National College of Ireland
BSc in Computing
2016/2017

Franciele Firmino
X13111787

Mobile Application for Drogheda Animal Rescue

Technical Report

1 Contents

Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name: ***Franciele Firmino***
Student ID: ***x13111787***
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: ____ Date: ____________

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.

2 EXECUTIVE SUMMARY

This document explains the process of developing an application for Drogheda Animal Rescue, a non-profit organization focused on the rehoming of cats and dogs. The application was developed for Android devices, with both an administrator and general user section, it allows users to view animals available for adoption, check for Lost & Found pets in the area, apply for volunteering and donate to the organization using a PayPal account or credit card. In the admin section, administrators can communication using the group chat, a feature powered by Firebase. The application has proven to help the organization on its day to day tasks.

3 INTRODUCTION

3.1 BACKGROUND

For this project Drogheda Animal Rescue was contacted, DAR is a non-profit, voluntary organisation committed to rehoming both dogs & cats. A mobile application would be developed to benefit the organization. The main functions for the app were agreed together with DAR's volunteer committee, and freedom was given to add extra functionality, if required.

3.2 AIMS

The aim of this project was to develop a mobile application, that would be useful for DAR (Drogheda Animal Rescue). It would help them to organize themselves internally, keep track of pets in their care, display dogs and cats available for adoption and lost & found pets. It would serve as an extra tool in addition to their Facebook page and website.

3.3 TECHNOLOGIES

The mobile application was developed in Android Studio, and it is powered by Firebase, a free mobile platform that takes care of the push notifications, stores the application data as photos, user details and it also allows to remotely modify parts of the application by using the user-friendly Firebase Console. To accept payments, PayPal Rest API has been used, the API provides a secure way to perform card payments and allow users to conveniently pay using their PayPal accounts. An interesting new feature of PayPal API is that it allows the user to scan their debit/credit card using their camera.

3.4 STRUCTURE

Below you will find detailed information about the application, in the System section you will find details about the app's functionality, architecture, designing approach and how the application was tested and evaluated.

3.4.1 Vocabulary

Clarification on terminology/words used in this report.
DAR = Drogheda Animal Rescue
Lost Pet = Someone's pet that is missing
Found Pet = Animal found roaming streets, owner not known.
General users = users with no administrative credentials
Admin users = app administrators, organization volunteers

4 SYSTEM

4.1 REQUIREMENTS

As DAR is a charity focused on rehoming of cats and dogs, the main section of the application is the one that allows the public to see animals that are available for adoption and submit their details for a viewing. The second one is part that allows people to sign up as volunteers and donate.

The objective of this application is to make it easier for people to browse for animals available for adoption and serve as a tool that makes the volunteers' job easier by having a section where volunteers can communicate with each other. It also includes a Lost & Found section were all users can actively participate by adding lost pets and announcing missing pets. The application must be divided in two sections Admin and general users.

4.1.1 Functional requirements

The user requirements were discussed with DAR committee, we agreed that the main functions would be to allow the general users to see animals available for adoption and book an appointment to view them, also as the organization is run by volunteers to allow users to sign up to be volunteers in different areas within DAR is also important. As the staff is very busy with daily vets and foster homes drop offs and picks ups, the app will allow admin users to communicate using the app to keep all communication in one place.

The general users can also report and search lost & found pets, to report a lost dog the user can upload a picture of the animal and enter the pet details. To report a pet that was found roaming the streets, the user takes a picture of the dog and can enter a location or give his current location (Uses GPS), those details are them added to the database, allowing the owner of the pet to be found. Donations are also vital for the organization so users can donate using their PayPal account or credit card.

4.1.2 Use Case Diagram

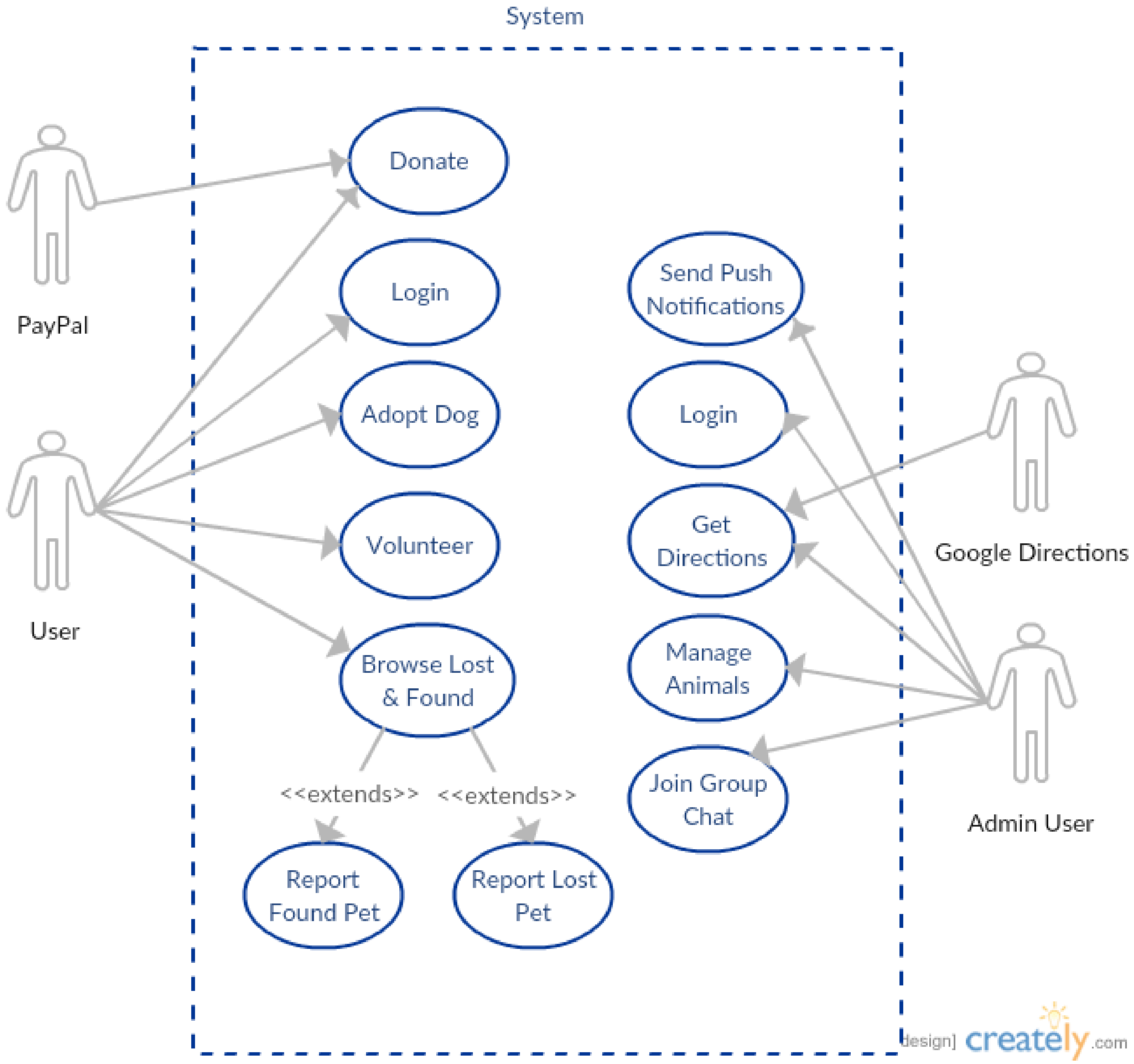


Figure 4–1 Use Case Diagram

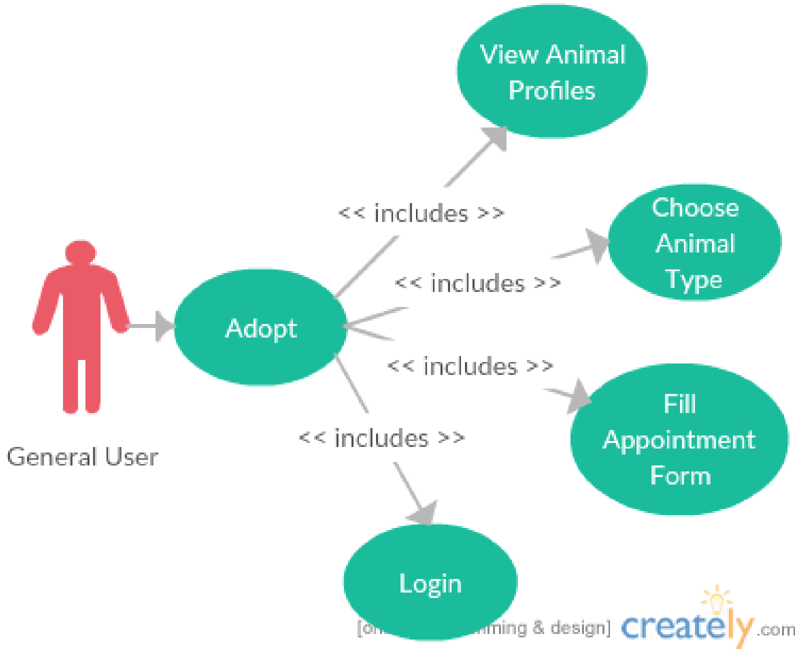
Requirement 1: Adopt

User must be logged in to view cats and dogs available for adoption. The user can then register their interest to adopt an animal by filling a form with their details.

Description & Priority

The user must be able to view animal for adoption as it is one of the main goals of the organization, to rehome animals.

Use Case



Scope

The scope of this use case is to provide the user access to the animal's profiles per their preference.

Description

The use case describes the process of displaying animals' profile based on the user's preference.

Flow Description

Precondition

User credentials are verified.

Activation

Use case starts when the user clicks on the "Adoption" button.

Main Flow

1. User selects an option:
   a. Cats for Adoption
   b. Dogs for Adoption

2. Request is sent to server (See E1)
3. Response is sent back to the user (See A1)
4. User view animals' list
5. User view animal profile
6. User make an appointment to view animal

A1: Server Time Out

1. The user awaits response from the server
2. The app will produce a timeout after 1 minute
3. The user will attempt main flow 1

E1: No Internet Connection

1. The user's device has no internet connection to make a successful request
2. The app produces a message to ask the user to resolve the issue
3. User starts at main flow 1

Termination

The use case terminates when the user makes an appointment to view animal

Post Condition

The app awaits another request from the user

Requirement 2: Send Push Notifications

Admin user can use the Admin Console to send push notifications about upcoming events or important alerts

Description & Priority

The organization requires to send notifications in case of lost pet or upcoming fundraising events.

Use Case

Scope

The scope of this use case is to provide push notifications to the user

Description

The use case describes the process of receiving push notifications and how they are set up by the admin users.

Flow Description

Precondition

The admin user must be logged in to Firebase Console.

Activation

Use case starts when the user accepts to receive notifications and the admin user sets up a push notification on Admin Console.

Main Flow

1. User is subscribed to push alerts
2. App sends subscription service request to the server (See A1)
3. Admin user sets up push notification on the Console
4. Server sends service response to push alert engine
5. User receives push alert

A1: Server Time Out

1. The user awaits response from the server
2. The app will produce a timeout after 1 minute
3. The app will attempt main flow 3

Termination

User receives push alert on their device

Post Condition

The app awaits another request from the server

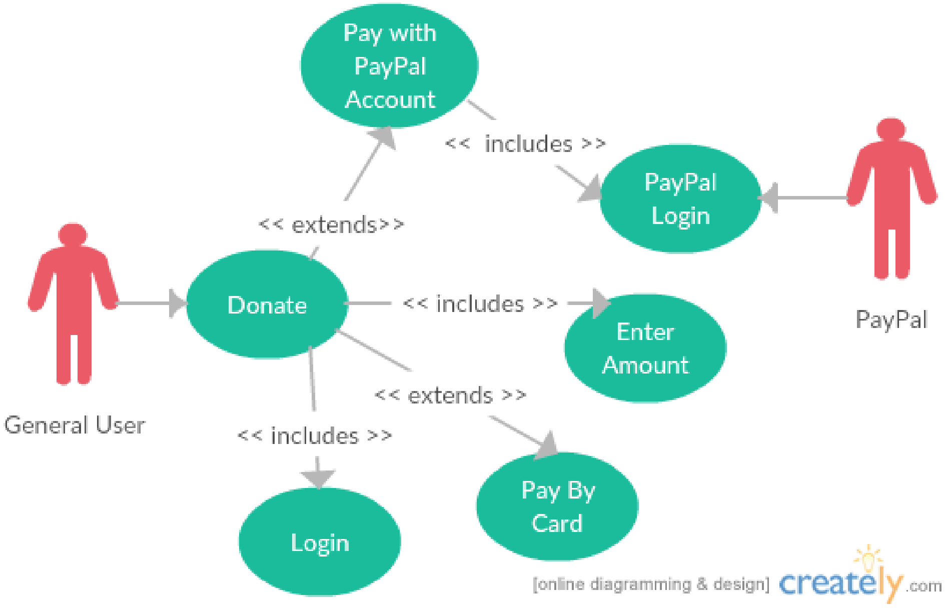
Requirement 3: Donate

User must be logged in to donate. The user can donate by scanning their credit card or using their PayPal account.

Description & Priority

The user must be able to donate as it is a vital for DAR to cover their daily expenses.

Use Case



Scope

The scope of this use case is to allow the user to donate to DAR.

Description

The use case describes the process of donating using a card or PayPal account.

Flow Description

Precondition

User credentials are verified.

Activation

Use case starts when the user clicks on the "Donate" button.

Main Flow

1. User selects an option:
    - **a.** Pay with PayPal
    - **b.** Pay with Card
2. PayPal Activity is open
3. The app takes the user payment details
4. Server is sent the payment details
5. Response is sent back to the user (See A1) (See A2)
6. The app prompts the user that donation was received

A1: Server Time Out

1. The user awaits response from the server
2. The app will produce a timeout after 1 minute
3. The user will attempt main flow 1

A2: Wrong Payment Details

1. The server sends a response
2. The app prompts the user to re-enter payment details
3. The user will attempt main flow 1

E1: No Internet Connection

1. The user's device has no internet connection to make a successful request
2. The app produces a message to ask the user to resolve the issue
3. User starts at main flow 1

Termination

The use case terminates when the user successfully donates to DAR.

Post Condition

The app awaits another request from the user

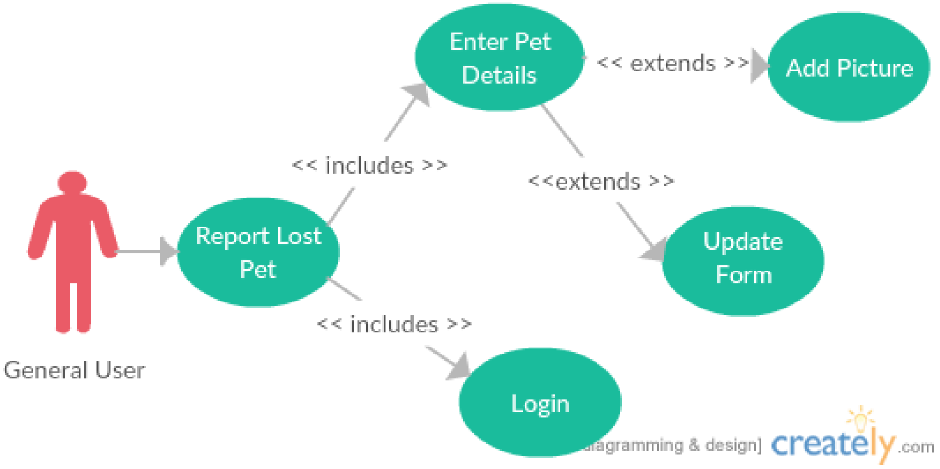
Requirement 1: Report Lost Animal

User must be logged in to report a lost dog or cat. The user can then register the animal in the Lost & Found section by adding the animals' details and adding a picture to it.

