Basit Ali Soomro

OneWorld

Find volunteers and set up volunteering events

Information Technology
2019

Aknowledgement

I would like to thank all my teachers at VAMK who helped me to get to this point. It has taken me much longer than I expected to finish this degree program and I appreciate the patience of the teachers whose courses I retook every year trying to pass them.

I would also like to thank Pirjo Prosi for being my supervisor and guiding me through my thesis process and helping me to structure my thesis to the best of my abilities.

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Information Technology

ABSTRACT

Author	Basit Ali Soomro
Title	OneWorld - Find volunteers and set up volunteering events
Year	2019
Language	English
Pages	37
Name of Supervisor	Pirjo Prosi

Volunteering is a cornerstone of our society. We always tend to think on a much wider perspective hence the reason why people agree to work without pay by choice. Therefore, working for a non-profit organization that benefits society is appealing to us.

The purpose of this thesis is to develop a concept which would make the volunteering process easier. It's commonly known that people often have a strong desire to volunteer yet never proceed further because the process of finding a cause your passionate about and joining that cause can be a tedious task.

As a solution to this problem, An Android application will be developed so that it can be used to find volunteering events as well as create your own. These events will be easily accessible and simple to register to. The application will be developed using Java and Android studio will be my IDE and Firebase will serve as my server.

Keywords Java, Android, Firebase, Mobile, Application

CONTENTS

LIST OF FIGURES AND TABLES

LIST OF CODE SNIPPETS

LIST OF ABBREVIATIONS

JSON	JavaScript Object Notation
OS	Operating System
GUI	Graphical User Interface
IDE	Integrated Development Environment
SDK	Software Development Kit
ADT	Android Development Tools
JRE	Java Runtime Environment
API	Application Programming Interface
DBMS	Database Management System
OOP	Object Oriented Programming
JSP	Java Server Programming
JDK	Java Development Kit
URL	Uniform Resource Locator
OHA	Open Handset Alliance
ART	Android Runtime
WORA	Write Once, Run Anywhere

1 INTRODUCTION

Mobile phones are becoming increasingly common and information is easily accessible. Nowadays there are mobile applications aimed at simplifying most social activities people are interested in. For example, Airbnb and Uber are two giants who have a supply and demand model and filled huge gaps in the market. However, There is one area that appears to be not well facilitated with mobile applications and that's is volunteering. There is a need for an application that can facilitate the volunteering process as it is currently quite difficult to find and participate in volunteering events.

The main objective of my thesis is to develop an Android application which will make the volunteering significantly easier. Mobile phones are available to the most part of our population and making an application will be the best way to reach a worldwide audience.

Essentially, the application will provide a platform for individuals/companies to set up events and look for volunteers as well as provide an opportunity to individuals to sign up and help in any way they can.

1.1 Background

Inspiration for this topic appeared when I was contemplating the increased immigration happening around the world. People are leaving their homes and travelling to foreign countries, whether they are students or going for work or just moving to get better living conditions. With the increased immigration comes issues with integration, most people who move abroad face difficulties with assimilation into the new society. Being alone in a foreign land is hard at first and can make people feel homesick. It takes time to adjust to a new environment and volunteering is a great way for foreigners to meet new people while integrating into local communities.

1.2 Objective

The objective of this thesis is to provide an easier way for individuals to volunteer for events in their area and support local causes which can therefore help integrate them into the community better.

The OneWorld mobile application is for people who want to create events and find volunteers for their cause and for individuals who want to support such causes.

The application has two uses, one for individuals and one for companies that want to set up events for a specific cause. Once you are creating an event, you are able to describe the position you are creating, include the tasks involved, the number of people you need and the type of cause. The other side is for persons who can sign up and choose the type of causes they are interested in and if a volunteer position opens-up which corresponds to their wishes a notification will be sent and it is possible to sign up.

2 SIMILAR PROJECTS

During the research, I came across quite a few similar technologies that are currently operating, but most of them appear to be limited in terms of the areas they operate in.

2.1 ELEV8SOCIETY

Elev8society is a non-profit volunteer application where non-profit organizations can submit their upcoming events and volunteering opportunities. You can create your profile on the app and see all the events in the surrounding area, but currently the application only operates in the United States. Users also have the possibility to create their own events and find volunteers for causes that they hold dear. There is also an option to chat with other users who have similar ideas which can lead to collaboration and more meaningful events. /1/

2.2 CHIP'N

Chip'N is a volunteer application that connects people and encourages growth within the community. Users can create a profile which they can then use to sign up to volunteering events or create one for something they would like to support. Chip'N also has a reward system which are called "chips" which can be earned by volunteering in your community and then those chips can be spent in the marketplace within the app which offers various products, tickets, discounts and various other rewards. /2/

2.3 GiveGab

GiveGab first started as a web based social platform which helps connect volunteers and non-profit organizations. A user can sign up as either a supporter which means that you are actively looking for volunteering opportunities or looking to do some fundraising for a cause you care about. It is also

possible to sign up as an administrator which means you can raise funds for the non-profit organization you work for. At the same time, you can also manage the volunteer opportunities for your organization and promote your cause within the volunteer base. /3/

2.4 DEED

Deed is a mobile application which makes the whole volunteering process easier and fun. They currently only operate in New York and Los Angeles. It is a platform that can be used by volunteers, non-profit organizations, companies and schools. Volunteers can sign up and find opportunities in their areas and non-profit organizations can also use their integrated dashboards to search for volunteers through their database and it comes with integrated background checks too. Companies can also use this platform to promote volunteering which can then improve the workplace environment, they provide easy to use applications and provide the employer the opportunity to track the records of their company. Schools can also use this platform to create a fun and motivating volunteer experience which can encourage students to take up volunteering in their free time as well. /4/

3 TOOLS AND TECHNOLOGIES

In this section, the main tools and technologies used will be described. This application will be developed for the Android operating system. The application will be developed in Android studio and will be written in Java. Firebase will serve as the backend.

