University of Glasgow | School of Computing Science

Teleport: A Location-Based Image Request App

Magda Kowalska

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Level 4 Project — March 28, 2016

Abstract

Nowadays it is easy to share photos with people around the world from wherever you are and whenever you want. However there is no currently available photo sharing applications which would allow you to see how somewhere looks like in the moment when you request it. This project aims to solve this problem by creating a location-based image request application. The project consists of designing, implementing and evaluating the application.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: _____ Signature: _____

Contents

Chapter 1

Introduction

1.1 Motivation

An immense amount of currently available technologies enables people to communicate between one another in many different ways. Online communication takes on many forms such as messaging, sharing photos and videos or writing publicly visible posts. Applications like Facebook, Skype, Whatsapp, Snapchat, Instagram, Yik Yak or Periscope are changing the way in which we communicate every day. Through these applications, users broadcast their thoughts, photos, videos and other users can view, endorse and share them. It is never the case that a user can request certain content which would be impromptu created. A considerable amount of users access these applications on their mobile devices however these applications rarely take advantage of what mobile devices are very good at, i.e. knowing their current location. We argue that the possibility of requesting certain content based on its location can lead to an exciting and meaningful way of communication.

Existing photo sharing applications provide static content without any interference between the producer and the consumer. They sometimes provide location feature, i.e. the user can specify where the photo was taken, however this functionality often fails at making the photography more meaningful because the location it refers to is too general.

The content shared in most of the currently available communication applications is eternal; every piece of data that users create and share online stays there forever. This makes users vulnerable to companies that are interested in monetising this stored data in order to approach them with very well targeted advertisements. On the other hand, ephemeral ways of communication seem to attract more and more users who find them more secure and fun. The success of Snapchat is a great example of this trend[36]. However, ephemeral communication is still something very new and currently not that prominent among application producers.

The idea of Teleport was born in order to investigate how a location-based ephemeral photo request application would affect the photo sharing experience. Ultimately we are interested in how it would affect day to day communication. Thorough investigation of currently available applications used for any kind of communication purposes have made us realise about the lack of similar technologies. Therefore we decided to bring to life a novel application and investigate how people would use it.

1.2 Aims and Objectives

The overall aim of the project was to design, develop and distribute an image request application in order to investigate how the novel way of location-based communication alters the way users communicate. The project

can be divided into two parts - the first being the implementation of the application and the second being the evaluation and the analysis of its use. There were hence two sets of main objectives:

- Design and Implementation
  - Identification of functional and nonfunctional requirements
  - Interface design
  - Architecture design
  - Implementation of a basic prototype
  - Iterative implementation of the final product
- Evaluation
  - Qualitative data collection in a form of field trials to see how the application is used by end users
  - Evaluation of field trials in a form of interviews in order to find out about users' experience
  - Quantitative analysis of user behaviour based on logging data

1.3 Outline

This report outlines all the steps that were taken during the development of Teleport, from an idea definition to the evaluation of the final product. Below is a synopsis of the report divided into the respective subcategories:

Chapter 2 - Background This chapter describes existing mobile applications used for communication and photo sharing purposes and the existing research on aspects important for the topic.

Chapter 3 - Requirements This chapter discusses the project's requirements and the variety of ways in which they were elicited.

Chapter 4 - Design This chapter describes the visual designs of the application and the iterations process it went through.

Chapter 5 - Implementation This chapter describes in detail the implementation process and decisions that were made throughout the process.

Chapter 6 - Evaluation This chapter covers a detailed overview of the evaluation actions and outcomes, and the conclusive analysis.

Chapter 7 - Future This chapter outlines what the future holds for Teleport, including feedback from the user evaluation and my own expectations on how to expand and improve the application.

Chapter 8 - Outcome This chapter presents the results of the project.

Chapter 9 - Conclusion This chapter outlines summary and reflection on the project as a whole.