Description & Priority

The user must be able to add animals to the Lost & Found section as it is an important part of DAR services.

Use Case



Scope

The scope of this use case is to allow the user access to add an animal to the Lost & Found section of the app.

Description

The use case describes the process of reporting a lost animal.

Flow Description

Precondition

User credentials are verified.

Activation

Use case starts when the user clicks on the "Lost & Found" button.

Main Flow

1. User selects an option:
   a. Found
   b. Lost
2. Request is sent to server (See E1)
3. Response is sent back to the user (See A1)
4. The app displays Lost & Found list
5. User clicks on "Add" button
6. User fills the form with the animals' details
7. Form is sent to server. (See A2)
8. Response is received from the server.
9. The Lost & Found list is updated and displayed.

A1: Server Time Out

1. The user awaits response from the server
2. The app will produce a timeout after 1 minute
3. The user will attempt main flow 1

A2: Form is incorrectly filled

1. The user is prompt to fill missing fields.
2. The user will attempt main flow 6

E1: Exceptional Flow

1. The user's device has no internet connection to make a successful request
2. The app produces a message to ask the user to resolve the issue
3. User starts at main flow 1

Termination

The use case terminates when the report is successfully received.

Post Condition

The app brings the user to the Lost & Found list.

4.1.3 Data requirements

One of the administrative tasks in an Animal Rescue Centre is to keep record of the animals in their care. The animals must be corrected stored to allow for easy search and display. All animals should have: bio, status (available, reserved), profile picture and other details as gender, breed, age and neutering status. For both admin and general users, we store their name, password, profile picture and email that are used to identify users when they first log in, depending on their credentials they will have access to different areas of the app.

4.1.4 Usability requirements

The application shall be designed in a way to make it easy to see where to enter the data, know what needs to go in each field (explanatory text) and have fields for all the data they need to enter. It includes a help section with information about how to use the different sections in the application.

4.1.5 Maintainability requirement

DAR's committee shall receive training on how to maintain and updated the application. By accessing Firebase Admin Console, they can modify the database contents. The code is saved on the developers private GitHub account and a backup will be done to the cloud.

4.1.6 Security

Only logged in users shall be able to access the application, to avoid users with malicious intentions to have access to the payment section of the app or order sensitive data. Passwords shall be encrypted and saved in Firebase Authentication Database and in case of password change a confirmation email must be sent to the user. Email verification shall be requested upon registration.

4.1.7 Aesthetics

The GUI shall be clutter free and easy to use, all buttons in the system shall adhere to establish button conventions.

4.1.8 Access

There will be two group of users, admin and general users. The Admin group shall have access to all animals' data and administrative sections, while the general users group shall be given access to only certain areas of the application.

4.1.9 Portability

The system shall be compatible with the most Android devices.

4.2 Design and Architecture

4.2.1 Class Diagram

The classes in the app are distributed into different packages, there are 8 additional packages besides the main one, each corresponding to a specific function in the application. The BaseActivity is the one that contains methods that are reused between classes. All classes extend the BaseActivity.

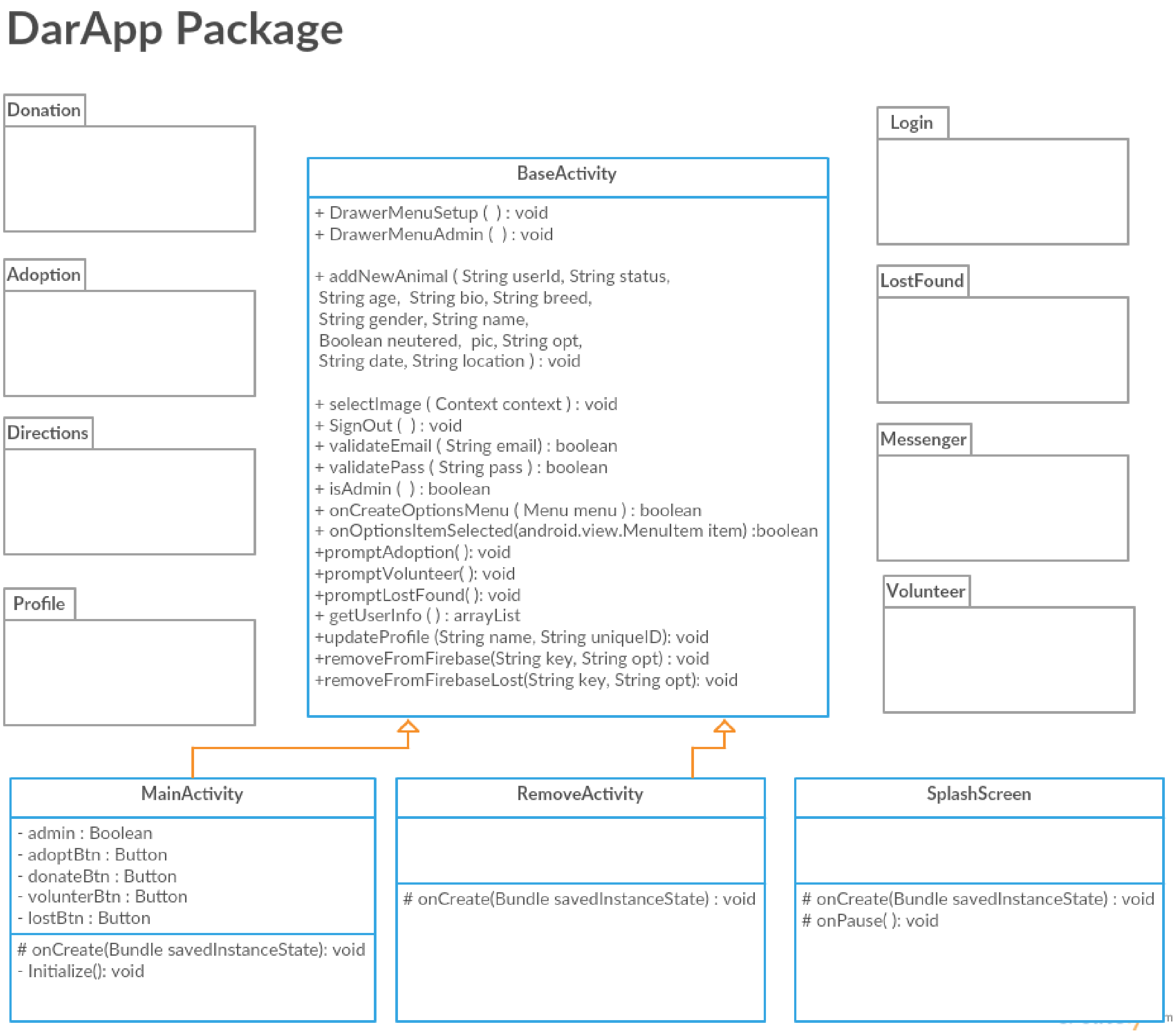

DarApp Package
Donation
Adoption
Directions
Profile
Login
LostFound
Messenger
Volunteer
BaseActivity
+ DrawerMenuSetup ( ) : void
+ DrawerMenuAdmin ( ) : void
+ addNewAnimal (String userId, String status, String age, String bio, String breed, String gender, String name, Boolean neutered, pic, String opt, String date, String location) : void
+ selectImage (Context context) : void
+ SignOut ( ) : void
+ validateEmail (String email) : boolean
+ validatePass (String pass) : boolean
+ isAdmin ( ) : boolean
+ onCreateOptionsMenu (Menu menu) : boolean
+ onOptionsItemSelected(android.view.MenuItem item) :boolean
+promptAdoption( ): void
+promptVolunteer( ): void
+promptLostFound( ): void
+ getUserInfo ( ) : arrayList
+updateProfile (String name, String uniqueID): void
+removeFromFirebase(String key, String opt) : void
+removeFromFirebaseLost(String key, String opt): void
MainActivity
- admin : Boolean
- adoptBtn : Button
- donateBtn : Button
- volunterBtn : Button
- lostBtn : Button
# onCreate(Bundle savedInstanceState): void
- Initialize(): void
RemoveActivity
# onCreate(Bundle savedInstanceState) : void
SplashScreen
# onCreate(Bundle savedInstanceState) : void
# onPause( ): void

Figure 4–2 Class Diagram

4.2.2 System Architecture Diagram

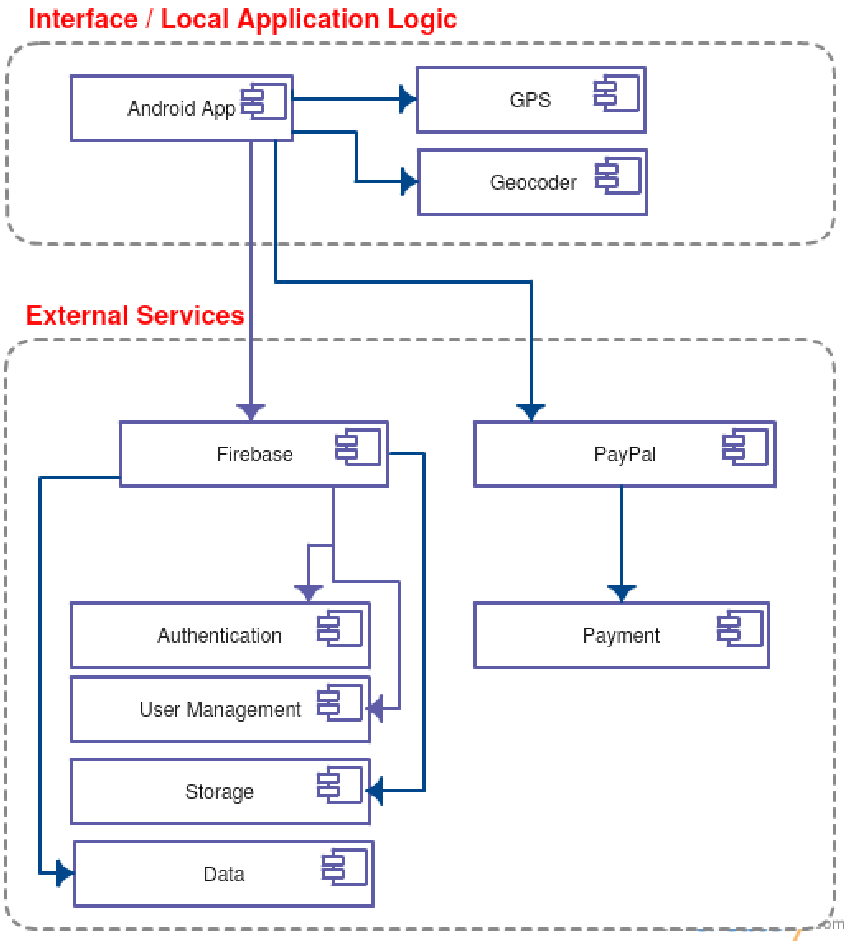


Figure 4–3 System Architecture Diagram

4.3 Graphical User Interface (GUI) Layout

The application has a clear design; it is user friendly. It has large and clear pictures on the animals' profile and it is visually pleasing. As much as possible there are alert messages explaining step by step how to use the application, although navigation is very intuitive. The application theme is called "AppTheme.DAR" based on the "AppCompat.Light.DarkActionBar", the colours for the theme were chosen based on DAR's logo. Theme main colours:

Primary Colour	▬	#63637b
Primary Dark	▬	#1b193f
Accent	▬	#c45c03

DAR's logo

For the application side menu, an external library developed by Mike Penz (github.com/mikepenz, 2017) was used, this drawer menu was easy to set up and was easily added to all activities. See screenshots below of the admin and general user's drawer menu.