3.1 ANDRIOD OPERATING SYSTEM

Android operating system is an open source system that was initially developed by Android Inc but was later bought by Google. It was developed with the help of the members of the Open Handset Alliance. It is based on the modified version of the Linux Kernel version 2.6. The kernel is the main part of the operating system, it is used to manage the inputs and output requests from the software. It also provides the basic functionalities such as process management, memory management and device management etc. Android operating system is written mostly in Java and it also contains a set of Java core libraries. These libraries provide functionality such as to playing audio and recording video. ART is the application runtime environment which is another key component. It replaced the Dalvik Virtual Machine used in earlier versions and what ART does is translate an applications bytecode

into native instructions which are then carried out by the runtime environment. The application framework provides services to applications such as activity manager, location manager and notification manager etc. /5/ /6/ /7/

3.2 JAVA

Java is a general purpose, object-oriented programming language. It was developed by Sun Microsystems in 1991 in the USA, it was later bought by Oracle Corporation. The main motive behind developing Java was to make the language highly reliable, portable and simple. It was intended to be a WORA, which means that the compiled code can be run on different platforms without having the need to be recompiled.

WORA is achieved by compiling the Java program into an intermediary language called bytecode which is platform independent therefore it can be run on Windows, Macintosh or a UNIX machine without being recompiled or modified. /8/ /9/ /10/

3.3 ANDRIOD STUDIO

Android Studio is an IDE made by Google, it provides developers with tools that they need to build applications for the Android platform. It is available to use on Windows, Mac and Linux platforms. Android solutions can be developed with either Java or C++. Android studio working is based on a concept called continuous integration which means that developers can test their code continuously and if an error is reported it can be corrected immediately. Android studio also comes with three types of code completion called basic completion, smart completion and statement completion which can be used to speed up your workflow. These can be accessed using keyboard shortcuts. Android studio also comes with code sample browser which can help you find high-quality code samples provided by Google. Android Studio comes with internal debugging and performance improvement tools. Inline debugging can be used to enhance your code through in line verification of references, expressions and variable values. Performance profilers can be used to track your apps memory, CPU usage which can lead to finding deallocated objects and memory leaks which can lead to increased optimization. Memory profiler can be used to increase optimization even further as they can be used to track the memory allocation and keep an eye on where objects are being allocated when you perform certain actions. /11/ /12/

3.4 FIREBASE

Firebase is a comprehensive mobile development platform by Google. Firebase provides you with a variety of services, which if coded by the developer and can be time consuming. Hence Firebase provides the developer with more time and energy to improve the user experience. In traditional application development a developer would have to write both front end and back end parts of the application, however with firebase it provides the backend as a service. Firebase helps create the backend with ease, some of the things that firebase provides is Authentication, Realtime Database, Cloud Firestore, Cloud Storage, Cloud Functions, Firebase Hosting and ML Kit. Firebase Authentication makes it easy to perform secure logins which is rather difficult to implement on your own. Firebase Realtime Database provides database services which give you real time updates to data as its changed in the database and the Cloud Storage provides extremely scalable storage since it scales to exabytes of data. Cloud Functions enable you to write and deploy code without having to worry about maintaining or scaling as Google takes care of the rest. Firebase Hosting is a secure web hosting platform, which is great at delivering static content and can be deployed instantly. And finally, the ML Kit lets you take advantage of Googles expertise in machine learning without knowing much about it. For example, ML Kit can recognize text, faces and landmarks in pictures captured by a device. /13/ /14/

4 APPLICATION DESIGN

The OneWorld app is a mobile application which is used for searching open volunteering positions as well as creating your own volunteering events you are passionate about.

This application was created in Android Studio and written in Java. Firebase was used as the backend. The UI holds a lot of importance as much as the initial idea, the point was to make the whole volunteering process easier and within everyone’s reach.

So hence to drive the idea home the UI is designed to be as simple as possible to keep the focus of the users on the most important thing which was to help others. I used Java because it is a largely popular language with a vast support system, other reasons were the scalability and the significant amount of tools and libraries available to use freely.

4.1 SIGN IN WINDOW

The first window that appears when you first opening the application is the login screen which is made using java. It consists of two text fields Email and Password, it also consists of two buttons which is Sign in and Register. The data is saved on the database through Firebase. If you are already a member you can log in using your email and password which is used to authenticate your identification, the authentication is done through communication with Firebase which covers the back-end of things.

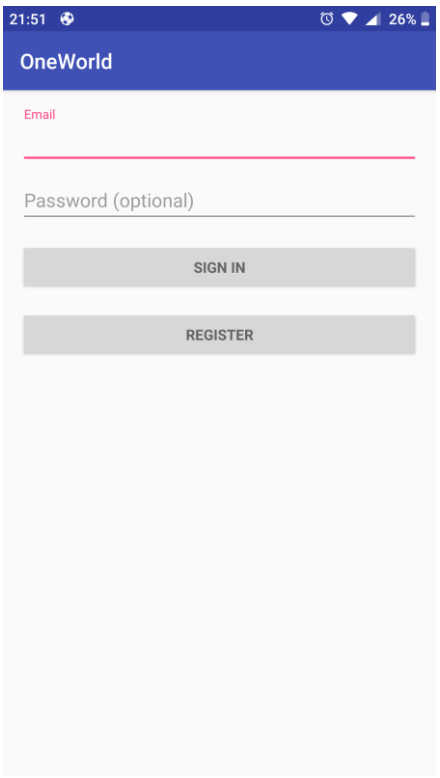


Figure 1. Login Page

4.2 REGISTRATION WINDOW

If you are not a member, you can click on the register button on the main screen which redirects you to another window. This window consists of six text fields such as email, password, retype password, full-name, nationality and phone-number. There is a date field for your date of birth as well as a button for gender. There is also a button at the bottom of the page. Once you have filled in your information and pressed the submit button this information is saved to the database within Firebase. You can use your information for logging in next time.