Chapter 2

Background

2.1 Existing Artefacts

2.1.1 Instagram

Instagram [17] is a photo sharing application - one of the most popular in its category. Users can post photos to their feeds and browse photos taken by people they follow. Instagram is strongly based on the concept of hashtags, concise single-worded descriptions that outline contents of photos. Users can also browse photos by their hashtags alone. Photos are publicly posted and there is no direct interaction between users except liking each other's photos.

2.1.2 Snapchat

Snapchat [22] first introduced the ephemeral way of communication among mobile applications. In the application, users can take a photo(or record a video), draw on it, apply filters and then share it with the specified list of recipients. Shared content can only be seen once and for just a few seconds. There is no history of sent and received snaps. Besides sharing single photos, users can also create Stories - a series of snaps that can be viewed many times but disappear after 24 hours.

2.1.3 Yik Yak

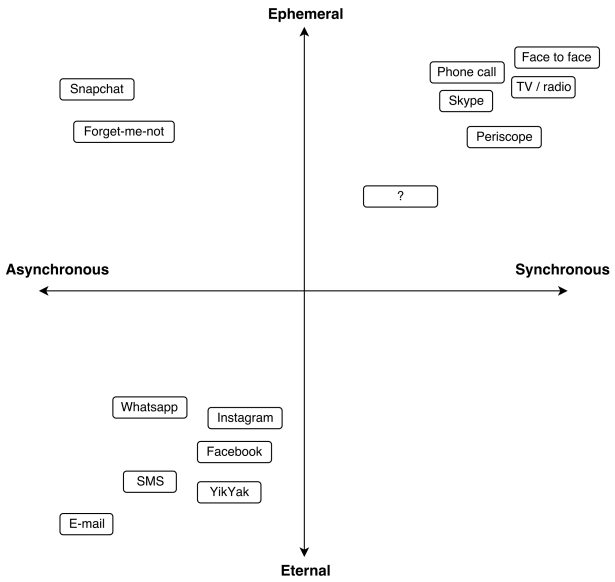
Yik Yak [27] is an anonymous social media application where users post messages that are publicly shared with users within a 5 mile radius. The messages can be up-voted and down-voted to gain or lose popularity. There is no friends or followers functionality. A Yik Yak account is verified by a user's phone number. Yik Yak is often described as a bulletin board for local areas.

2.1.4 Periscope

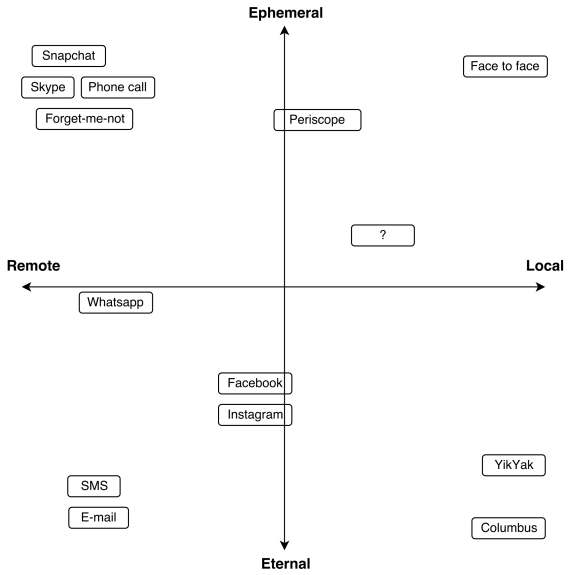
Periscope [21] is a location-based live streaming application. A user can live stream videos from his phone and other users can watch it in real time. There is no history of the streams publicly visible. Users also get notified if there is a live stream currently happening in their area.