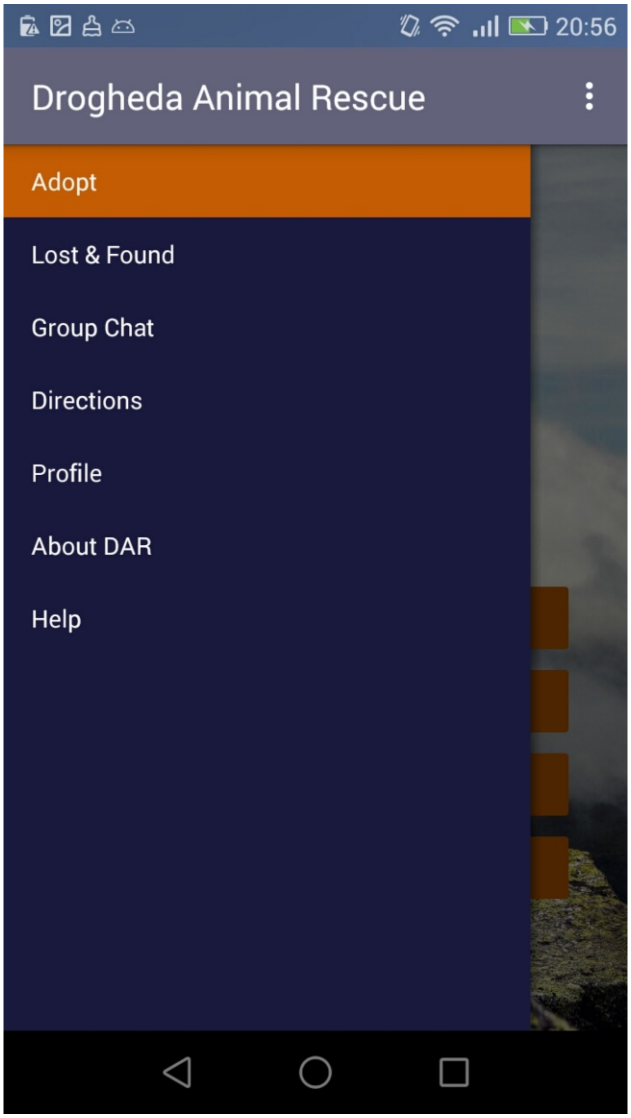


Figure 4–4 Admin User Menu

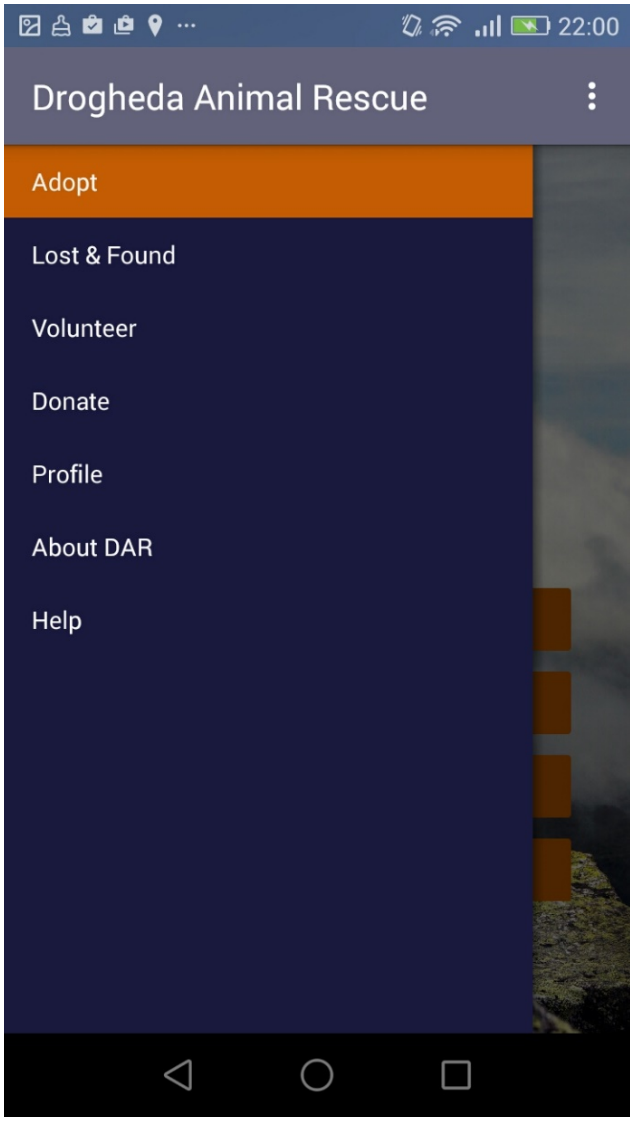


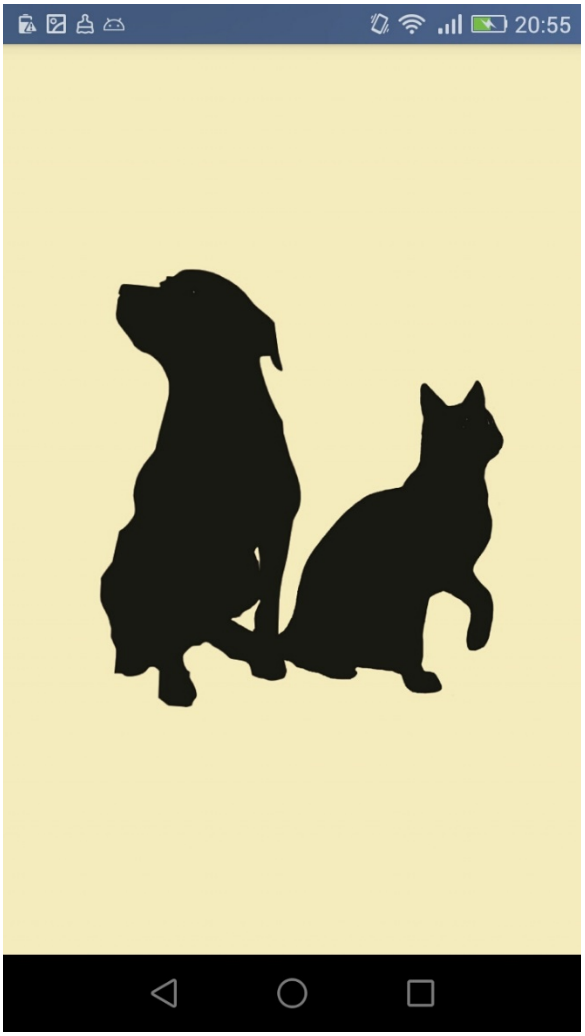
Figure 4–5 General User Menu

4.3.1 Photos & Images

The photos used in this application are high resolution, copyright free, selected from pexels.com. They were resized on Adobe Illustrator and colour corrected on Adobe Lightroom. See preview of main Images below.

Figure 4–6 Background Images

4.3.2 Description of the app screens



This is the splash screen which was designed using Adobe Illustrator and it aims to give the user an idea of the application purpose

Figure 4–7 Splash Screen

.

This is the main screen which the user sees after login. The background for it contains elements of DAR's logo added to it. Buttons were added for only the most used functions in the app to avoid cluttering.

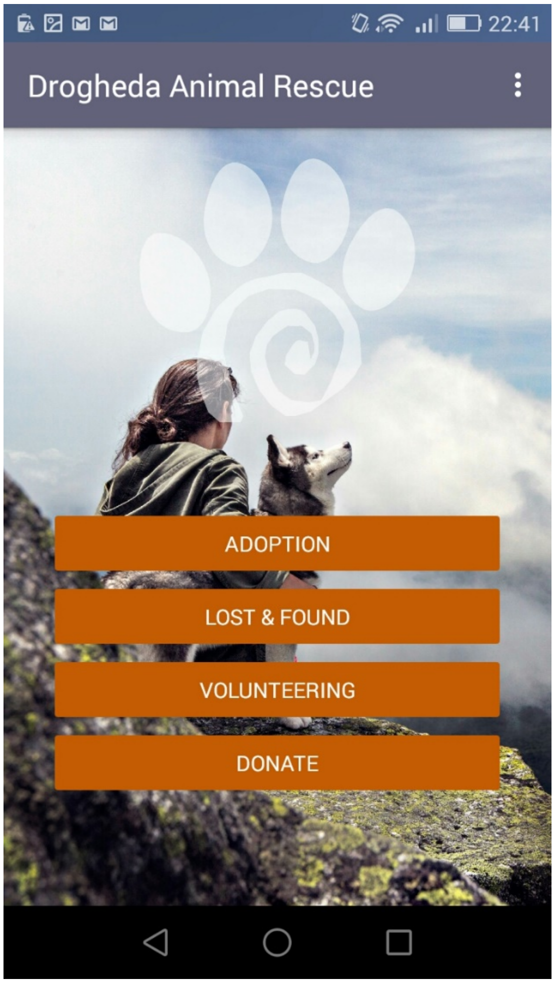
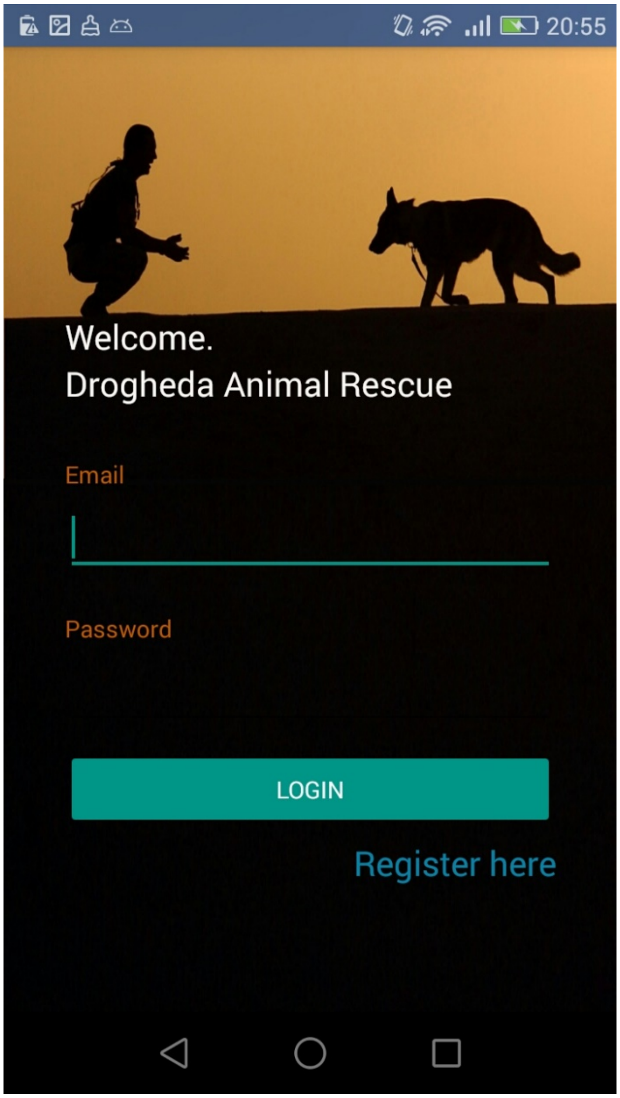


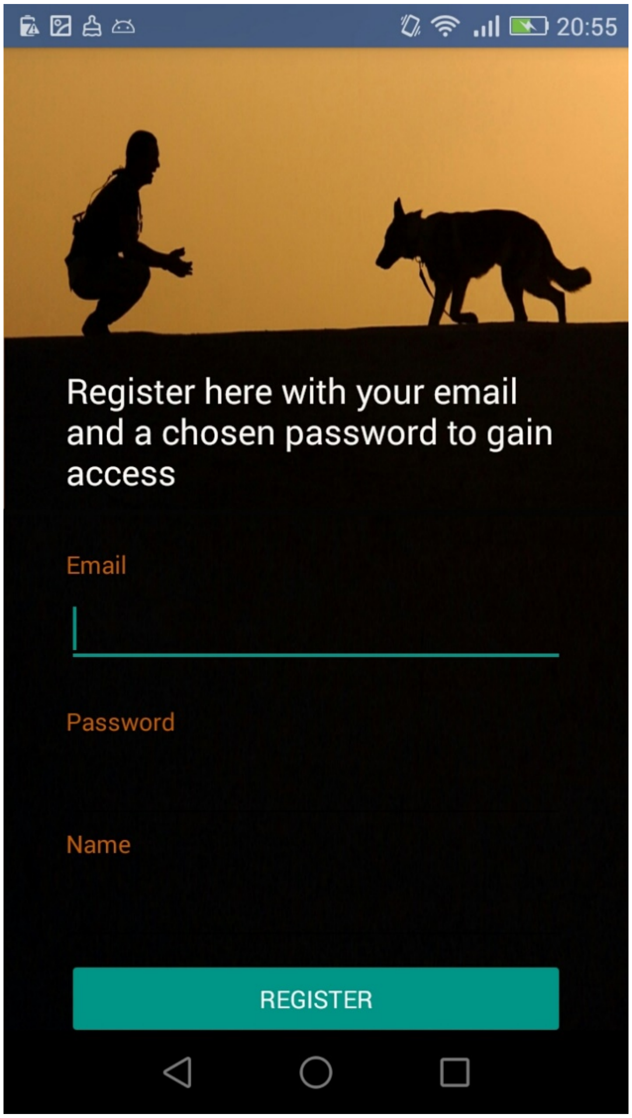
Figure 4–8 Main Screen



Welcome.
Drogheda Animal Rescue

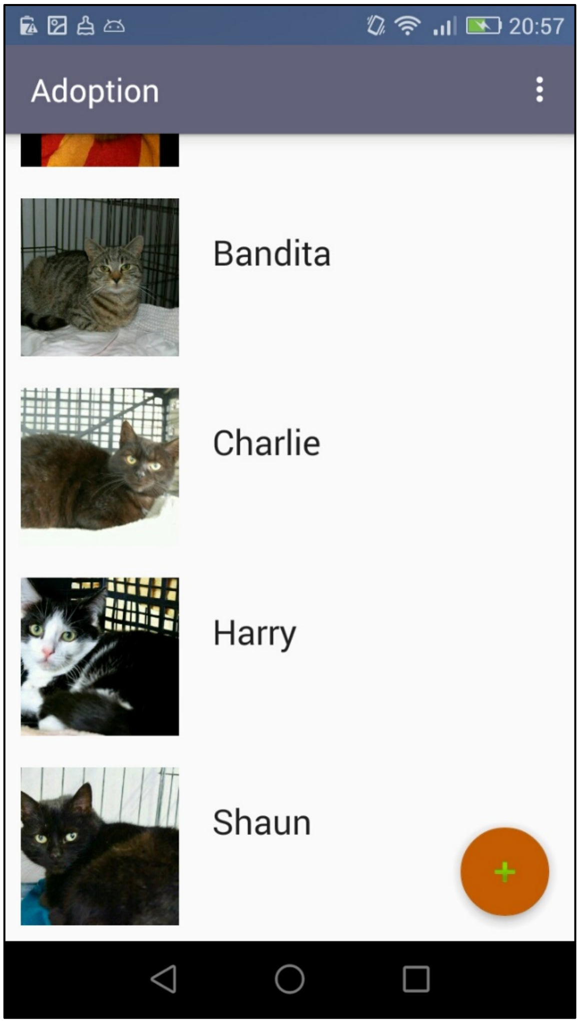
Figure 4–10 Login Screen

Those are the login and registration screen, both follow same design because they are actually one activity, the only difference between one and the other is the visibility of different text inputs and buttons.



Register here with your email and a chosen password to gain access

Figure 4–9 Registration Screen



Bandita

Charlie

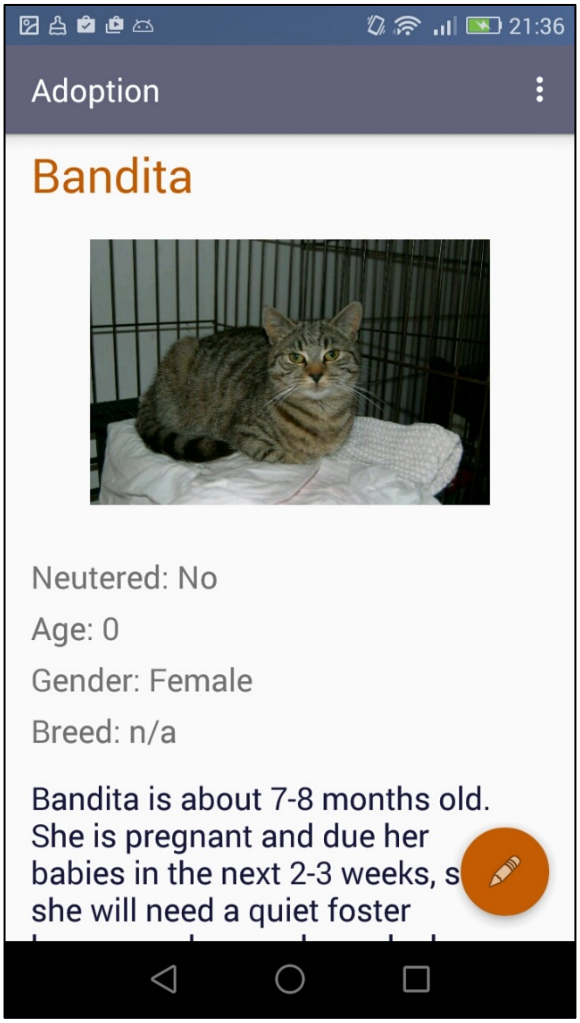
Harry

Shaun

Figure 4–11 Adoption List

This is the list of animals available for adoption which shows the animal's pictures and names. When the user clicks on the animal's name he/she is brought to the screen below (Figure 4-11).

This is the animal's profile screen, the name is clearly displayed at the top of the screen, extra details about the animal such as gender, breed, bio are shown. For the admin users, the edit button is available so the user can modify the details or delete the profile. See the edit screen (Figure 4-12).