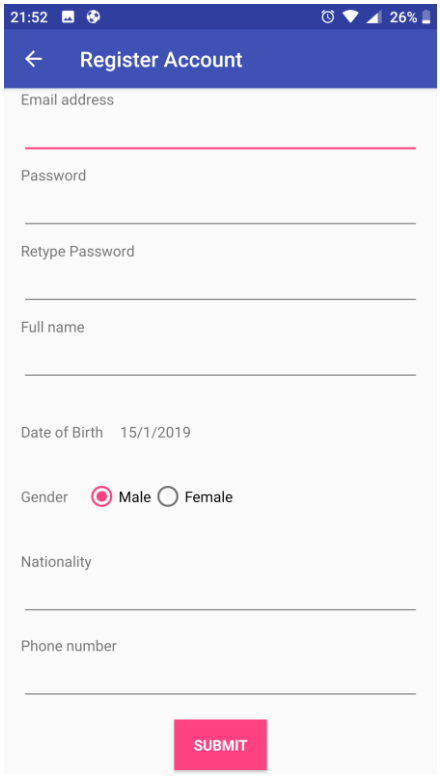


Figure 2. Register Account

4.3 MAIN WINDOW

After you have successfully logged in or registered you move to the main page. When you reach the main page, a request for your location will pop up which is used to show you the volunteer positions in the area you are residing. Once you have accepted that request it will show you the positions open in your area. It also has three tabs at the bottom of the page which are linked through some code. When you click home, you will be redirected to the home page. The create post tab will redirect you to the page where you can create a new volunteering event for a cause you are passionate about.

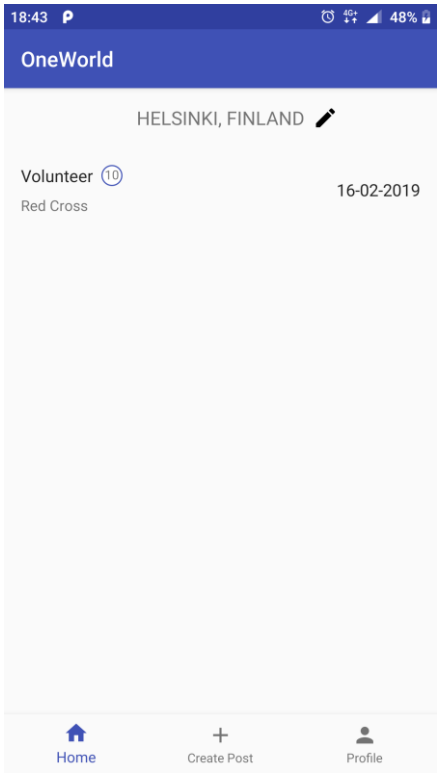


Figure 3. Main Page

4.4 WINDOW

In the profile tab you can find all your personal information. You can also find all your events created and the applications received. There is also a sign out button which will lead you back to the Sign In/ Registration page. The buttons on the bottom will stay on screen whichever tab you are working on and when you are on a specific tab it will be highlighted in blue.

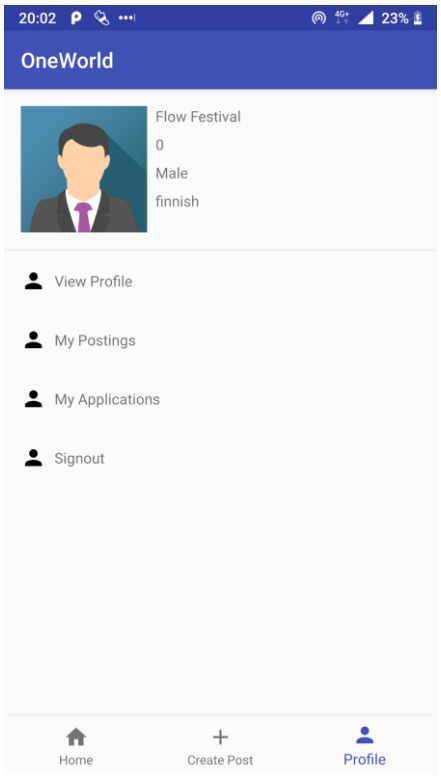


Figure 4. Profile Page

4.5 VIEW PROFILE

In the View profile tab, you can use your user profile, which has the information you registered with.

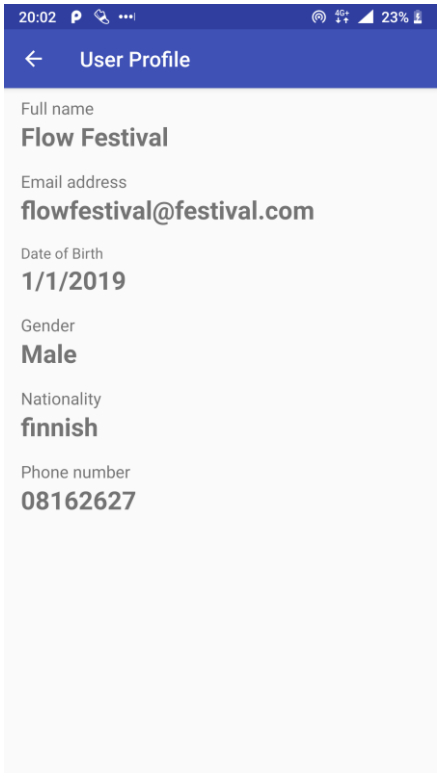


Figure 5. User Profile

4.6 MY POSTINGS

In the “My Postings” page you can see all the positions you have created. Every time you create a new event it will show up here so you can keep track of the events you created in the past. Once you click on a position you created you can see the details of the position as well. From the position details page, you can also view the applications sent to your open position so you can see who has applied and choose who you would want to work for you.