2.2 Analysis of Communication Applications

Figures 2.1a and 2.1b show the results of analysis of popular communication applications in terms of ephemerality, synchronicity and locality aspects. The Y axis represents the ephemeral and the eternal nature of communication. The ephemeral is everything that exists in the moment, is not recorded or saved therefore cannot be accessed after the moment in which it exists. For instance, face to face conversation or a Snapchat message. The eternal represents ways of communication that keep in their history all shared content, such as E-mail or Facebook. In Figure 2.1a, the X axis represents asynchronous and synchronous ways of communication. In a synchronous example, both sides that communicate with each other have to be present at the same moment in order to communicate. For example Skype call or Periscope stream. On the other hand, asynchronous communication allows the producer to share some content and leave it available for consumers to access it at any time. For instance, Instagram photo or Yik Yak post. We believe the reason why the Synchronous and Eternal quarter is empty is because if communication is synchronous then it must be ephemeral. In figure 2.1b the X axis represents the remote and the local nature of communication. In a remote example, the location of producers and consumers is not an important factor. For example Skype call, Whatsapp message or E-mail; users are not interested in each other's location when communicating using these tools. Facebook and Instagram have location-tagging functionality therefore are placed to the right from the aforementioned applications. Periscope is also in-between. For producers, it is a very local way of communication because they share videos of their current surroundings, however it is remote for the consumers as they can be anywhere in the world while watching the videos. This process helped us identify the lack of ephemeral location-based applications.



(a) Asynchrnous vs. Synchronous



(b) Remote vs. Local

Figure 2.1: Communication applications analysis

2.3 Analysis of Existing Research

Although Teleport introduces a novel idea of requesting photos from locations, there exists research on similar topics which has helped with requirements capturing and design process. To start with, in 2007 Miller and Edwards carried out an empirical study that focused on photo sharing culture [37]. In the study they investigated how people use popular the photo sharing website Flickr [14]. They classified two user groups - Kodak Culture and Snaprs. Kodak Culture was first described by Chalfen and refers to "whatever it is that one has to learn,

know, or do in order to participate appropriately in the home mode of pictorial communication" [30]. In other words Kodak Culture is the old generation of traditional camera users which only believes in physical copies of photos and shares them with friends and family during various gatherings. On the other hand, Snaprs are strongly focused on belonging to online communities, their main activity being "photo-strolling", prefering to meet online rather than face-to-face [30]. They are what modern users of photo sharing applications could be defined as. Due to the dramatic increase of popularity of mobile devices and the creation of applications such as Facebook, Instagram or Snapchat, most of our society belongs to Snaprs nowadays. For both Kodak Culture and Snaprs, the photo sharing model requires two end points - a producer and a consumer. The producers take photos and share them with the consumers. In Teleport we are extending this model by adding a requester. It is no longer about being just a producer. The producer follows a requester and produces a photo on his demand. The requester is therefore a co-producer of the photo and a consumer at the same time. People share photos online because of the commitment to the online community and because they feel socially rewarded while doing so [38]. In Teleport we want to explore if this phenomenon is preserved if the photo is requested by another user, not just freely posted by one. We are interested in whether or not users feel like they belong to temporal communities interested in certain locations.

Locality is another important aspect of Teleport and has been widely covered in various researches. Due to the ubiquity of GPS-enabled mobile devices, we are experiencing an expansion of location-sharing applications that allow users to continuously sense and share their current location. Tang et al. [44] distinguish purpose-driven and social-driven location sharing. Purpose-driven mostly applies to one-to-one communication, where users are in need of providing crucial information of their current location to their close friends or family. In this example users usually allow the location to be much precise and perceived as much more secure. Social-driven location sharing favours semantic location names (less precision) and is used to attract attention, and boost self-representation. Barkhuus et al. [28] argue that location-sharing is not just about practical need or social boost, but has to do with emotional and moral interest. Furthermore it is used to express moods, lifestyle and events. In Teleport we want location sharing to work in an indirect way. Users would not share their location with others but would be asked to take a photo of their current surroundings if someone's request happens to be within a proximity of their location. Essentially, we want to investigate if this current research can apply to this model. Geotagging photographs is another way of sharing your location. However this concept is not that novel and has been widely used by applications like Flickr Map [15] or Panoramio [20]. Essentially Google Street View [16] is a an enormous collection of geotagged 360 degree photographs where users can see many places around the world. These ones however lack the presence of human factor. They are collections of photographs of spaces. Speed [43] argues that it is the human activity that transforms a space into a meaningful and identifiable place. A study on MobiPhos [32], a collocated photo sharing application, revealed that photographs become more meaningful if we can associate them not only with the place itself but with the the person who took it and its perspective. We hope this will apply to Teleport hence we are interested in what users valued the most about the photos of spaces they received. A study on Columbus [42], a mobile application that allows people to see photos at the location they were taken in, reveals great value of photo sharing experience when combined with a physical presence in the original location of the photo. In Teleport users can request photos from the comfort of their homes however the receivers are in that moment physically present in the location of their photographs. We hope this will benefit Teleport users in terms of the quality of the photo sharing experience.