Bandita

Neutered: No

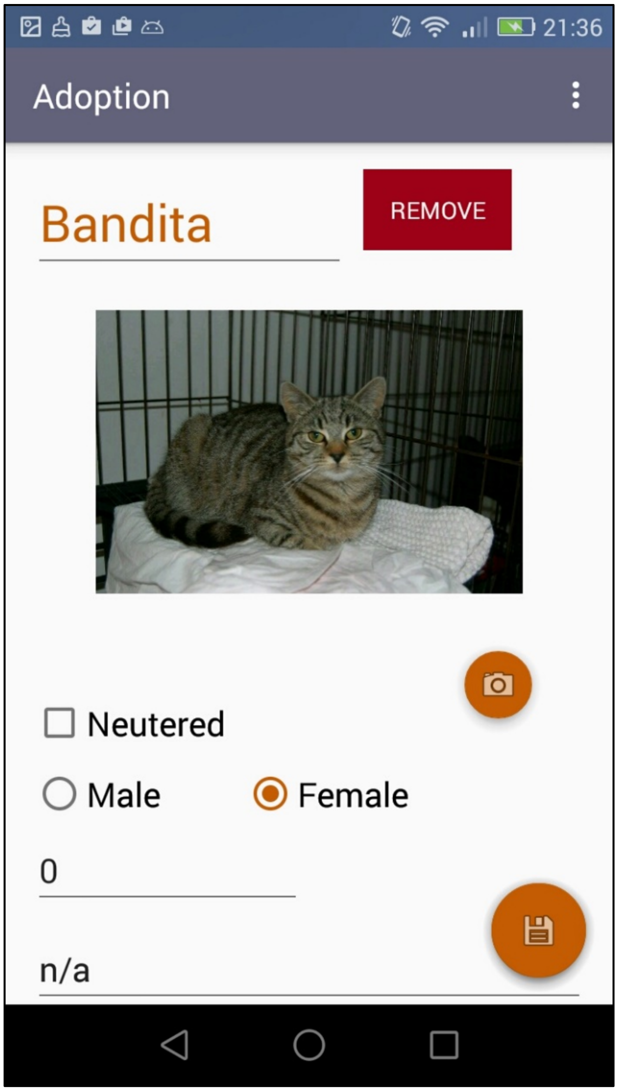
Age: 0

Gender: Female

Breed: n/a

Bandita is about 7-8 months old. She is pregnant and due her babies in the next 2-3 weeks, s she will need a quiet foster

Figure 4–12 Adoption Profile



Bandita

☐ Neutered

○ Male ● Female

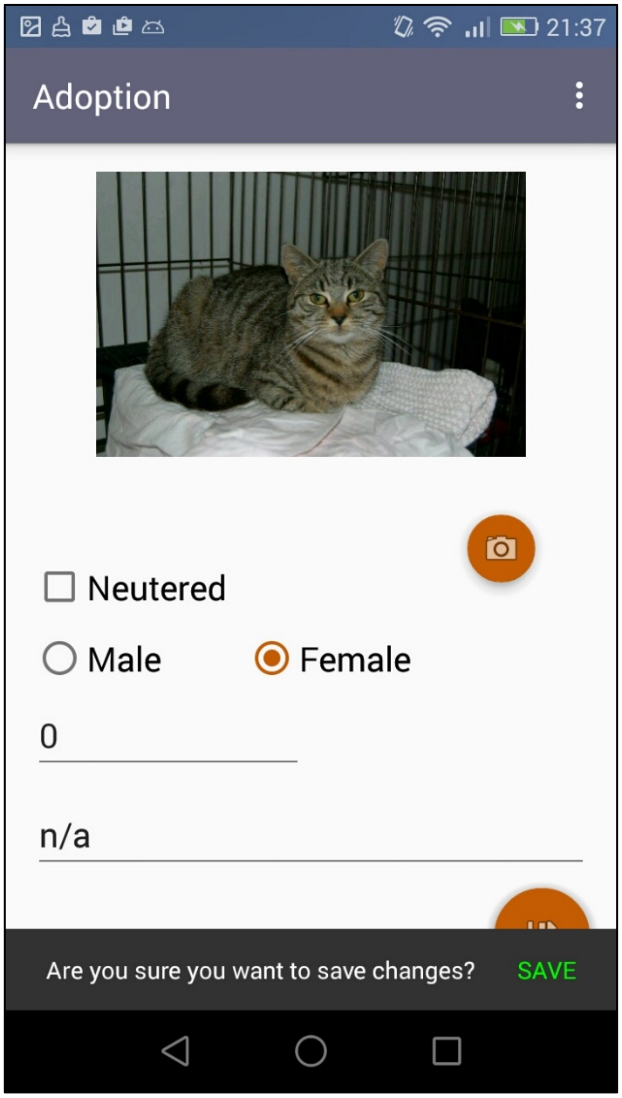
0

n/a

Figure 4–14 Edit Animal Details Screen

This screen show how the user can edit the pet details or delete the profile. The screenshot below shows that when the user clicks save on the right corner he/she is prompted to confirm the changes. See screenshot below (Figure 4-14).

In this screenshot, we can see the confirmation that is shown to the user before the changes are sent to the database.



☐ Neutered

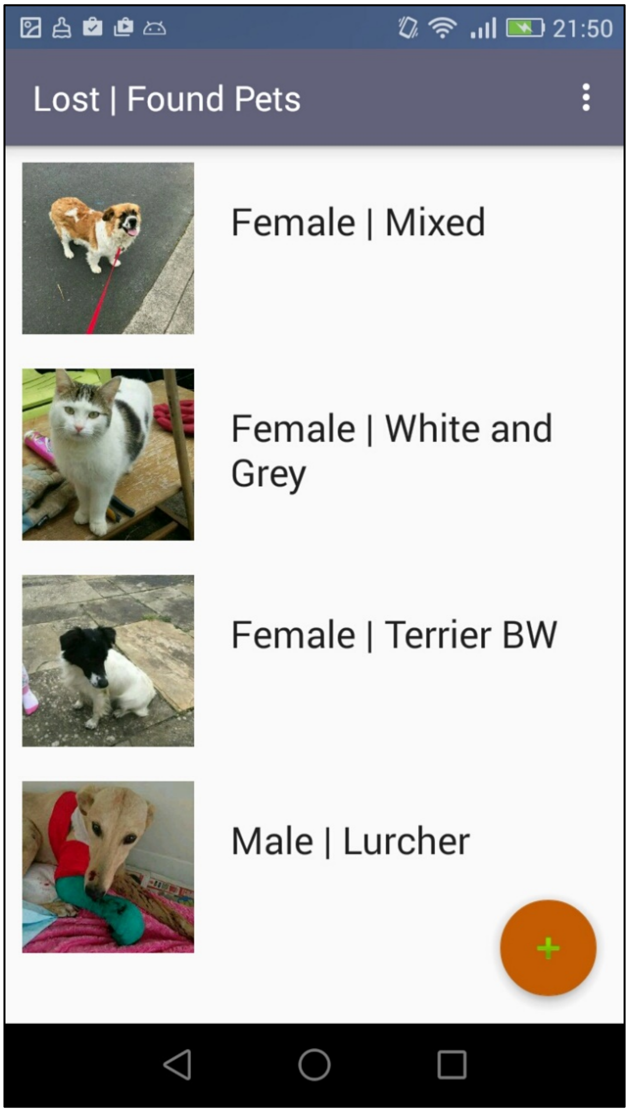
○ Male ● Female

0

n/a

Are you sure you want to save changes?

Figure 4–13 Saving Changes Screen



21:50

Lost | Found Pets

Female | Mixed

Female | White and Grey

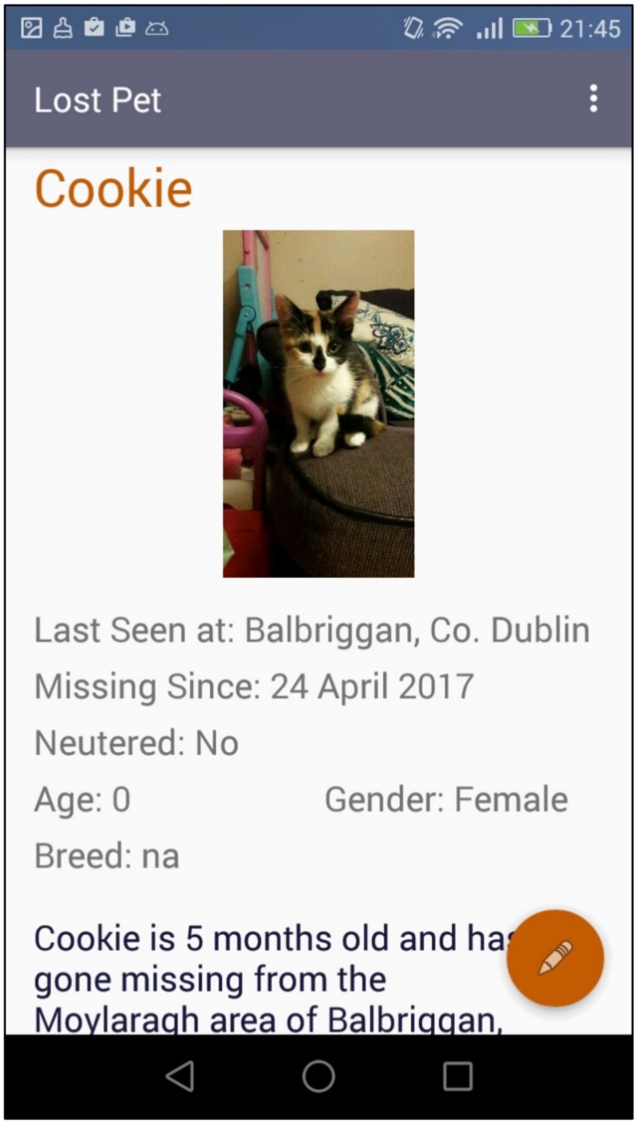
Female | Terrier BW

Male | Lurcher

Figure 4–15 Lost & Found List

This is the list of Lost & Found pets. For the found pets it shows their gender and breed, for the lost pets it shows their name and breed.

This screen shows the lost pet profile with details as location and date he/she was last seen. The pet's name is displayed at the top similar to the adoption profile. In the right corner is the edit button



21:45

Lost Pet

Cookie

Last Seen at: Balbriggan, Co. Dublin

Missing Since: 24 April 2017

Neutered: No

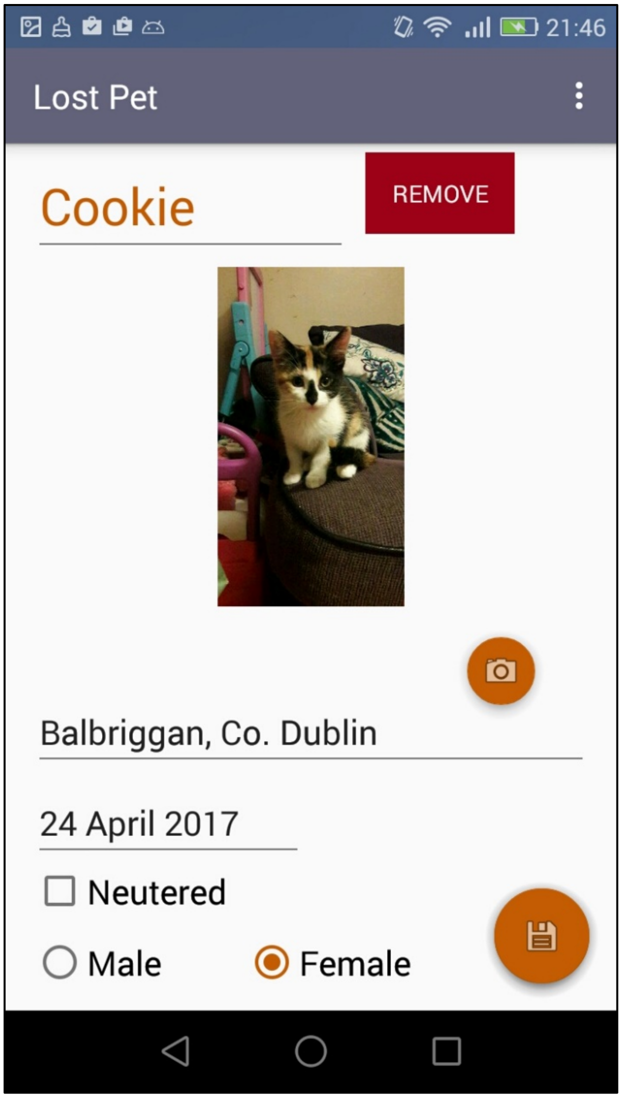
Age: 0

Gender: Female

Breed: na

Cookie is 5 months old and has gone missing from the Moylaragh area of Balbriggan,

Figure 4–16 Lost Pet Profile



Cookie

Balbriggan, Co. Dublin

24 April 2017

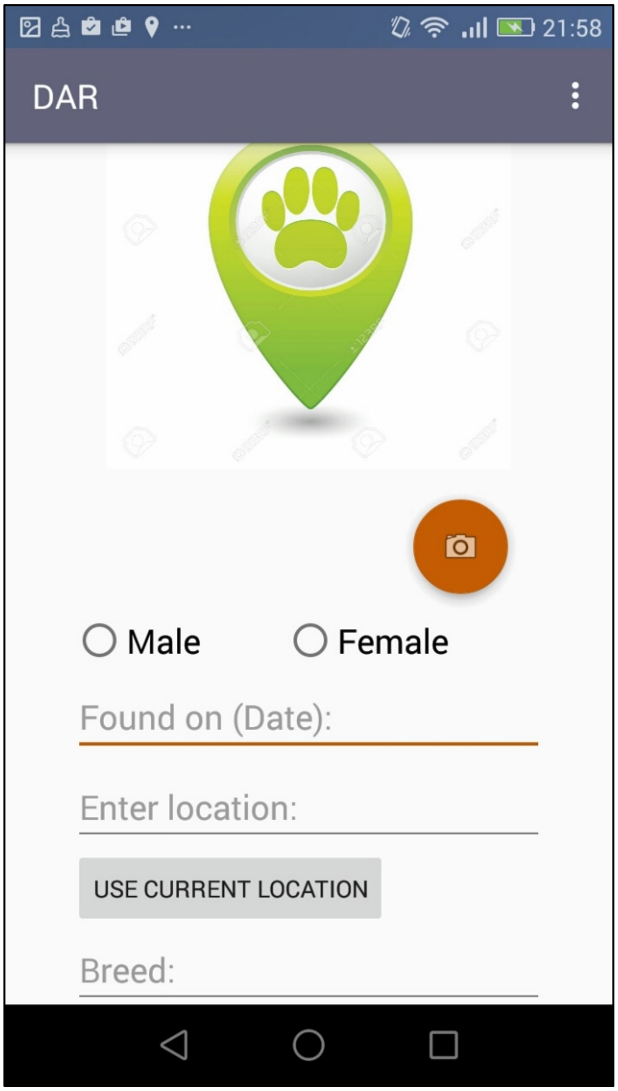
☐ Neutered

○ Male ● Female

Figure 4–18 Edit Screen Lost Pet

This screen is the one shown when the user clicks on updating a lost pet details, similar to the adoption section (See Figure 4- 13) it allows to delete the profile, change picture and overall information.

This section shows the screen the user creates a new found animal profile. The user has the option to use its current location instead of typing in one. The default image was chosen because it is related to the marker Google maps use to represent locations. The user can modify the image and take a photo of the animal.



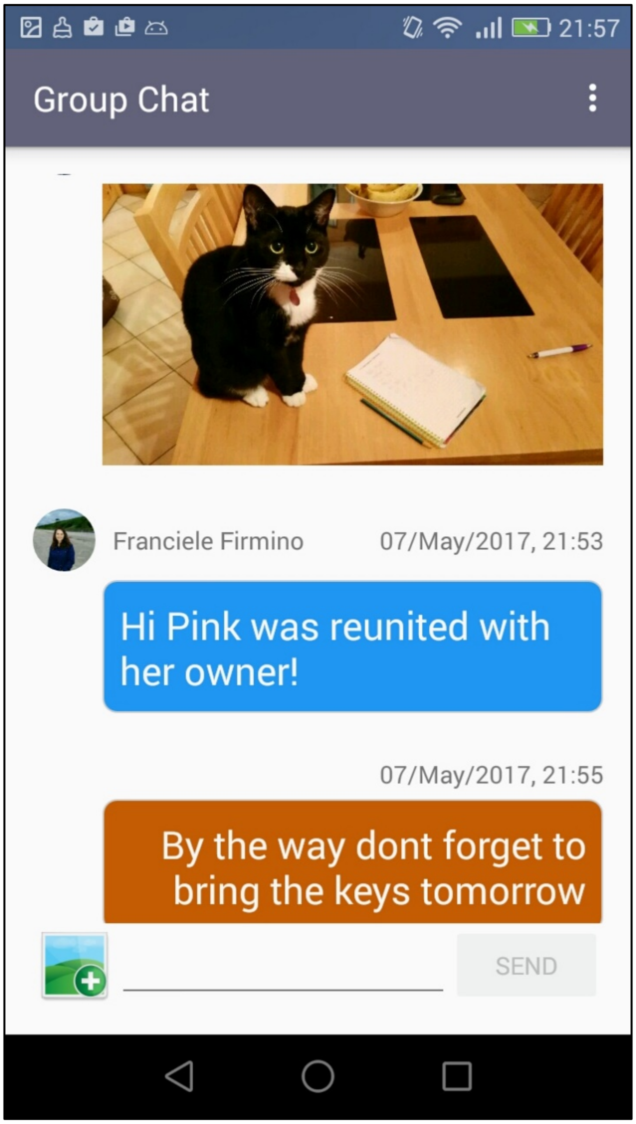
○ Male ○ Female

Found on (Date):

Enter location:

Breed:

Figure 4–17 Add Found Pet



Franciele Firmino 07/May/2017, 21:53

Hi Pink was reunited with her owner!

