in me becoming very busy in work on the farm, as a result I've been working 12-14 hours a day and only getting home about 9 p.m. in the evenings. Leaving very little time for me to get a look at my college work before bedtime.

16th March 2018

I emailed Dominic today to get feedback on my profile and poster so far. He suggested I make some minor changes to my summary of my profile and he said my poster has got off to a good start which I am happy with considering early stages of designing the poster.

April 2018

7th April 2018

Made further progression in my Android studio application today. I have added a real time tracker of the user pinning their longitude and latitude. I originally intended to show their location on map, but I have decided that since they could be competing against strangers it is not a good idea for somebody to know your exact location.

14th April 2018

I have added a navigation sliding menu to my Android app today, it is beginning to take shape. I can now successfully login a user through Google with Firebase authentication and they are brought to their profile page and using the menu they can navigate to track a run.

17th April 2018

I have a project due for my cloud application development module this coming Saturday as a result I have been busy with my Ruby on Rails work for that module.

22nd April 2018

I have two exams this coming Wednesday and Friday, so they will be taking priority for the coming days

May 2018

4th May 2018

Now all my other projects and exams from other modules are finished my focus is now completely on my final year software project.

5th May 2018

Due to still working crazy long hours in work I really must take advantage of my free time at the weekends to work on my project. Luckily, I have made good progress on the project in the past few days probably more than I have in the previous month.

6th May 2018

The application has really taken shape this weekend, a lot of functionality has begun to work together well, allowing users to compete against each other in a real-time race tracking both time and distance.

9th May 2018

Emailed Dominic again today, although it this late in the semester I updated him on why I had been so far behind in my software project work as of now but I have got a lot of work done and I am very happy with what I have managed to achieve in the time considering my work schedule and being a part-time student. I also got some advice for my report and what key things I should include in it.

11th May 2018

Spent today tidying up my code and adding some user interface features as I had previously been using the Android studio default theme. The application now looks aesthetically pleasing and works for its main purpose to allow two users to race each other.

12th May 2018

I am nearly finished my project and I have to attend a wedding this weekend, so I will fix any minor changes that have to be fixed before tomorrow's deadline, while also finalizing my document.