Another aspect we want to change in regards to existing geotagged photography applications is the addition of the ephemerality condition. Snapchat, which incorporates the ephemeral idea, although has only been on the market for less than 5 years, is currently the third most popular social network [39]. Charteris et al. [31] in their study about disappearing social media explain what ephemeral messaging is for, as "Ephemeral messaging with disappearing data enables users to capture and share temporary moments rather than posting more permanent images". Sharing the small moments is what Bayer at al. [29] identify as the main incentive for using Snapchat. They argue that that most other social media applications focus on sharing big moments, the ones that have big value to the producer. In Snapchat on the other hand, users can share the most spontaneous moments of their lives. Roesner et al. [41] carried out a thorough investigation on Snapchat usage trends. They found out that that the case of time restriction makes Snapchat more fun to use and the experience is better since users do not have

to think about the history of old messages. Following this research, in Teleport we decided to choose 10 minutes as the timeout period in which a request is active, i.e. when users can reply to requests and view photos. There is no history of sent or received photos and requests after this time.

Finally a question arose about how the area to which users send requests to can be identified as. Automated name generation for an area of that size seemed computationally infeasible. Moreover, we want Teleport users to be able to communicate with each other, not just share static photos, and thus thought of a functionality where users could name the area themselves. The study about GeoNotes, an application that allows users to virtually place notes in physical places, showed that users' creativity in naming the places was beyond author's expectations [33]. This encouraged us to allow users to name the location themselves. There are no constraints in naming the place so that users can use their creativity to the fullest.

The last aspect of the research conducted for this project had to do with the content users might share while using Teleport. Despite common opinion about Snapchat's use for inappropriate content [35, 40], the study reported [41] that Snapchat users mainly share non-sensitive content (e.g. funny content). This possibility however made us aware of the need to apply appropriate content-rating when publishing the application [34].

2.4 Conclusion

Background research reinforced the motivation for this project by pointing to a gap in communication technologies. We wanted to keep exploring what has never been done before. The next step of this process was to identify the functional and nonfunctional requirements of the application.

Chapter 3

Requirements

3.1 Requirements Gathering

The key concept of the application was to have a photo requesting functionality strongly focused on locality. To achieve the best user experience for this idea a lot of design decisions had to be made. What is the threshold of locality? How can we define ephemerality? How many photos can a user receive from one request? Finally, how can location functionality be implemented without tracking the user?

3.1.1 Brainstorming

These, as well as other questions arose during weekly meetings with the project supervisor after we established what the general concept of the application was. Initial sketches along with user scenarios were created in order to explore different solutions and ideas. The biggest challenge at this point was defining locality and how to let the user choose a specific location he is interested in. Taking the research into consideration, we chose a 200m radius area to which the user can send a request so that the privacy concerns are respected and that the locality aspect is preserved.

At this time, different name options were considered. The project initially was called Instaprompt, which was a working name for the first few weeks of the project. The second idea was Snamp, which was a variation of words “snap” and “tramp”. However, due to pejorative interpretation of the word “tramp” we decided to keep exploring different name possibilities. As the result, we decided to name the application Teleport. This added a unique value to the application; requesting to see photos from places you are not currently in is like teleporting.

3.1.2 Sketches and Informal Discussions

Another way of deciding on specific features and clarifying how they would work in the final product was visualising how they would appear in the application. In the early stage of the project new sketches were created after each brainstorming session. Although these sessions were very fruitful they did not answer all the outstanding questions. As the idea evolved, informal interviews and discussions were conducted in order to validate some of the ideas with potential end users. Those helped with deciding whether users should be anonymous or not, how big the time limit should be and deciding on implementing a reward system in the form of likes.