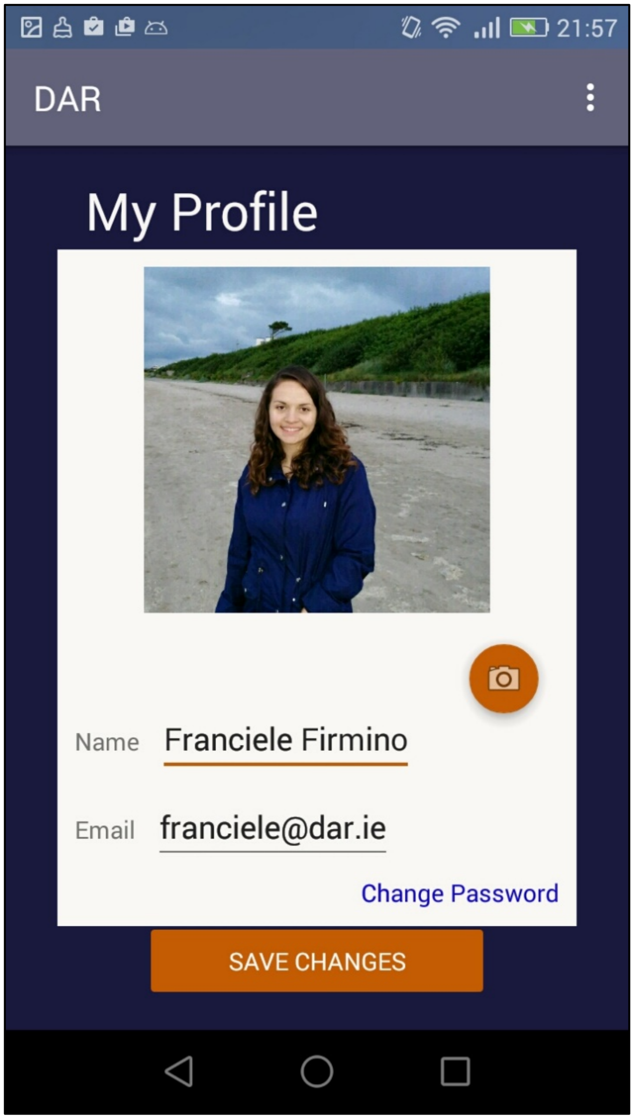
07/May/2017, 21:55

By the way dont forget to bring the keys tomorrow

Figure 4–20 Group Chat

This screen shows the group chat room, in which admin users can communicate. A received message is seen in blue and a sent message has an orange background. The chat allows images to be sent and received.

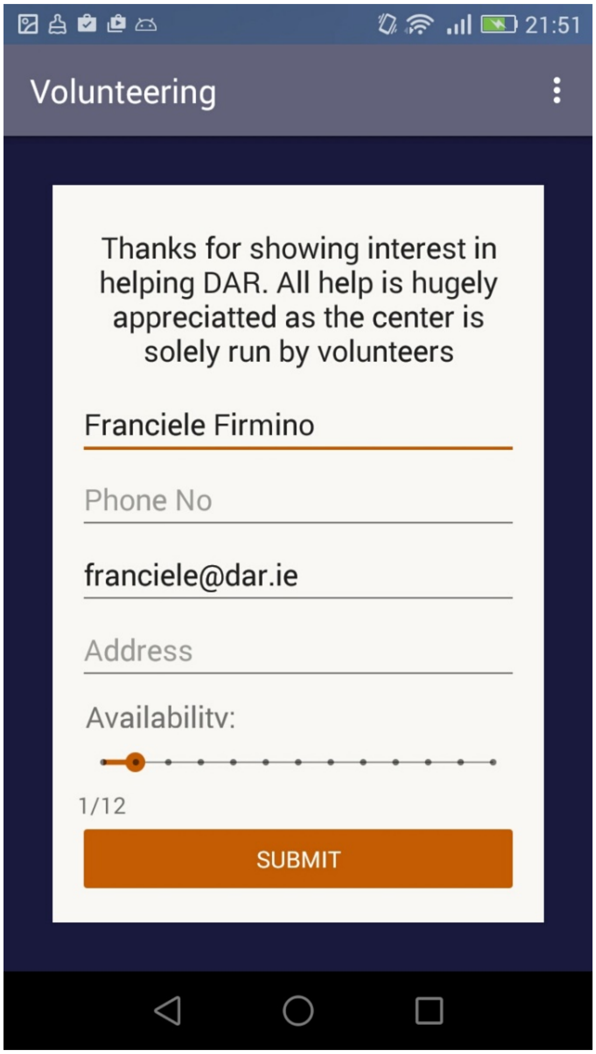
This is the profile screen in each users can update their details such as profile image, email and password.



Name: Franciele Firmino

Email: franciele@dar.ie

Figure 4–19 User Profile



Volunteering

Thanks for showing interest in helping DAR. All help is hugely appreciatted as the center is solely run by volunteers

Franciele Firmino

Phone No

franciele@dar.ie

Address

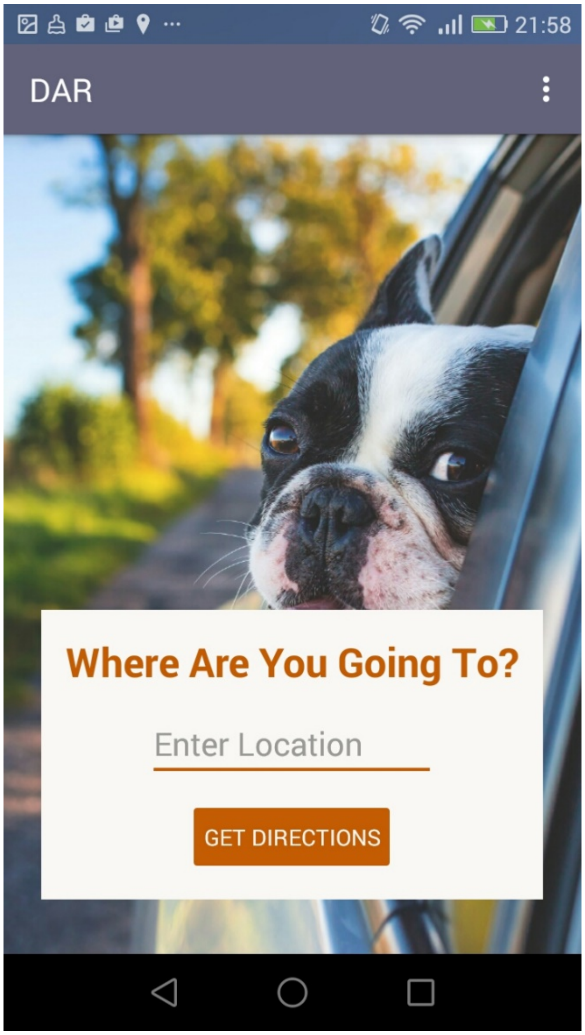
Availability:

1/12

Figure 4–22 Volunteering Form

This is the form users fill when applying for volunteer work within DAR.

That is the directions screen which allow users to get directions to places leaving from their current location. The background is relevant to the screen functionality and to the application.



DAR

Where Are You Going To?

Enter Location

Figure 4–21 Directions Screen

4.4 Implementation

The application makes use of most of the available features in Google's Mobile Platform, Firebase. Firebase is a great tool for mobile development as each allow developers to spend less time on back end set up. The best thing about this platform is that it offers analytics and crash report, by login in to the Firebase console the app administrator can view how many active users the app has, and details about its users such as demographics. This app makes use of Firebase's real-time database, push notifications, remote configuration, email authentication and storage. See below screenshots of the analytics section of the console.

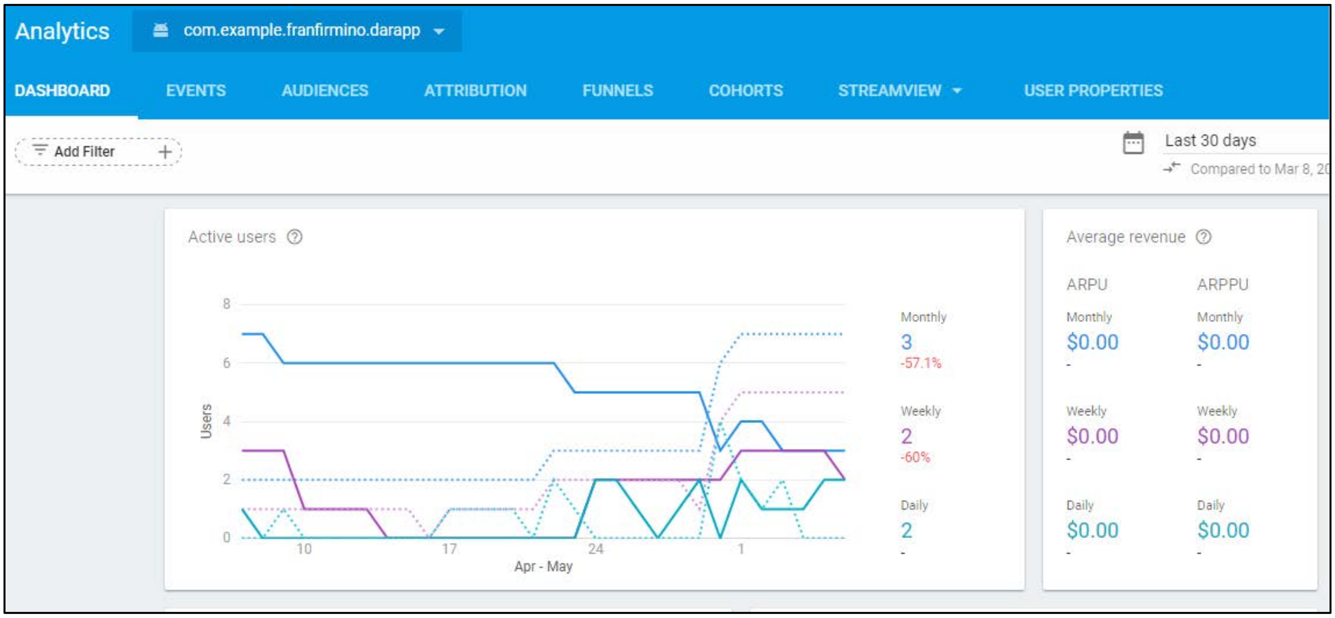


Figure 4–23 Firebase Analytics

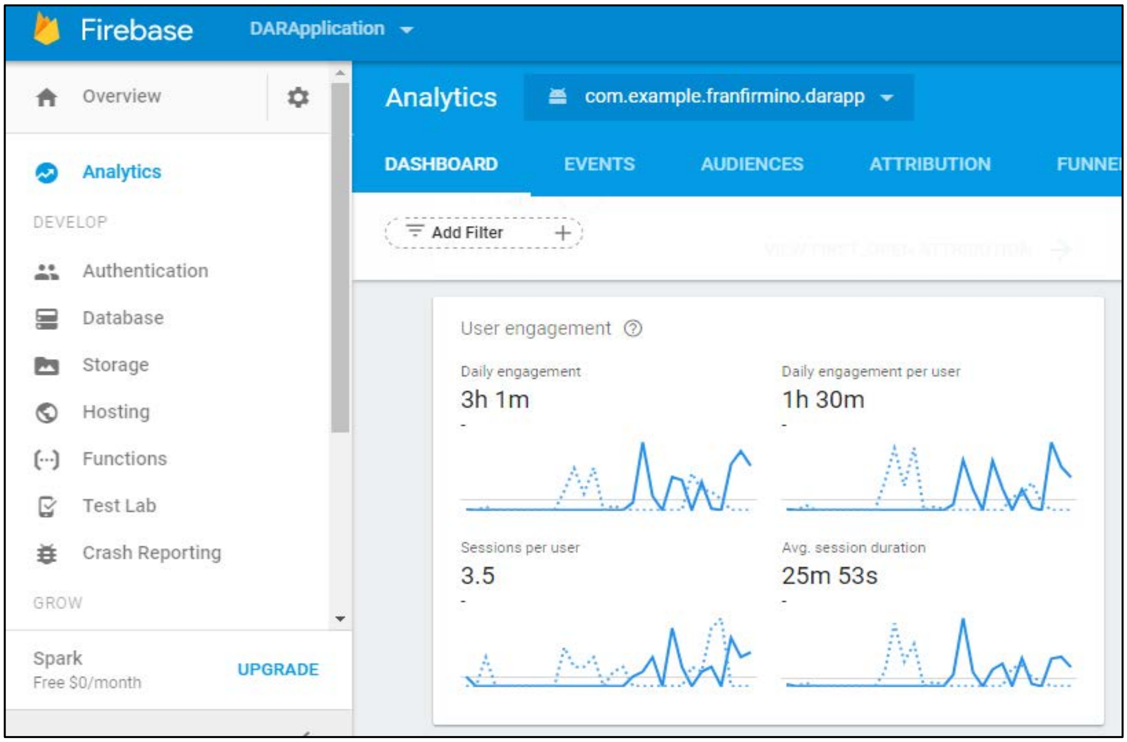
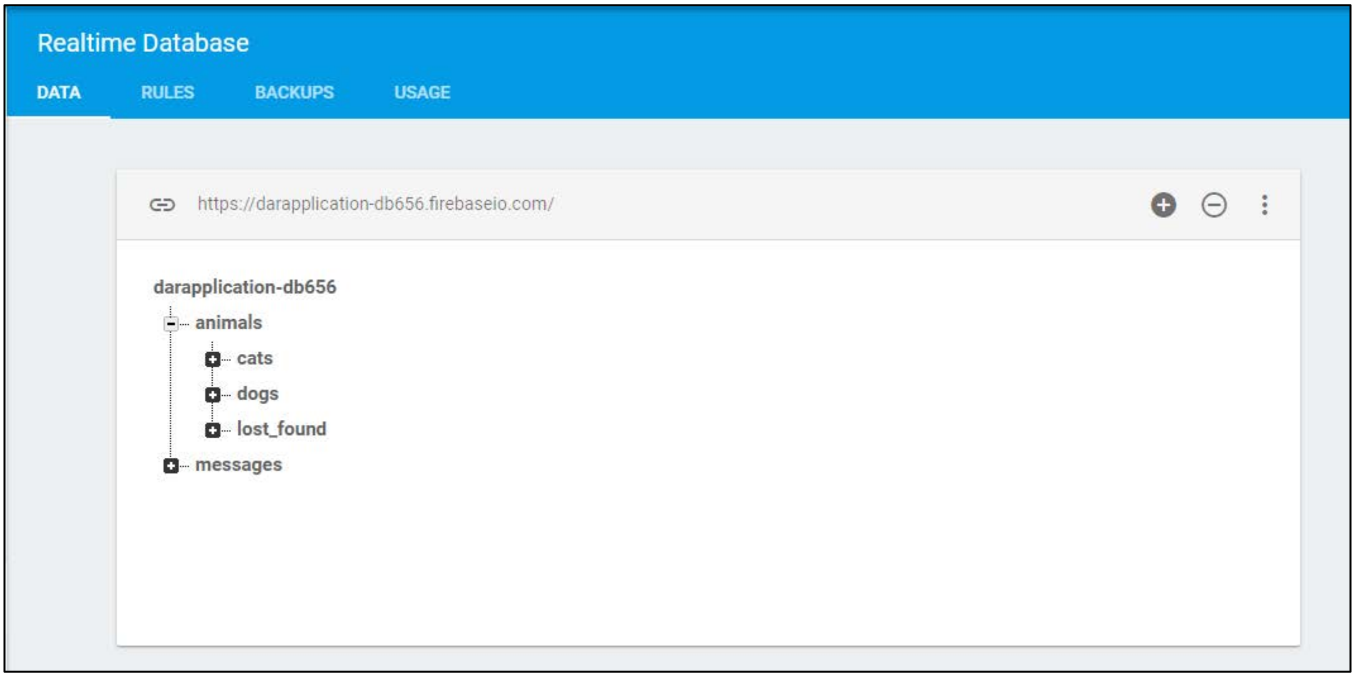