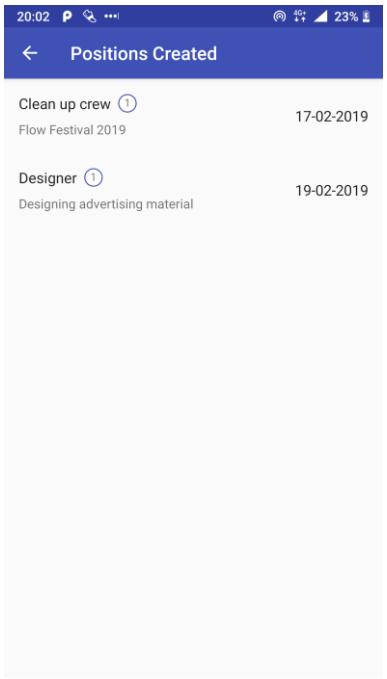


Figure 6. Positions Created

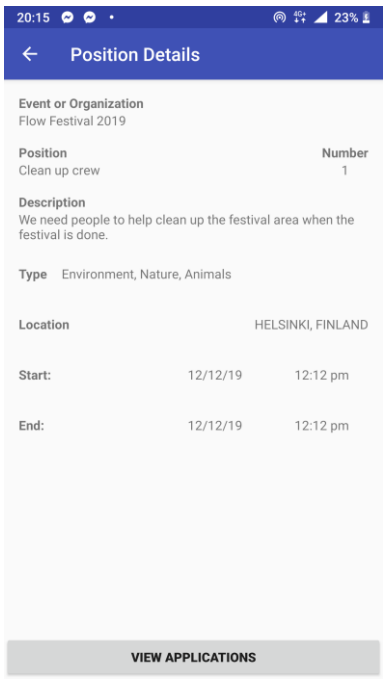


Figure 7. Position Details

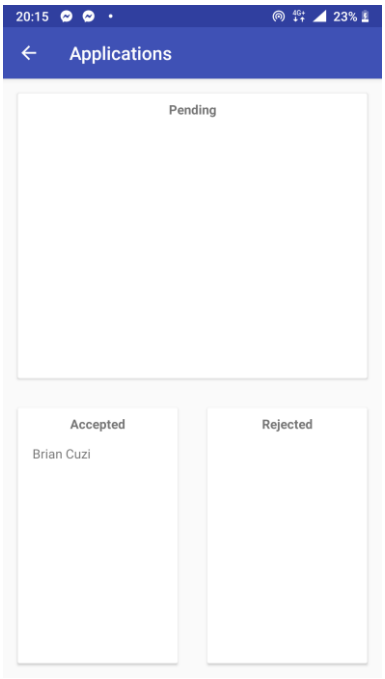


Figure 8. Applications

4.7 CREATE POST WINDOW

The create post tab will be used to post a new volunteering event. When an individual is registered and has a profile on the app, they are able to sign up to different volunteer events as well as creating their own volunteering events about which causes, they are passionate about. You can write the name or organizer of the event as well as add the position name. You can write a description regarding the position you are creating including what the organization does, why the position is important, and lastly what kind of duties it includes. You can also select the type of cause. Finally, you need to select location, start time and end time. This page has three text boxes, a drop-down selection for the type of cause. There is a location tab where you can select where you live so you can find relevant people to help your cause. It also has two sets of date and time pickers for selecting the start and end time of your cause. Once you fill in everything and click create post, the data is sent to the database and is published to the main page where people residing in the area will be able to see it and apply.

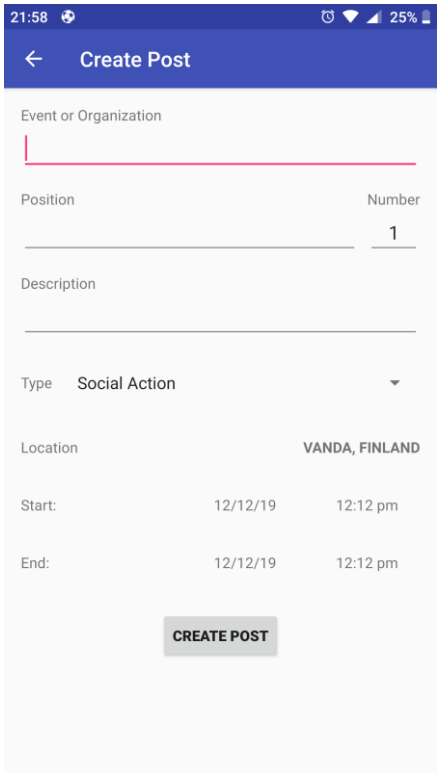


Figure 9. Create Post Page

5 IMPLEMENTATION

In this section the code will be explained in detail. The application was written using Java and the backend was handled using Firebase.

Demonstrated below is how an event is set up and how a person can join an event, it also shows the code behind this process as we go along.

5.1 REGISTER ACCOUNT

In the registration page, the user must fill in some basic information which is required to create the account. Once the user fills in all the fields, they click the submit/register button. It calls the createAccount() function. The code snippet below shows the createAccount() function in more detail. The form fields such as email and password are checked for validation which is not included in the code snippet. But if the validation passes, a User object is created and the Firebase Authentication createUserWithEmailAndPassword() function is called which creates a new account for the user in Firebase.

```
public void createAccount() {
    // perform validation on fields filled by user
    final User user = new User(emailAddress, fullname, dob, gender, nation, phone);

    mAuth.createUserWithEmailAndPassword(emailAddress, confirmedPassword)
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
          @Override
          public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()) {
              // Registration is successful
              saveProfileToDB(user);
            } else {
              // If sign in fails, display a message to the user.
              Toast.makeText(context, "Authentication failed.",
                    Toast.LENGTH_SHORT).show();
              registrationLayout.setVisibility(View.VISIBLE);
              progressLayout.setVisibility(View.GONE);
            }
          }            });   }
```