3.2 Functional Requirements

Functional requirements capture the intended behaviour of the system. All of the defined functional requirements are equally important in order to achieve completeness of the application. See Table 3.1.

FR	Requirement	Description
1	Registration and Login	User must be able to register an account by providing their email, password, username, usual location and profile photo. They must also be able to login to the application using the email and password specified in the registration phase.
2	Sending requests to locations	User must be able to navigate in the map to their desired location and send a request.
3	Including short message in a request	While sending a request user must be able to type in a short message they wish to include in the request.
4	Receiving requests under location match condition	User must be able to see other users' request if and only if location of the user and location of the request are at most 200m away.
5	Reply to requests by sending photos	User must be able to reply to a request by taking a photo. After sending first photo user can access the request's gallery and add more photos.
6	Requests are only active for 10 minutes	Lifetime of a request starts the moment a user creates it and lasts for 10 minutes. No history or photos of the request exist after that time.
7	Likes and Dislikes	User must be able to like or dislike other users' photos. This provides feedback on whether a photo was good or not. Likes add up to user's overall score so that the feedback .
8	Notifications	User must get notified (even if the application is off) about important events such us: new upcoming request, new photo added to a request user created or uploaded a photo to.

Table 3.1: Teleport Functional Requirements

3.3 Nonfunctional Requirements

Nonfunctional requirements are used to denote non-behavioural qualities of the system. See Table 3.2.

NFR	Requirement	Description
1	Security	The application uses users' location in order to determine if a request is in their proximity. In order not to track users system should be able to achieve this task without storing users' current location in the database.
2	Usability	The application introduces novel idea but it should remain similar look and feel to popular applications to make users more confident while using it. The interface should be appealing and easy to use.
3	Availability	The application must be published in Google Play Store in order to create bigger user base.

Table 3.2: Teleport Nonfunctional Requirements

Chapter 4

Design

To decide on a design of the application we took an iterative design approach. Creating the design of the application took place simultaneously with conducting background research, brainstorming and creating user scenarios. This process altogether took two months and every weekly meeting can be counted as a different iteration. In this chapter we focus on the most important steps of this process.

4.1 Sketches

The initial idea and the design of the application were highly simplistic. In this design, the main view contained a text box to type in the name of the place and a button saying "Go now!". The user was only able to see a received photo once and it was accessible by clicking on the notification in the notification bar. The idea was based on one-to-one interaction where a user requests a photo and only one other user in that location could reply with a photo. Sketches of this iteration can be seen in Figure 4.1. A problem we identified with the initial design was that it is computationally infeasible to determine a user's physical location using his phone's GPS coordinates in regards to requested location being just a name of the place. For example if a user sends a request by typing "University of Glasgow", there is no data that determines which coordinates are inclusive to that specified area.

Therefore, we decided to add a map as the main view of the application in which users could navigate to the area they are interested in and send a request there. The application got extended by two more views; the view to the left from the map contained the users' current requests that they made themselves; the view to the right from the map being other people's requests that the users can reply to. Sketches can be seen in Figure 4.2. The view "Requests I made" contains a list of all active requests a user has made with the indication of how many photos other users have replied to and how much time the user who made them has left to access them. The view "Others' requests" shows active requests in the proximity to the user. User can choose to either reply to them by taking a photo or declining them and removing the from the list. The design at this point was still quite basic but gave a great overview of what we expected from the application.

The main requirement of the design was to keep it simple and clean. As specified in the nonfunctional requirements the interface should resemble interfaces of popular applications. The main map view is the the center of the application with two views emerging to the left and to the right from it. This idea was taken from Snapchat, where the main view consists of a camera, the left view containing photos you received from friends and right view containing of stories your friends have shared.