Figure 4–24 Firebase Analytics

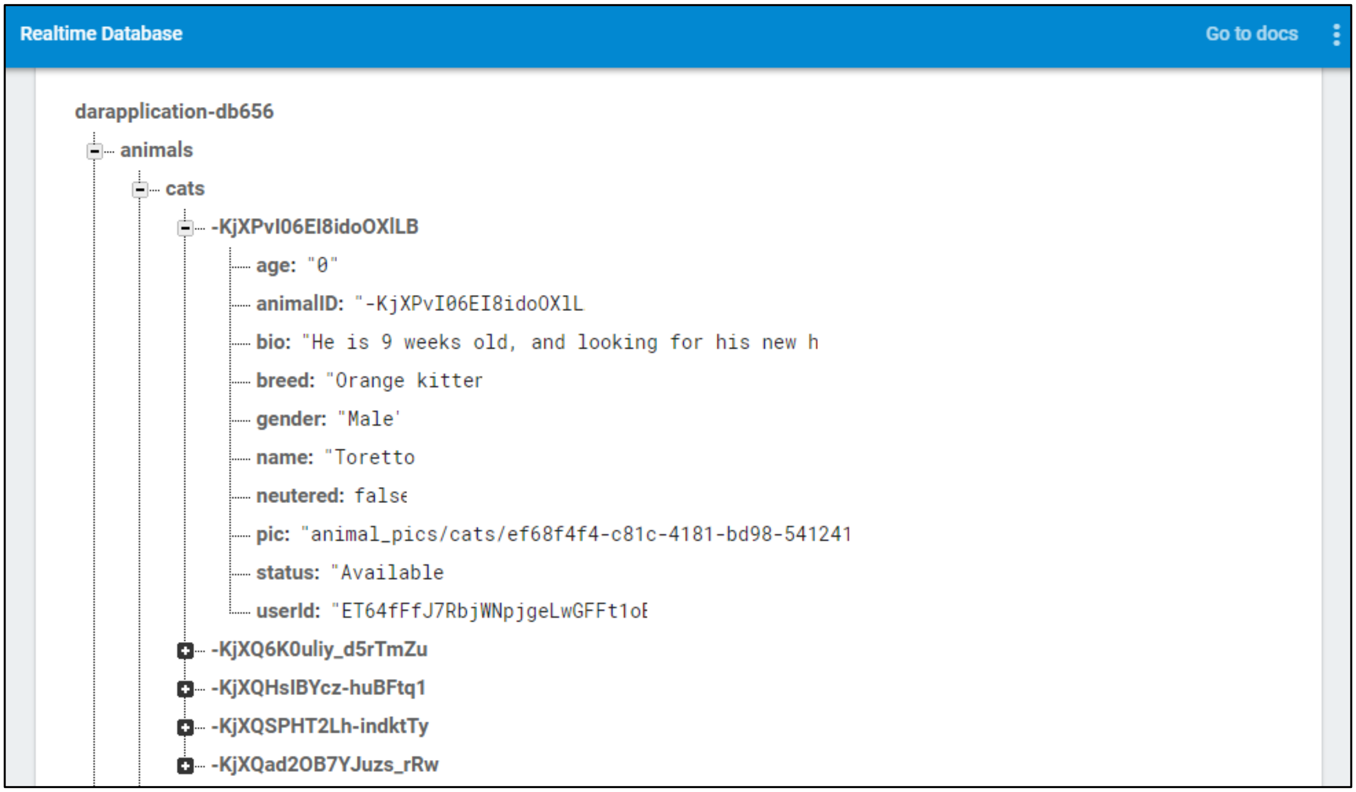
4.4.1 Realtime Database

Mostly admin users can read, write, update and delete files using the application and by log in into Firebase's Console. The database is stored in a JSON format.



```
darapplication-db656
 animals
    cats
    dogs
    lost_found
 messages
```

Figure 4–26 Database Screenshot



```
darapplication-db656
 animals
    cats
       -KjXPvl06El8idoOXlLB
          age: "0"
          animalID: "-KjXPvI06EI8idoOXlL
          bio: "He is 9 weeks old, and looking for his new h
          breed: "Orange kitter
          gender: "Male'
          name: "Toretto
          neutered: false
          pic: "animal_pics/cats/ef68f4f4-c81c-4181-bd98-541241
          status: "Available
          userId: "ET64fFfJ7RbjWNpjgeLwGFFt1oE
       -KjXQ6K0uliy_d5rTmZu
       -KjXQHslBYcz-huBFtq1
       -KjXQSPHT2Lh-indktTy
       -KjXQad2OB7YJuzs_rRw
```

Figure 4–25 Detailed Database Screenshot

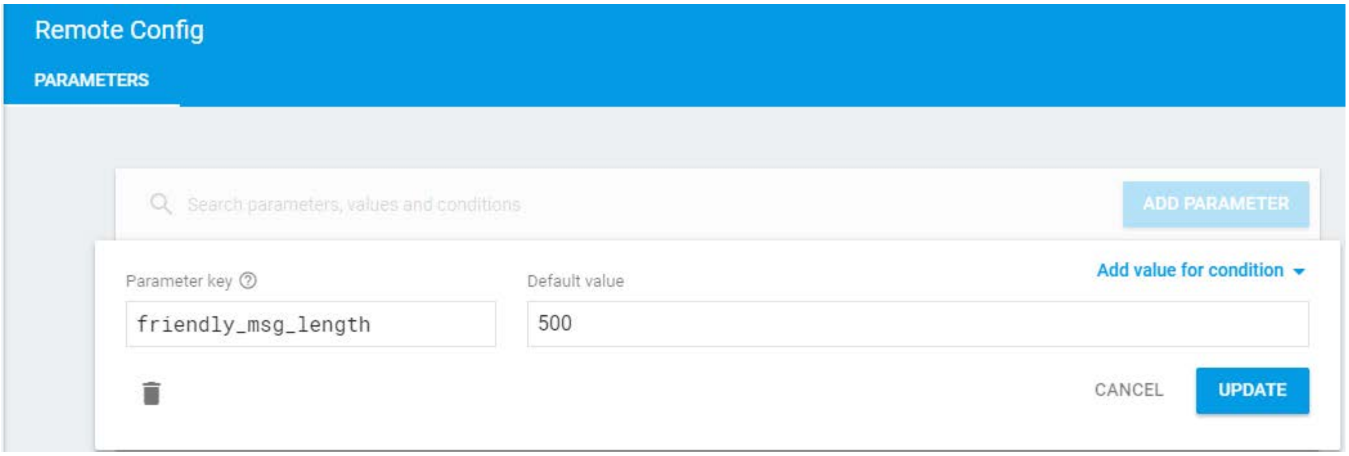
4.4.2 Storing Images

All the images uploaded to the application are saved in the external storage. To avoid images being overridden because of similar names, a unique ID is given to the image before it is uploaded, and the path to the image is saved in the database. See code snippet below of the “saveImage” method from the “BaseActivity.class”.

```
// store image in Firebase
public void saveImage(ImageView addImage, String opt, String uniqueID) {

    StorageReference picRef = null;
    // Create a storage reference from our app
    StorageReference storageRef =
FirebaseStorage.getInstance().getReferenceFromUrl("gs://darapplication-db656.appspot.com");

    if (opt.equals("cats") || opt.equals("dogs")) {
        picRef = storageRef.child("animal_pics/" + opt + "/" + uniqueID + ".jpg");

    } else if (opt.equals("user")) {
        picRef = storageRef.child("user_pics/" + uniqueID + ".jpg");

    } else if(opt.equals("lost")){
        picRef = storageRef.child("animal_pics/lost_found/" +opt+ "/"+ uniqueID +".jpg");

    }else if (opt.equals("found")){
        picRef = storageRef.child("animal_pics/lost_found/" +opt+ "/"+ uniqueID +".jpg");
    }
    // Get the data from an ImageView as bytes
    addImage.setDrawingCacheEnabled(true);
    addImage.buildDrawingCache();
    Bitmap bitmap = addImage.getDrawingCache();
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.JPEG, 100, baos);
    byte[] data2 = baos.toByteArray();

    UploadTask uploadTask = picRef.putBytes(data2);
    uploadTask.addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception exception) {
           Log.w(“Error”, exception);
        }
    }).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
        @Override
        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
            Uri downloadUri = taskSnapshot.getMetadata().getDownloadUrl();
        }
    });
}
```

4.4.3 Remote Configuration

Firebase allows to change configurations without having to modify the code. In this application, the remote configuration was used in the “ChatActivity.class” to mofidy the number of characters a user can enter in the message textbox(“mMessageEditText”). The default length has been set to “10L” in the app’s code, the name of the variable being remotely edited is “friendly_msg_length”. Below is a screenshot of the Console from where the message max length can be remotely changed.

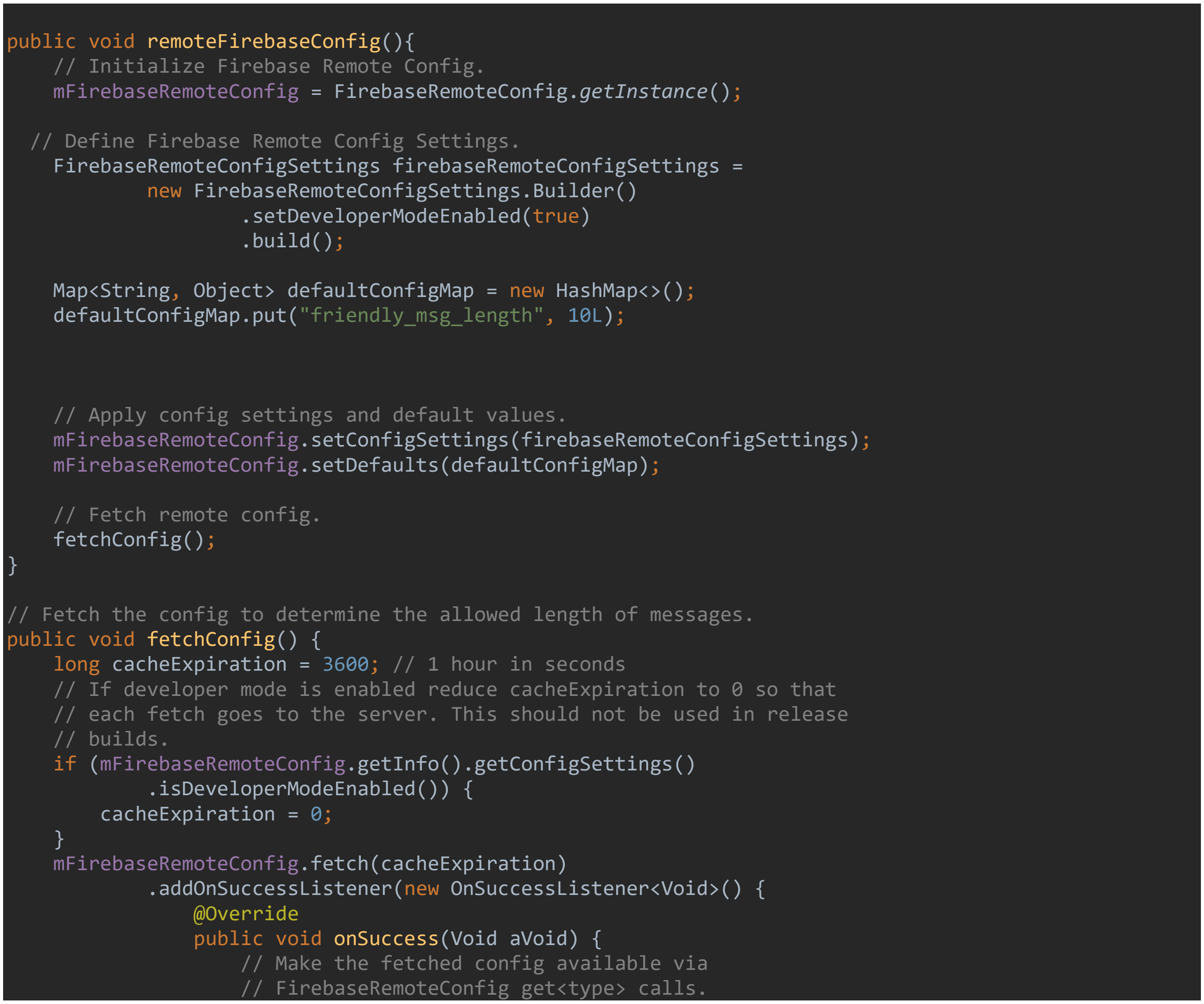


Parameter key: friendly_msg_length

Default value: 500

Figure 4–27 Remote Config Screenshot

Code below shows how the remote configuration was added to the activity.



```
public void remoteFirebaseConfig(){
    // Initialize Firebase Remote Config.
    mFirebaseRemoteConfig = FirebaseRemoteConfig.getInstance();

  // Define Firebase Remote Config Settings.
    FirebaseRemoteConfigSettings firebaseRemoteConfigSettings =
            new FirebaseRemoteConfigSettings.Builder()
                    .setDeveloperModeEnabled(true)
                    .build();

    Map<String, Object> defaultConfigMap = new HashMap<>();
    defaultConfigMap.put("friendly_msg_length", 10L);



    // Apply config settings and default values.
    mFirebaseRemoteConfig.setConfigSettings(firebaseRemoteConfigSettings);
    mFirebaseRemoteConfig.setDefaults(defaultConfigMap);

    // Fetch remote config.
    fetchConfig();
}

// Fetch the config to determine the allowed length of messages.
public void fetchConfig() {
    long cacheExpiration = 3600; // 1 hour in seconds
    // If developer mode is enabled reduce cacheExpiration to 0 so that
    // each fetch goes to the server. This should not be used in release
    // builds.
    if (mFirebaseRemoteConfig.getInfo().getConfigSettings()
            .isDeveloperModeEnabled()) {
        cacheExpiration = 0;
    }
    mFirebaseRemoteConfig.fetch(cacheExpiration)
            .addOnSuccessListener(new OnSuccessListener<Void>() {
                @Override
                public void onSuccess(Void aVoid) {
                    // Make the fetched config available via
                    // FirebaseRemoteConfig get<type> calls.
                    mFirebaseRemoteConfig.activateFetched();
                    applyRetrievedLengthLimit();
                }
            })
            .addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    // There has been an error fetching the config
                    Log.w(TAG, "Error fetching config: " +
                            e.getMessage());
                    applyRetrievedLengthLimit();
                }
            });
}


private void applyRetrievedLengthLimit() {
    Long friendly_msg_length = mFirebaseRemoteConfig.getLong("friendly_msg_length");
    mMessageEditText.setFilters(new InputFilter[]{new
InputFilter.LengthFilter(friendly_msg_length.intValue())});
}
```

4.4.4 Reuse of Code

As most classes share methods between them, and share the same drawer menu a base Activity was created and all the app activities extend this base activity, which contains all the main functionality, thus avoiding code repetition and making the development quicker.

```
//Firebase SIGN OUT
public void SignOut() {
    FirebaseAuth.getInstance().signOut();
    Intent intent = new Intent(getApplicationContext(), LoginActivity.class);
    startActivity(intent);
}

// validating emails
public static boolean validateEmail(String email) {
    String EMAIL_PATTERN = "^[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9-]+)*@"
            + "[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)*(\\.[A-Za-z]{2,})$";

    Pattern pattern = Pattern.compile(EMAIL_PATTERN);
    Matcher matcher = pattern.matcher(email);
    return matcher.matches();
}

//check if user is admin
public boolean isAdmin() {
    Boolean Admin = null;

    FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
    if (user != null) {
        String email = user.getEmail();


        if (email.contains("@dar.ie")) {
            Admin = true;
```

```
       } else {
           Admin = false;
    }
    return Admin;
}

//   Add new animal to Firebase
final public void addNewAnimal(String userId, String status, String age, String bio, String
breed, String gender, String name, Boolean neutered, String pic, String opt, String date,
String location) {

    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference myRef = null;

    if(opt.equals("cats") || opt.equals("dogs")) {
        myRef = database.getReference("animals/");
        String key = myRef.child(opt).push().getKey();
        Animal animal = new Animal(key, userId, status, age, bio, breed, gender, name,
neutered, pic);
        myRef.child(opt).push().setValue(animal);

    }else if(opt.equals("lost") || opt.equals("found")){
        myRef = database.getReference("animals/lost_found");
        String key = myRef.child(opt).push().getKey();
        AnimalLostFound animal = new AnimalLostFound(key, userId, status, age, bio, breed,
gender, name, neutered, pic, date, location);
        myRef.child(opt).push().setValue(animal);
    }

    Toast.makeText(getApplicationContext(),"Added to the List", Toast.LENGTH_SHORT).show();

    if(opt.equals("lost") || opt.equals("found")) {
        Intent addedI = new Intent(getApplicationContext(), LFBrowseActivity.class);
        addedI.putExtra("OPT", opt);
        startActivity(addedI);
    }else if(opt.equals("cats") || opt.equals("dogs")){
        Intent addedI = new Intent(getApplicationContext(), BrowseActivity.class);
        addedI.putExtra("OPT", opt);
        startActivity(addedI);
    }

}
```

4.5 TESTING

The application main functions have been manually tested and the results can be seen below.