Code Snippet 1. Create Account Function

The code snippet below shows the saveProfileToDB() function. It saves the user profile information to the Firebase database in the users table. If the profile is successfully saved to the database it takes the user to the home screen and if not, the user gets a pop-up message saying registration has failed.

```
public void saveProfileToDB(User user) {
      FirebaseUser newUser = mAuth.getCurrentUser();
      DatabaseReference mDatabase = FirebaseDatabase.getInstance().getReference();
      mDatabase.child("users").child(newUser.getUid()).setValue(user).addOnCompleteListener(new
OnCompleteListener<Void>() {
         @Override
         public void onComplete(@NonNull Task<Void> task) {
            launchMainActivity();
         }
      }).addOnFailureListener(new OnFailureListener() {
         @Override
         public void onFailure(@NonNull Exception e) {
            Toast.makeText(context, "Registration failed.",
                  Toast.LENGTH_SHORT).show();
            registrationLayout.setVisibility(View.VISIBLE);
            progressLayout.setVisibility(View.GONE);
         }
      });
   }
```

Code Snippet 2. Save Profile to Database Function

5.2 LOGIN PAGE

In the login page, we first initialize the Firebase Authentication using the following snippet of code.

```
// Initialize Firebase Auth
mAuth = FirebaseAuth.getInstance();
```

Code Snippet 3. Initialize Firebase

In the onStart() function, the app checks if there is already a user signed in. If there is, it will automatically redirect to the home screen.

```
@Override
public void onStart() {
      super.onStart();
      // Check if user is signed in (non-null) and update UI accordingly.
      FirebaseUser currentUser = mAuth.getCurrentUser();
      if(currentUser != null)
         launchMainActivity();
}
```

Code Snippet 4. Check User Login Status

If the user is not signed in, they will have to type in their email and password into the login form. The code snippet below shows the attemptLogin() function. Once the user hits submit, the email and password are checked for validation which is not shown in the code snippet below. If the validation passes, the signInWithEmailAndPassword() function from Firebase Authentication is called. If the sign in is successful, the user is taken to the home screen.

```
private void attemptLogin() {
     // Get login details
      if (cancel) {
         // There was an error; don't attempt login and focus the first
         focusView.requestFocus();
      } else {
         // Show a progress spinner, and kick off a background task to
         // perform the user login attempt.
         showProgress(true);
         mAuth.signInWithEmailAndPassword(email, password)
               .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                  @Override
                  public void onComplete(@NonNull Task<AuthResult> task) {
                     if (task.isSuccessful()) {
                        // Sign in success, update UI with the signed-in user's information
                        openHome();
                     } else {
                        // If sign in fails, display a message to the user.
                        showProgress(false);
                        Toast.makeText(context, "Authentication failed.",
                              Toast.LENGTH_SHORT).show();
                     }                  }               });      }   }
```

Code Snippet 5. Login Function

5.3 MAIN PAGE

The code snippet below shows the navigation tabs function and the process behind it. The home screen contains the navigation tab in the bottom of the page. There are three different tabs including the “posts”, “create post” and “profile”.

```
private BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener
        = new BottomNavigationView.OnNavigationItemSelectedListener() {

    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {

        switch (item.getItemId()) {
            case R.id.navigation_home:
                return loadFragment(homeFragment);
            case R.id.navigation_create_post:
                return createPost();
            case R.id.navigation_profile:
                return loadFragment(profileFragment);
        }
        return false;
    }
};
```

Code Snippet 6. Navigation Tab

5.3.1 POSTS PAGE

The first tab shows the list of volunteer positions available based on the location selected. By default, it will get jobs based on the user’s current location. The code snippet below shows how the job positions are retrieved from the Firebase Database.

```
public void getPosts() {
      DatabaseReference postRef = mFirebaseDatabase.getReference("posts");
      postRef.addListenerForSingleValueEvent(new ValueEventListener() {
         @Override
         public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if(dataSnapshot.getValue() != null) {
               postsList.clear();
               for (DataSnapshot postSnapshot: dataSnapshot.getChildren()) {
                  Post post = postSnapshot.getValue(Post.class);
                  post.setId(postSnapshot.getKey());
                  postsList.add(post);
               }
               adapter.notifyDataSetChanged();
            } else {
               postsList.clear();
               adapter.notifyDataSetChanged();
            }
         }

         @Override
         public void onCancelled(@NonNull DatabaseError databaseError) {
            postsList.clear();
            adapter.notifyDataSetChanged();
         }
      });
   }
```

Code Snippet 7. Open Volunteering Positions

5.4 CREATE POST

This code snippet shows the createPost() function which is called when the user clicks on the create post button. When all the fields are filled in, there is a list of validation checks that runs to make sure all the data is valid. If the data is valid, a Post object is created. And this post object is saved in the posts table in the Firebase database.

```
   public void createPost() {
      //perform form validations
     if(checkPassed) {

        final Post post = new Post(titleInput.getText().toString(), positionInput.getText().toString(),
              descInput.getText().toString(), Integer.parseInt(numberInput.getText().toString()),
              typeChosen, postLocation, startTime.getTimeInMillis(), endTime.getTimeInMillis(),
              mAuth.getUid());

        final DatabaseReference mDatabase = FirebaseDatabase.getInstance().getReference();
        final String postID = mDatabase.child("posts").push().getKey();
        mDatabase.child("posts/" + postID).setValue(post).addOnCompleteListener(new
OnCompleteListener<Void>() {
           @Override
           public void onComplete(@NonNull Task<Void> task) {
              Map<String, Object> timestamp = new HashMap<>();
              timestamp.put("timestamp", ServerValue.TIMESTAMP);
              mDatabase.child("posts/" +
postID).updateChildren(timestamp).addOnCompleteListener(new OnCompleteListener<Void>() {
                 @Override
                 public void onComplete(@NonNull Task<Void> task) {
                    finish();
                 }
              });
           }
        }).addOnFailureListener(new OnFailureListener() {
           @Override
           public void onFailure(@NonNull Exception e) {
              Toast.makeText(context, "Post failed.",
                    Toast.LENGTH_SHORT).show();
           }
        });      }   }
```