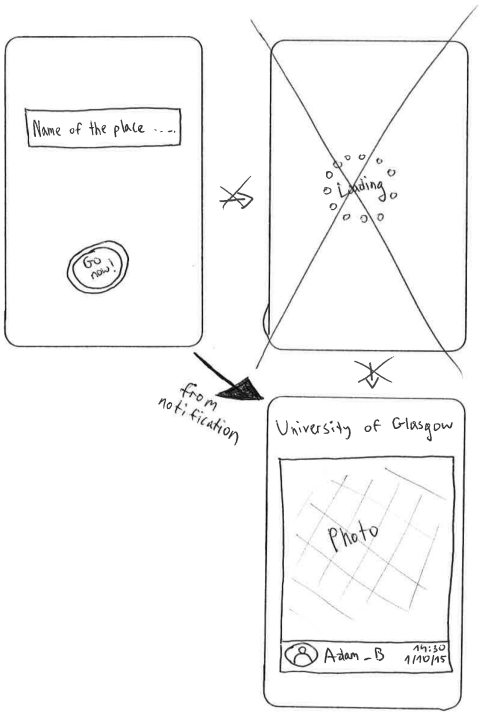


Figure 4.1: Sketch of the initial idea

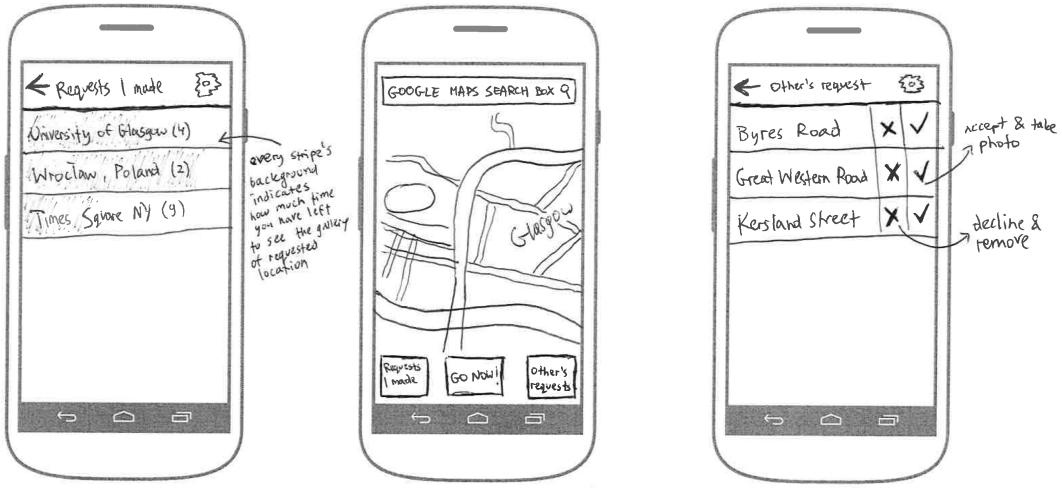


Figure 4.2: Sketch of the first design with a map

4.2 Balsamiq Wireframes

In the next stage of creating the design, sketches were translated into Balsamiq wireframes. Balsamiq [5] is a graphical user interface mockup builder, which allows the designer to arrange pre-defined widgets using a drag-and-drop editor. It provides more a formalised view of the application and provides a real look and feel that corresponds more to the actual implementation. For the first time, gallery and settings views were considered in the design. Balsamiq wireframes can be seen in Figure 4.3.