4.5.1 Test Cases

Module Name	Add Animal for Adoption
Test Designed by	Franciele Firmino
Test Designed Date	20/04/2017
Test Executed By	Franciele Firmino
Test Execution Date	20/04/2017
Test Title	Add Animal for Adoption
Test Summary	The objective of this test is to verify that the adding a new animal to the adoption list is working appropriately
Pre-condition	The user must be an Admin The admin user is logged in
Test Steps	1. User clicks on "Adoption" button 2. Chooses type of animal (dog or cat) 3. Views the list of animals available for adoption 4. Clicks on add button 5. Fills form with animals' details 6. Clicks on "Add"

Test Data	**Set 1:**
	Name = "Lolla"
	Gender ="Female"
	Neutered = "No"
	Age = "3"
	Breed = "Terrier X"
	Bio = " Friendly boy"
	
	Set 2:
	Name = ""
	Gender ="Male"
	Neutered = "No"
	Age = "3"
	Breed = "Terrier X"
	Bio = " Friendly boy"

	Set 3:
	Name = “Lolla”
	Gender =”Female”
	Neutered = “Yes”
	Age = “”
	Breed = “Terrier X”
	Bio = “ Friendly boy”

Expected Result	Set 1: Animal is added to Firebase and confirmation message is shown Set 2: User is prompt to enter animal’s name Set 3: User is prompted to enter an age
Post-condition	The user is brought back to the list of animals for adoption and app waits for a new request
Actual Result	**Set 1:** Animal is added to Firebase and confirmation message is shown **Set 2:** User is prompt to enter animal’s name **Set 3:** User is prompted to enter an “age”
Status	Pass

Module Name	Update Animal Details on the Adoption Section
Test Designed by	Franciele Firmino
Test Designed Date	20/04/2017
Test Executed By	Franciele Firmino
Test Execution Date	20/04/2017
Test Title	Update Animal Details on the Adoption Section
Test Summary	The objective of this test is to verify that the updating a new animal is working appropriately
Pre-condition	The user must be an Admin The admin user is logged in
Test Steps	1. User clicks on “Adoption” button 2. Chooses type of animal (dog or cat) 3. Views the list of animals available for adoption 4. Clicks on an animal name 5. Clicks on the “Edit” button 6. Updates details 7. Saves details

Test Data	**Current Details:**
	Name = “Archie”
	Gender =”Male”
	Neutered = “No”
	Age = “4”
	Breed = “Siberian Husky”
	Bio = “Energetic little boy, gets along with kids”
	
	New Details 1:
	Name = “Archie”
	Gender =”Female”
	Neutered = “Yes”
	Age = “4”
	Breed = “Siberian Husky”
	Bio = “Energetic little boy, gets along with kids”
	
	New Details 2:
	Name = “”
	Gender =”Female”
	Neutered = “Yes”

	Age = “4”
	Breed = “Siberian Husky”
	Bio = “Energetic little boy, gets along with kids”
	
	New Details 3:
	Name = “Archie”
	Gender =””
	Neutered = “Yes”
	Age = “4”
	Breed = “Siberian Husky”
	Bio = “Energetic little boy, gets along with kids”

Expected Result	**New Details 1:** Gender and neutered are updated successfully **New Details 2:** User is prompted to enter “name” **New Details 3:** User is prompted to enter “gender”
Post-condition	The user is brought back to the list of animals for adoption and app waits for a new request
Actual Result	**New Details 1:** Gender and neutered are updated successfully **New Details 2:** User is prompted to enter “name” **New Details 3:** User is prompted to enter “gender”
Status	Pass

Module Name	Get Directions
Test Designed by	Franciele Firmino
Test Designed Date	20/04/2017
Test Executed By	Franciele Firmino
Test Execution Date	20/04/2017
Test Title	Get Directions
Test Summary	The objective of this test is to verify that the get directions function is correctly working
Pre-condition	The user must be an Admin The admin user is logged in Location must be enabled
Test Steps	1. User clicks on "Directions" option on the side menu 2. Enters the pace he/she is going to 3. The application opens google maps with the directions from the users location to the chosen destination

Test Data	**Set 1:**
	Destination = "National College of Ireland"
	Set 2:
	Destination = ""

Expected Result	Set 1: User is shown directions from his current location to chosen destination Set 2: User is prompted to enter a destination
Post-condition	The application is stopped. Google Maps is on the foreground.
Actual Result	Set 1: User is shown directions from his current location to chosen destination Set 2: User is prompted to enter a destination
Status	Pass

Module Name	Update Lost Pet Details
Test Designed by	Franciele Firmino
Test Designed Date	20/04/2017
Test Executed By	Franciele Firmino
Test Execution Date	20/04/2017
Test Title	Update Lost Pet Details
Test Summary	As the Lost & Found section of the app is more interactive and allows general users to enter animals to the Lost & Found list. This use case tests that if the user is logged in as a general user, if he/she will be able to update a pet that he/she previously added to the list.
Pre-condition	The user must be an Admin or the general user the added the pet initially The user is logged in
Test Steps	1. User clicks on “Lost & Found” button 2. The user choses the “Lost” option 3. The user clicks on the animal profile he previously added 4. The profile is shown 5. The user updates the profile and saves the details
Expected Result	The edit button is displayed to the user and he/she successfully update the pet’s details.
Post-condition	The application is stopped. Google Maps is on the foreground.
Actual Result	The edit button is displayed to the user and he/she successfully update the pet’s details.
Status	Pass

4.6 CUSTOMER TESTING

The testing was conducted to gather feedback about the apps usability and overall user experience, using two different testing approaches, Five Second Test and Heuristic evaluation. The results of the tests are outlined below

Participants

The total number of participants was 7…

- Of those, 5 completed a 5 seconds testing.
- Of those, 2 completed a heuristic testing.

Profile of Participants

- Five Seconds Test: picked at random, the only criteria was that the person did not have any previous knowledge about the project.
- Heuristic Evaluation: DAR committee was contacted and two active administrators of the organization undertook the test. These participants will be the ones administrating the application when it goes live.

4.6.1 Five Second Test

This test was designed based on the Five Seconds Test website (Fivesecondtest.com, 2017) which provide guidelines on how to make the most of it. The participants were shown the home screen for 5 seconds then they were asked the questions below.

Questions

Q1. Say all you can remember. Keywords.

Q2. What do you think the app is for?

Q3. What did you like the most about the design?

Q4. What did you like the least about the design?

Results

*** Please refer to Appendix's Section 8.3.1***

Discussion about Results

As the results show the main screen is visually pleasing and by looking at the main screen the volunteers could understand what the application was for and who it was for. Most of them assumed it was only focused on adoption of dogs because of the background image. As it was predicted that most users would make this assumption the splash screen which is shown before as the application loads, contains an image of a dog and cat, therefore it makes it clear the application is for both.

4.6.2 Heuristic Evaluation

The admin users were asked to download the app to their phones and complete some tasks, such as reserve an animal for viewing, exchange messages, add, update and delete animal profiles. The tasks covered all the application existing functionalities. Afterwards questions were asked about the overall user experience.

Questions

Topic: Aesthetic and minimalist design

Q1. What did you think of the app overall look?
Q2. Does it have a minimalistic approach?
Q3. What do you think of the colour scheme?
Q4. Is the app cluttered?

Topic: Help and Documentation

Q1. What is your overall impression of the help section?
Q2. Is the information well organized?
Q3. Could you easily find the help section on the app? Is it available on every screen?

Topic: Consistency and Standards

Q1. Is the application layout consistent?
Q2. Do you think the app follows a pattern? Or are the screens completed unrelated to each other.
Q3. Is it easy to know where you are in the app?

Topic: Match between system and the real world

Q1. Is the app vocabulary easy to understand and relevant?

Results

*** Please refer to Appendix's Section 8.3.2***

Discussion about Results

Aesthetic and minimalist design

The results show that the users found the application visually pleasing and liked the colours and images used. It is a good sign as the aim of the application from a design point of view, was to create an application that is not only practical but that also provides a good user experience considering that it will affect how users view DAR as an organization.

Help and Documentation

Both users could easily find the help section which is a very good result as effort was put into making into as detailed and informative as possible.

Consistency and Standards

Overall the users agreed that the application is consistent. There were comments about the login page being in a different style but that was done on purpose, as the login screen is viewed as an outside element. The login page was inspired by Twitter's android app login page which also contains a full screen image.

Match between system and the real world

The users agreed that the application match the real world, using relevant vocabulary. It was an aim of the application to use same vocabulary and terms that are already used by DAR on its Facebook page and website to make the app an extension if these other tools that are already in use.

4.7 Evaluation

As the application deals, mainly with the storage and display of data, its scalability was evaluated. To carry the evaluation the real data was collected. With that in mind, DAR provided a list of animals looking for new homes and a list of lost & found pets. The real data was then added to the application. All the profiles contained a high-resolution picture and long description bios. The load time of the adoption lists and profiles were tested.

As most of the lists and profiles follow a similar layout, for this evaluation we will focus on the adoption section.

The adoption list during development contained 3 animal's profiles with low resolution images and little profile details. Now with the real data contains 20 profiles with detailed information and as mentions above high-resolution images. The loading time for case one (3 profiles) with Wi-fi was less than a second, with mobile data 4 seconds. For case two (20 profiles) with Wi-Fi enabled was 3 seconds, with mobile data it increased to 8 seconds.

As seem by the results above, although there was a considerable increase in load time, due to the nature of the application, it is certain it won't host more than 50 profiles, therefore the loading time will not exceed 15 seconds. If in the future there is a need to reduce loading time, the code will be modified to only load the first 5 profiles and has the users scrolls down it will load then the additional ones.

5 Conclusion

The application achieved the expected goals, it covered all DAR's main requirements. It will help DAR in its day-to-day activities, and will provide a modern look to the organization. Together with the DAR's website it will allow users to have easy access to animals in DAR's care. The Lost & Found section is great because it allows users to post about a lost animal very quickly facilitating the volunteers' work and making it all more interactive.

Another great achievement was to be able to have an admin section and a general user section in the same application. Both users initially navigate through the same screens but according to their permissions they are allowed to navigate deeper into a section. For example, on the adoption screen (See Figure 4-10) admin users can view the "add" button located in the right corner which allows to add a new animal to the list, but general users are not displayed that button.

6 Further development

This application in the future could have extra functionality added which would allow messages to be exchanged between not only admin users, but also between general users and admin, thus avoiding the use of emails and keeping all the conversations in one place. The user profile could be expanded so users could save animals to their favourites section for later viewing.

The plan is to develop the same application for iOS users has the organization has a considerable amount of IPhone users.

7 REFERENCES

Firebase. (2017). *Firebase*. [online] Available at: https://firebase.google.com/ [Accessed 9 Apr. 2017].

GitHub. (2017). *mikepenz/MaterialDrawer*. [online] Available at: https://github.com/mikepenz/MaterialDrawer [Accessed 8 April 2017].

Developer.paypal.com. (2017). *REST API reference - PayPal Developer*. [online] Available at: https://developer.paypal.com/docs/api/ [Accessed 8 May 2017].

Fivesecondtest.com. (2017). *The original Five Second Test • UsabilityHub*. [online] Available at: http://fivesecondtest.com/ [Accessed 22 Apr. 2017].

8 APPENDIX

8.1 PROJECT PROPOSAL

Project Proposal

Mobile Application for Drogheda Animal Rescue

X13111787, Franciele Firmino, firminofranciele@gmail.com

BSc (Hons) in Computing

Mobile Technologies

12 October 2016

CONTENTS

BACKGROUND

For my project I decided to contact Drogheda Animal Rescue, non-profit, voluntary organisation committed to assisting both dogs & cats in need. So I offered to develop a mobile application that will benefit this organization. We agreed on the main functions for the app, and they also gave me freedom to add extra functionality.

Objectives

The objective of this project is to develop a mobile application, that will be useful for DAR (Drogheda Animal Rescue). It will help them to organize themselves internally, so keep a communication between staff, and also showcase dogs and cats available for adoption/fostering, to the general public. There will be two sides to the application, an admin side designed for active volunteers and a common side, that is for the general public. The application will be divided in two sections Admin and General Public. Details about functionality for each section are explained below.

Both sections admin and for the general public will have a profile section in which they can add/update/delete profile details as picture, name, general contact information. And there will be a settings section to manage notifications and privacy.

ADMIN SECTION

Manage Dogs & Cats
There will be an option to add a profile pic and information about the animal as animal type (dog or cat), breed, age, name, gender (Male, Male Neutered, Female, Female Spayed), color, hair length. And there will be tick boxes to select if the dog is looking for adoption or/and foster home.

Calendar of Events/Tasks
There will be a calendar which all staff can see with events or/and tasks that are coming up. The staff can assign tasks to other staff members, that will receive a notification on their phone about the task, and they can choose if they want to accept the task or decline it. If they accept the task, the task will be added to the to do list.

Manage to Do List
Each member of staff will have their own to do list. This list is filled with tasks that were assigned to them by other members of staff, or they can also add tasks to their lists themselves. They will them receive reminders about the tasks on their phone.

Select Best Path to Drop or/and Collect Animals
As some staff members need to collect animals for foster homes, drop animals to the vet and bring food to foster homes. It can all get a bit confusion, so with this feature the staff member can enter all the runs he/she has to do in a day, and the app will calculate the most efficient route to perform all those tasks.Each time a location is reached, it is taken off the map and considered complete. So the staff member can keep track of his/her performance.

If not all runs are going to be performed at the same time, a time can be set for each run and the app will remind you when a run is due. So if an animal was dropped at the vet at 1pm and needs to be collected again at 5pm, the app will send a reminder about the collection that is due.