Code Snippet 8. Create a New Position

5.5 PROFILE

The profile shows the basic user information which is fetched from Firebase database. The user profile is saved in the user table and the code snippet below shows how the data is fetched.

```
mDatabase.child("users/" + mAuth.getUid()).addListenerForSingleValueEvent(new ValueEventListener()
{
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
          if(dataSnapshot.exists()) {
            User user = dataSnapshot.getValue(User.class);
            fullNameTV.setText(user.full_name);

            Calendar dob = Calendar.getInstance();
            dob.set(user.dob.get("year"), user.dob.get("month"), user.dob.get("day"));
            Date birthDate = dob.getTime();

            String age = String.valueOf(getAge(currentDate, birthDate));
            ageTV.setText(String.valueOf(age));

            if(user.gender == 0) {
              genderTV.setText(getString(R.string.signup_gender_female));
            }

            nationalityTV.setText(user.nationality);
            ratingTV.setText(String.valueOf(user.rating));
          }
        }
```

Code Snippet 9. Profile Request from Firebase

The "Signout" button logs out the user and takes him back to the sign in page. The code snippet below shows how the sign out button works.

```
ConstraintLayout signout = view.findViewById(R.id.profile_signout);
      signout.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
          mAuth.signOut();
          Intent i = new Intent(getActivity(), LoginActivity.class);
          getActivity().startActivity(i);
          getActivity().finish();          }       });
```

Code Snippet 10. Sign out Button

5.5.1 USER PROFILE

On this page users can view their own profiles, The information they filled in during registration. The code snippet below shows how the data is fetched form Firebase.

```
 database.getReference("users").child(userId).addListenerForSingleValueEvent(new
ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
           if(dataSnapshot.getValue() != null) {
              User user = dataSnapshot.getValue(User.class);
              fullName.setText(user.full_name);
              email.setText(user.email);
              String date = String.format("%d/%d/%d",user.dob.get("day"),user.dob.get("month") + 1,
user.dob.get("year"));
              dob.setText(date);
              if(user.gender == 0) {
                 gender.setText(getString(R.string.signup_gender_female));
              } else {
                 gender.setText(getString(R.string.signup_gender_male));
              }
              nationality.setText(user.nationality);
              phoneNumber.setText(user.phone);
           }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {
        }
      });
   }
}
```

Code Snippet 11. User Profile Data from Firebase

5.5.2 MY POSTINGS

On this page a user can see the volunteer positions created by him. The code snippet below shows how the list of volunteer positions you created are fetched from the database.

```
public void getMyPosts() {
      Query postRef = mFirebaseDatabase.getReference("posts")
            .orderByChild("creator").equalTo(user.getUid());
      postRef.addListenerForSingleValueEvent(new ValueEventListener() {
         @Override
         public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if(dataSnapshot.getValue() != null) {
               postsList.clear();
               for (DataSnapshot postSnapshot: dataSnapshot.getChildren()) {
                  Post post = postSnapshot.getValue(Post.class);
                  post.setId(postSnapshot.getKey());
                  postsList.add(post);
               }
               adapter.notifyDataSetChanged();
            } else {
               postsList.clear();
               adapter.notifyDataSetChanged();
            }
         }

         @Override
         public void onCancelled(@NonNull DatabaseError databaseError) {
            postsList.clear();
            adapter.notifyDataSetChanged();
         }
      });
   }
```

Code Snippet 12. Fetching Volunteer Positions Posted by User

5.5.3 MY APPLICATIONS

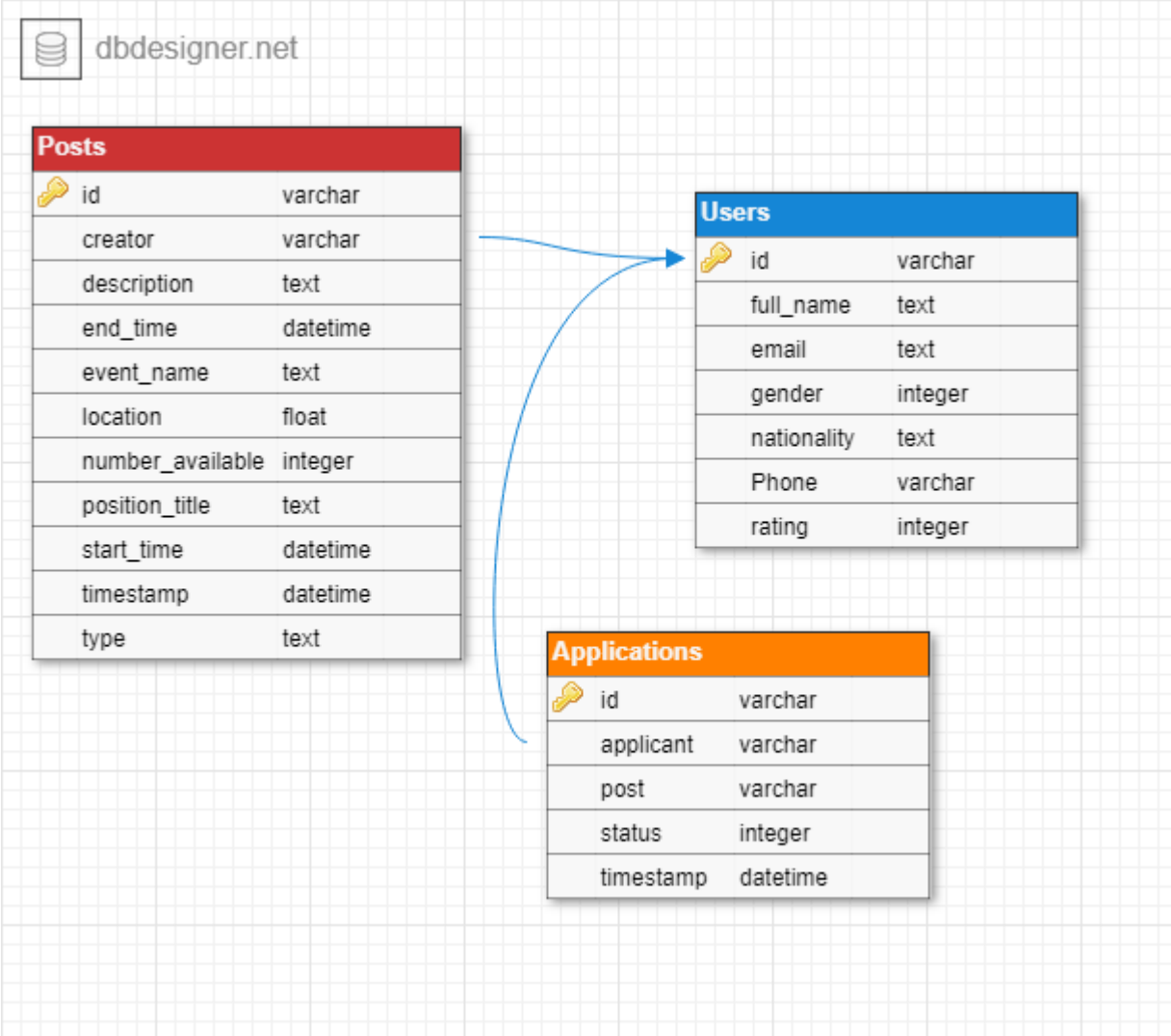
On this page a user can see the volunteering positions you have applied for. The code snippet below shows how the list of your applications is fetched from the database.

```
public void getMyApplications() {
      Query postRef = mFirebaseDatabase.getReference("applications")
            .orderByChild("applicant").equalTo(user.getUid());
      postRef.addListenerForSingleValueEvent(new ValueEventListener() {
         @Override
         public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if(dataSnapshot.getValue() != null) {
               postsList.clear();
               for (DataSnapshot postSnapshot: dataSnapshot.getChildren()) {
                  Application application = postSnapshot.getValue(Application.class);
                  Log.d(TAG, "Post found: " + application.post);
                  mFirebaseDatabase.getReference("/posts/" +
application.post).addListenerForSingleValueEvent(new ValueEventListener() {
                     @Override
                     public void onDataChange(@NonNull DataSnapshot postSnapshot) {
                        Post post = postSnapshot.getValue(Post.class);
                        post.setId(postSnapshot.getKey());
                        postsList.add(post);
                        adapter.notifyDataSetChanged();
                     }
                     @Override
                     public void onCancelled(@NonNull DatabaseError databaseError) {}
                  });
               }
            } else {
               postsList.clear();
               adapter.notifyDataSetChanged();
            }
         }
      });
   }
```