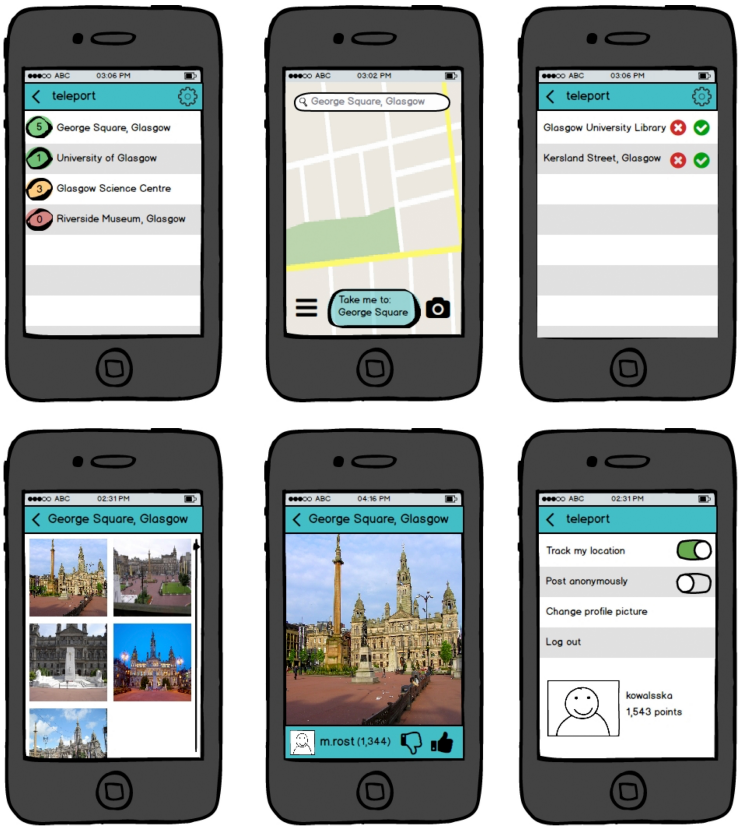


Figure 4.3: Design created using Balsamiq

4.3 Icon

Along with creating the design of the application, the icon also went through a couple of iterations. We wanted the icon to resemble common imagination about teleportation. For example Stargate's transportation rings [24] or Stark Trek's transporter [23]. At first the logo only consisted of the rings and fairly dull colours. In the second design a new colour scheme, taken from the professional colour scheme website Design Seeds [11], was applied and the name was added at the bottom. At this stage the logo looked much more professional but we felt like it was not exactly what we wanted. With help from a friend, the icon got its final shape. Circles got replaced by swirled arrows that still resemble the popular image of teleportation but look more interesting. To fill out the empty space in the middle of the icon, the letters TP (coming from TelePort) were inserted inside the arrows. The final version of the arrows design replaced the classic blue colour with quite a unique turquoise colour. It adds a unique branding value to the application. The process of creating the icon can be seen in Figure 4.4.

Figure 4.4: Icon design iterations

4.4 Final Design in Sketch

After a number of iterations the final design was created and can be seen in Figure 4.5. It answered all usability questions and incorporated all required functionalities. The circle was added to the Map View so that users could point it to the desired location and send a request precisely there. The popup box was added so that users were able to add a short message to their requests. A feedback information is displayed after a request is sent. The views displaying lists of requests got updated with additional information such as the number of photos in the gallery and timers. The gallery itself changed so that all photos are shown in their full size and users can scroll through them.

At this stage a basic implementation of the application was in place so these changes were applied to it. Section 4.5 explains in detail all functionalities and shows how the sketches were transformed into implementation.