View list of Volunteers

There will be lists of all volunteers' details that will be divided by foster homes, fundraisers, adoption day volunteers and DAR center volunteers. Those lists can then be filtered, for example, the foster homes' list can be filtered to find a foster home with no dogs, full time adult in the house, experience with big dogs etc.

Chat with Members of Staff

Staff will be able to chat with other members of staff that will receive a notification when they receive a message. They can chat with a specific member of staff or with groups, for example, the committee group or adoption day group. Chat groups can be created, update or deleted.

General Public Section

View Animals for Adoption

The user will be able to view animals available for adoption, and filter the list by their preferred age, breed, color. If there is an interest in adoption, they can then contact DAR by filling a form with their details, which will be send by email to the assigned member of staff, or they can save the animal details to the favourites section to review it later.

Become a volunteer

The user can put their details forward to volunteers' list to become a foster parent or/and a fundraiser or/and a volunteer in DAR's center.

Donate

Donations by PayPal or debit/credit card will be accepted through the app.

Check Calendar of Events

The user will be able to see which events that DAR is organizing.

Receive notifications

If looking to adopt an animal but none of them fits the criteria, the user can sign up to receive notifications based on their preferences so if an animal is added to adoption and fits the criteria the user will receive a notification on their phone.

Lost & Found

The user will be able to search for a lost pet, report a lost pet and report a pet that was found.

Technical Approach

Research

I will look for other similar applications that were developed for animals' adoption to get some ideas for layout and functionality that could be added to this application. So far I have found two mobile applications one that was developed for Venture County Animal Services and another called “Patinhas”, both have good layout and extensive functionality that will provide some guidance for the development of this application.

I will also do some research on APIs, that will help me with some of the functionalities as the chat and calendar, I might use Google Calendar for the events calendar and for the chat I still have to find a suitable one.

REQUIREMENTS CAPTURE

The main requirements will be collect form DAR’s committee and additional requirements will come from ideas found in other related mobile applications from similar organizations.

IMPLEMENTATION

I will work on the layout firstly to make sure the interface is easy to use, so I will have all the screens designed and from there I will work on connecting to the database and getting the CRUD functionalities working to get this out of the way and then I will focus on the more complex functionality.

USER INTERFACE

The user interface will be very intuitive based on DAR’s logo colors. Find below are images of the layout of the main screens.

SPECIAL RESOURCES REQUIRED

I will try to get access to the database currently used by DAR so I can connect their website’s database to the mobile application, thus avoiding having to update both platforms separately.

Technical Details

The application will be developed for Android phones using Android Studio, it will be connected to a database using PHP and the database will be hosted on an external free server. APIs will be used when possible.

Evaluation

I will run small tests after each new functionality is added. But after the application is completed, I will import the real data and run through all the features of the application, go through all possible scenarios of using the app in real life and check for reliability, precision, speed and ease-to- use.

After running all those tests on my own, I will bring it to the DAR’s committee I will download the application to their Android phones, preferably ones with different screen sizes to test the layout and also the functionality. I will let them get familiar with it and test it to see if the interface is easy to use and if all functionalities are working as expected.

Franciele Firmino 13/10/2016

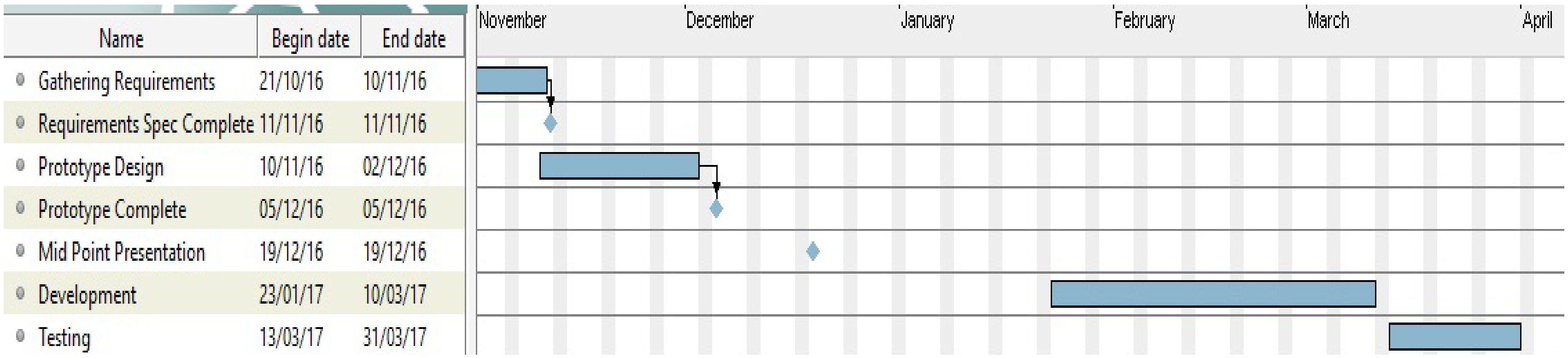
Signature of student and date

PROJECT PLAN

Below is an overview of the project plan. I will work on gathering the requirements first, then designing all activities and layout. For the prototype, I plan on having all activities designed and connected, and also have a stablished connection to the database, already collecting basic information.

In the development section I will focus on the most complex functionality first, I believe the most complex is the one that calculates the best route to perform vet runs in a day. And after having that one working I will focus on the APIs for the chat and calendar sections. Lastly, I will work on the CRUD functionality.

The last step will be testing, I firstly do the testing on my own by bringing the real data in, the exhausting the app with all different possible scenarios. Only after that I will bring it to be tested by an end user, the end user will include DAR'S member of staff.



Name	Begin date	End date
Gathering Requirements	21/10/16	10/11/16
Requirements Spec Complete	11/11/16	11/11/16
Prototype Design	10/11/16	02/12/16
Prototype Complete	05/12/16	05/12/16
Mid Point Presentation	19/12/16	19/12/16
Development	23/01/17	10/03/17
Testing	13/03/17	31/03/17

8.2 MONTHLY JOURNALS

8.2.1 SEPTEMBER

My Achievements

This month I was thinking about an application idea, I had a few but I wanted to do something that would be useful because I have done apps before that got discarded so, with that in mind, I contact Drogheda Animal Rescue to talk about a possible Mobile Application, that would benefit their staff and volunteers. Together with the DAR's committee I was able to put an interesting app idea together.

My Reflection

I felt choosing to develop an app for a real client it is a great idea, although I think it will bring a lot of pressure on me to make the app perform well. Unfortunately, I will not be able to develop for both IOs and Android so I chose IOs.

Next Steps

Next month I will try to put a proposal together to send it to DAR to let them know how I think the app will look and the functionalities that will be available. And I'll also need to upload this proposal until the 21st so I better start working on it soon.

I realised that I need to start learning how to develop an IOs application because there is something I have never done before, and with so many things to do this semester, I am afraid learning a whole new software could backfire later on.

8.2.2 OCTOBER

My Achievements

This month, I was able to start coding my project, I have been doing some research of ways to make the UI more dynamic and avoid creating too many activities. I have been basing the layout of my activities on application I found at Google Play store as I have too much functionality on this app so the way I organize it will make it easier to develop it.

My contributions to the projects included drawing a wireframe of all the project activities, I also send the project proposal to DAR so they will review it and give their opinion on it.

My Reflection

I felt, it worked well to reserve some time each week to work on the project although there are so many things to do for this semester I am afraid I will not be able to dedicated as much time to this project.

8.2.3 NOVEMBER

My Achievements

This month, I done a bit of research and discovered Firebase database which will help add a database to my application. I also met with my supervisor to talk about my progress.

My Reflection

I have been finding it really hard to keep up with all the assignments we have this year, also because I work part time so it seems there is never enough time for anything. I also noticed that I take too long to do things as to add a slider menu it took me nearly a whole day as I am learning as I go. So things moved a bit slow this month, I focused more on writing the Requirements Doc and tried to do some of the tutorials to learn how to use Firebase.

Next Steps

I have to complete my Technical Report and try to implement this Firebase database to my app before my mid-point presentation.

Supervisor Meeting

I met Dominc Carr on the 14th November to discuss my progress. He gave some ideas on where to find tutorials for Android Studio.

8.2.4 DECEMBER

My Achievements

This month, I put a PowerPoint presentation together for my mid-point presentation which I think went really well and I also started writing my technical report which I think will be helpful when writing my final report. I also got to understand how to implement the Firebase Realtime Platform to my app.

My Reflection

I did not progress to much in relation to implementing new functionality but by writing the technical report I got a better understanding on how I should approach the remaining parts of the project.

After the presentation I took the lecturers feedback and I ll try to implement some of their ideas to my app as for example adding a messaging functionality instead of using the mail function because the Firebase mobile platform I am using as a very easy to implement messaging system so I will have a look at it and see if I can implement it to my project to give an extra bit of functionality, I am not certain I ll add it but I will consider if I am okay time wise.

Intended Changes

After the exams in January, I will try to go back and complete the whole GUI with the slider menu, and dynamic pages, and hopefully then I ll have the animal’s data stored in the Firebase database.

Supervisor Meeting

Date of Meeting: 12 December

Items discussed: Firebase Database setup, I was having issues setting up the Firebase platform. We worked together to try and find a solution for it.

Action Items: Get ready for the mid-point presentation and continue working on the GUI.

8.2.5 JANUARY

My Achievements

This month, I was able to modify my material design and select a color pallete for the application. I focused more in the user experience, even modified some of the UI to make it easier to be navigated. I worked on adding a menu drawer to all the activities but after hours working on it I had no success so I decided to only add a menu drawer to the homepage which I found is the most common thing in Android applications. I also planned my next steps as I did not have much time to do anything over the exam period.

My Reflection

I felt, I did not do much this month due to the exams so I ll try to do more during February and start working on the core functionalities.

Intended Changes

Next month, I will try to add the picture to the Firebase Database, and load all the data from the database. I will also try to complete the PayPal functionality which I have halfway done.

I realised that I need to reserve around 10 hours week for this project, because every new piece of code I add to it, it has taken too much time to get it right, so I need to reserve more hours to it.

Supervisor Meetings

Date of Meeting: 10 February 2017 – 3pm to 3.30pm

Items discussed:

- Grades for Mid-point presentation
- Next Steps
- Final Marking Schemes

Action Items:

- Review my Gant Chart, to see if I am following what I planned
- Focus on adding the core functionality

8.2.6 FEBRUARY

My Achievements

This month, I was able to find an easier way to add the slider menu that was proven very hard to implement on the existing activities. To achieve that I used a library, so with a few lines of code I was able to add a fully function slider menu to all the activities needed.

My Reflection

I felt, it worked well to bring my laptop everywhere with me so every break I had between classes I used that time to try to fix the problems I encountered.

However, I was not successful in completing the tasks I thought would have been done by now as the database functionality and login as small things as the slider menu took too much of my time but due to the layout of the app the slider menu was definitely a must have on the app, thankfully as I explained above I found a library online which allowed me to fix the problem with easy.

Intended Changes

Next month, I will try to focus on core functionality and leave the layout, colors and etc for later stages as I notice I spend too much time on the GUI. So from this month on I'll focus on functionality and at the end I'll work on the look and feel of the app.

Supervisor Meetings

Date of Meeting: 10 March 2017

Items discussed: Currently progress and what I should work on during the reading week

Action Items: Implement the PayPal functionality and work on the Directions API.

8.2.7 MARCH

My Achievements

App is completed, I am happy with the overall result. Most of the requirements were met. And after testing all seems to be working as expected and the organization I am developing the app for is very pleased with the finish product.

Supervisor Meeting

Date of Meeting: 8 May 2017

Items Discussed: Look over the report to see if any changes are required.

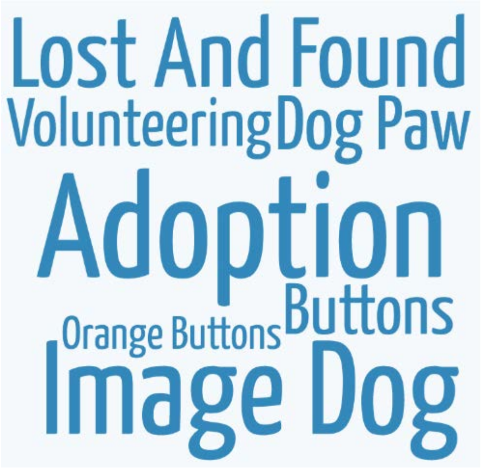
8.3 Customer Test Results

8.3.1 Five Seconds Test

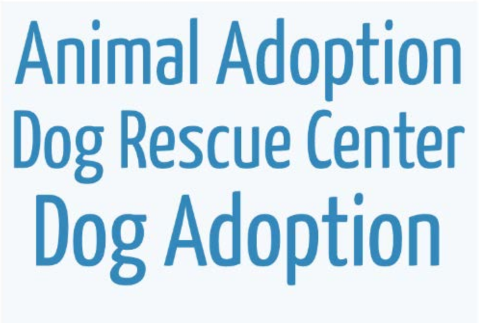
The results are shown in a Word Cloud format.

Questions

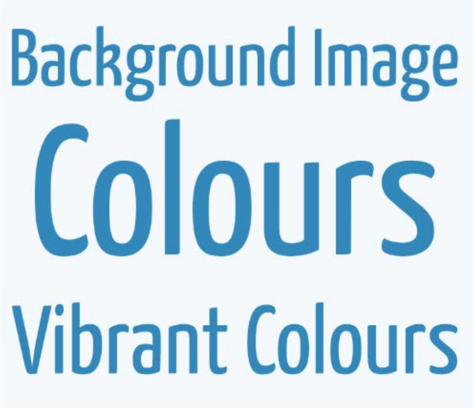
Q1. Say all you can remember. Keywords.



Q2. What do you think the app is for?



Q3. What did you like the most about the design?



Q4. What did you like the least about the design?

No Clear Logo
Nothing
Buttons Covering Image

8.3.2 Heuristic Evaluation Results

Person 1 looked at the general side of the application while
Person 2 looked at the admin section.

Aesthetic and minimalist design

Q1. What did you think of the app overall look?
Person 1: I like the home screen, it has vibrant colours and the background adds to it. I also like the way the animal profile looks with the large animal image.
Person 2: It is clean, colours are nice, background pictures are relevant to the topic.

Q2. Does it have a minimalistic approach?
Person 1: I believe it does, as only the necessary information is being displayed. For example, the homepage shows what we would be looking for when we first login.
Person 2: Yes, it does. The main screen only contains what it is needed. On the animal's screen their details are well display too.

Q3. What do you think of the colour scheme?
Person 1: I like it all blends very well.
Person 2: The colour combination is good.

Help and Documentation

Q1. What is your overall impression of the help section?
Person 1: The help section is good, very informative.
Person 2: It is good. I know if I was to look for help I could easily find it there as it explains about each screen in detail.

Q2. Is the information well organized?
Person 1: Yes, but I think maybe instead of giving all the information together. It could show only the help for the screen I am looking at.
Person 2: Yes, as I could see it is divided per apps sections.

Q3. Could you easily find the help section on the app? Is it available on every screen?
Person 1: Yes, it is on the side menu.
Person 2: It is not displayed on the screen but it can be found on the side menu which is available on every screen.

Consistency and Standards

Q1. Is the application layout consistent?
Person 1: Yes, it has same style throughout the application
Person 2: Well the login page has a different design if compared to the others but after login, all the screens follow the same design and colours.

Q2. Is it easy to know where you are in the app?
Person 1: Yes it normally says at the top the screen you are viewing
Person 2: Yes, I can see the title on every screen.

Match between system and the real world

Q1. Is the app vocabulary easy to understand and relevant?
Person 1: Yes, I could understand everything
Person 2: Yes, it is easy to understand and It is relevant to the topic