Code Snippet 13. Volunteer Positions Applied to by User

6 DATABASE

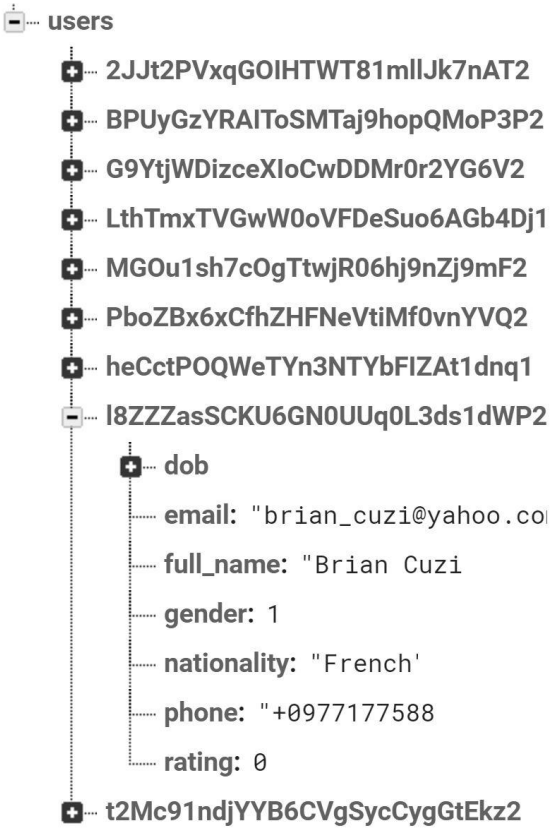
The database for the OneWorld application is handled by Firebase. Firebase is a Google service which provides functionality such as analytics, databases, messaging and authentication as a service. For this application I used the authentication and database features. Below the database scheme and the relationships can be seen.



Here is how my database looks within Firebase. I have three tables Applications, Posts and Users.

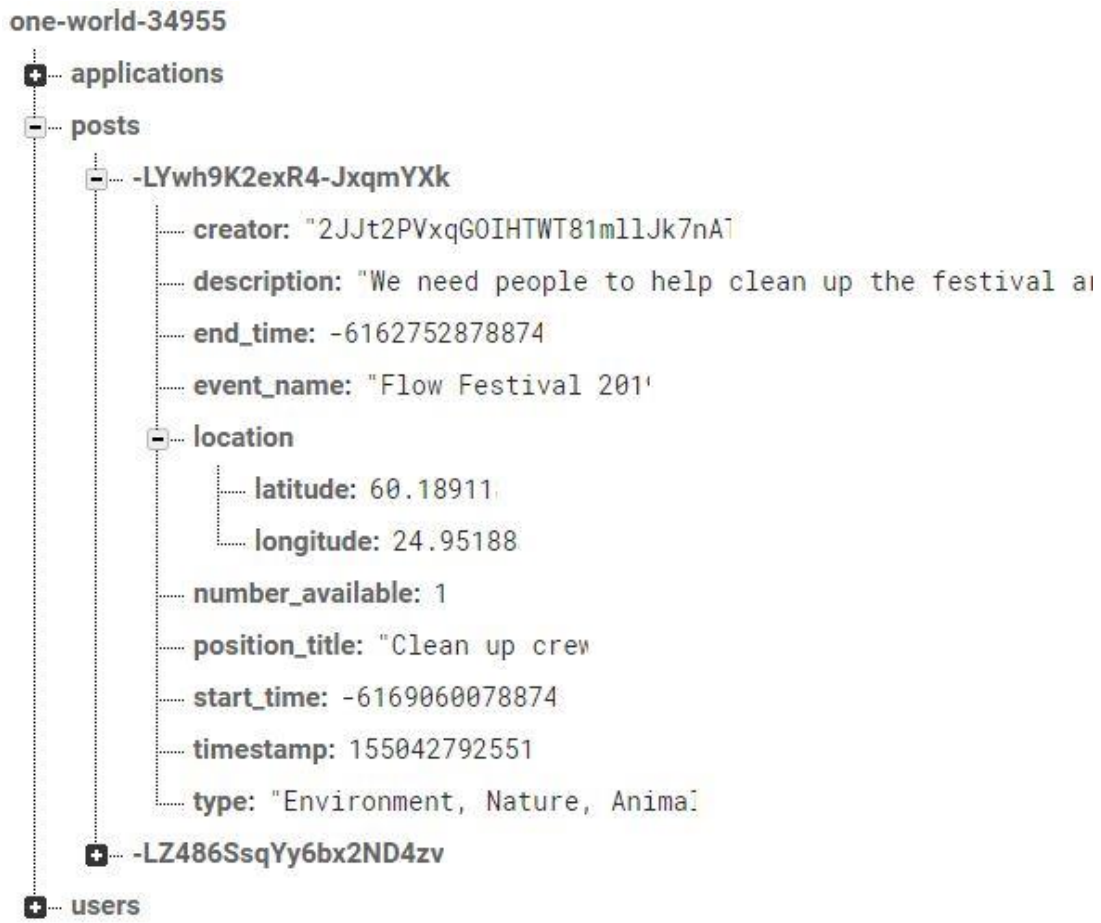
6.1 USERS

The table users hold all the information of the users who registered to the application. Each user is saved with a unique ID. In this cause the user we are looking at “Flow Festival” which has an ID of “2JJt2PVxqGOIHTWT81mllJk7nAT2”. The table includes six fields which are email, full name, gender, nationality, phone and rating. The gender field works that if you choose male as your gender it shows one and if you choose female it is entered as a two. Rating was something that could be implemented in the future, it is when a user has completed his volunteering position the one who created can give him/her a rating.



6.2 POSTS

The table posts hold all the different volunteering positions created by users. Each position is saved with a unique ID which in this case is “LYwh9K2exR4-JxqmYXK”. For this case I am showing a volunteering position created by the user “Flow Festival”. The first field is the creator which is linked to the unique ID of the user who created the position. Other fields include event name, position title, description, start time, end time, number of positions, type and location. The location field takes the coordinates from the users current position. The timestamp field just saves the time when the data was written to the database.



6.3 APPLICATIONS

The table applications hold the information of an applicant who applied for a specific position. Each application is saved with a unique ID which in this case is “LYxM7W-dgJrsXpG88TO”. For this case it shows the applicant with ID “18ZZZasSCKU6GNUUq0L3ds1dWP2” who is called “Brian Cuzi” and he is applying for the position posted by “Flow Festival” which has an ID “LYwh9K2exR4-JxqmYXK”. It has the fields applicant, post, status and timestamp. The field applicant and post are linked with the user table and post table accordingly. The field status refers to the status of the volunteering application. It has three states, when your application is pending its assigned a minus one and if it is rejected it is assigned a zero and if it is accepted it is assigned a one. In this case the application was accepted. The timestamp field just saves the time when the data was written to the database.

7 CONCLUSION

The main objective was to develop an android mobile application which was supposed to make the whole volunteering process easier. The OneWorld Android Application allows anyone with a passion for volunteering straight access to organizations. A user can create a profile and sign up for different volunteering positions available at the same time an organization or a normal user can create an event and choose the number of volunteers needed which is posted on the applications home page. A user can then see it and apply for that position.

The application was developed in Android Studio and it was written in Java. It was a challenge for me personally since I did not have a good understanding of code, but it has been a learning process.

Firebase was a great help during my application development process as it made the whole process simpler. It helps build applications faster by providing ready made functionalities such as analytics, databases, authentication and cloud storage.

As for the improvements, more functionally could be added later to make it more user-friendly and robust for organizations to use. Additionally, a rating system could be implemented, so organizations or users who create events have volunteers who worked for them and can give them a rating based on their performance during the event.

As a conclusion to this research paper, it is appropriate to say the research objective was met by providing a basic mobile application concept for making volunteering easier. Additionally, the thesis addressed not just the conceptual aspect of a volunteering application but also the technical side.

8 REFERENCES

/1/ Elev8society Application. Accessed 22.10.2018

http://elev8society.com/

/2/ CHIP'N Application. Accessed 22.10.2018

http://ichipn.com

/3/ GiveGab Application. Accessed 22.10.2018

https://www.givegab.com

/4/ DEED Application. Accessed 22.10.2018

http://godeed.today/

/5/ Features of Android. Accessed 24.10.2018

https://www.elprocus.com/what-is-android-introduction-features-applications/

/6/ Android OS. Accessed 24.10.2018

https://www.techopedia.com/definition/14873/android-os

/7/ Android Runtime. Accessed 24.10.2018

https://en.wikipedia.org/wiki/Android_Runtime

/8/ What is Java programming language. Accessed 24.10.2018

https://howtodoinjava.com/java/basics/what-is-java-programming-language/

/9/ Introduction to Java programming language. Accessed 24.10.2018

https://dzone.com/articles/java-tutorial-to-learn-java-programming

/10/ Java programming language. Accessed 25.10.2018

https://en.wikipedia.org/wiki/Java_(programming_language)

/11/ Learn more about the Android Studio IDE from Google. Accessed 25.10.2018

https://searchsoftwarequality.techtarget.com/feature/Learn-more-about-the-Android-Studio-IDE-from-Google

/12/ Android Studio user guide. Accessed 25.10.2018

https://developer.android.com/studio/intro/

/13/ What is Firebase. Accessed 26.10.2018

https://blog.usejournal.com/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0

/14/ Firebase features. Accessed 26.10.2018

https://firebase.google.com/?gclid=Cj0KCQiA8f_eBRDcARIsAEKwRGec-aHWRLzjDz7hB__4GrkdHDH0iecSYch3zUsKiD8TrreS6WP3CjHIaAo9LEALw_wcB