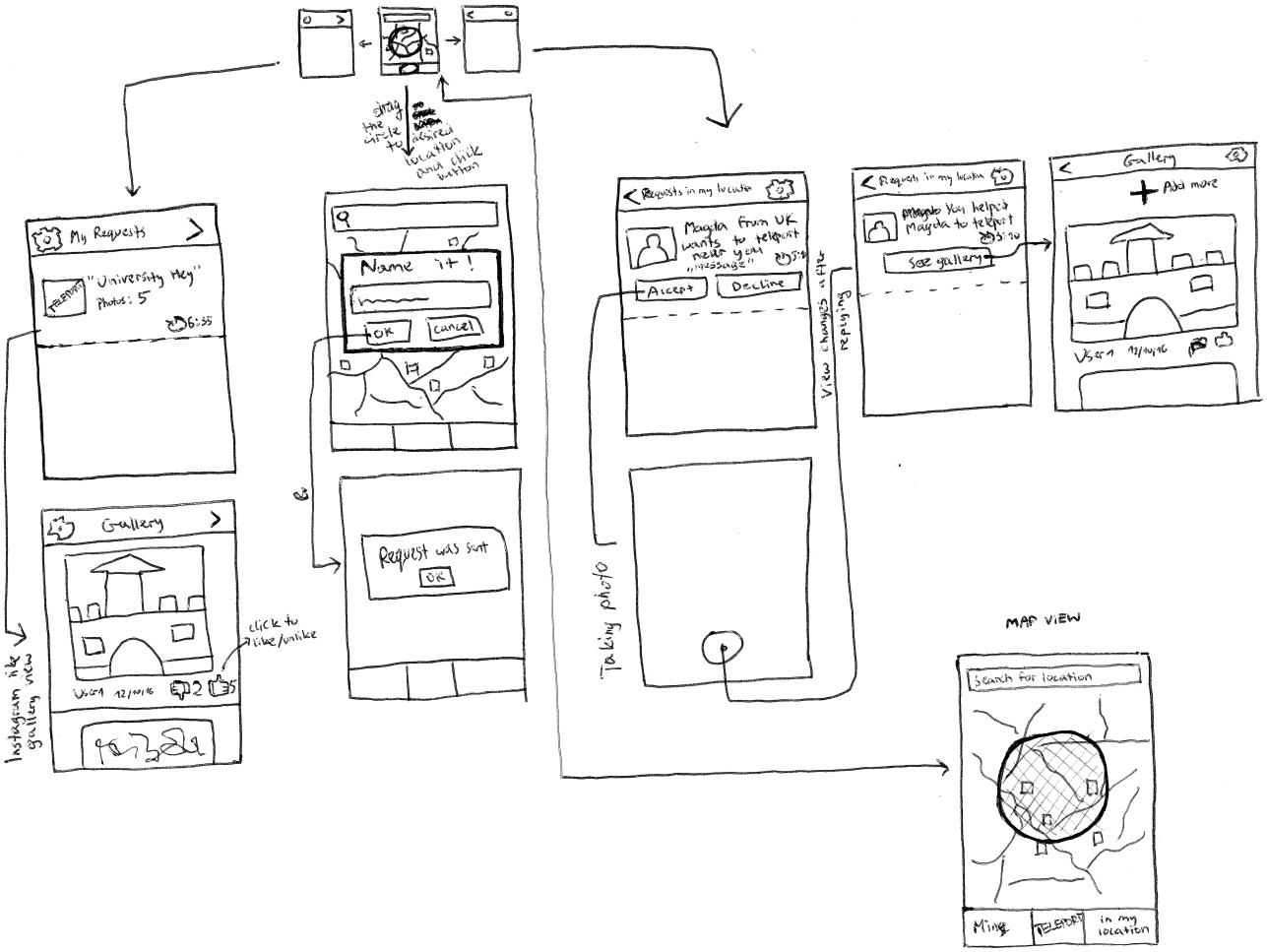
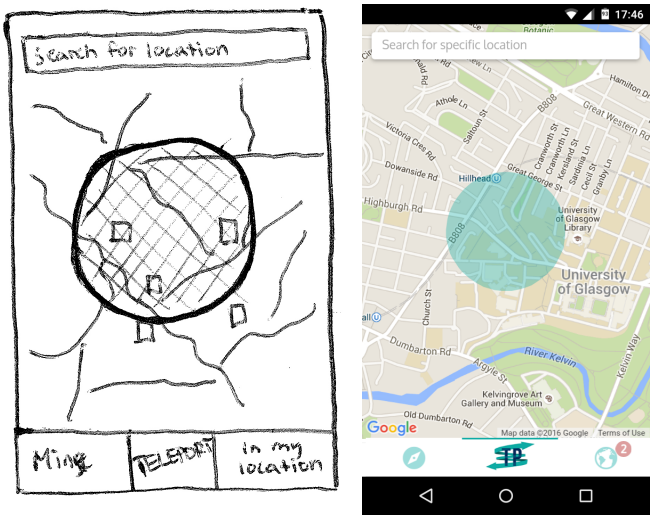


Figure 4.5: Sketch of the final design

4.5 Final Design Implemented

The main view of Teleport, the map, can be seen in Figure 4.6. The circle in the map always moves together with the map and always stays in the center of the screen. The circle itself is not draggable. The search box at the top allows users to search for specific locations. The tab bar at the bottom contains three buttons. The middle button

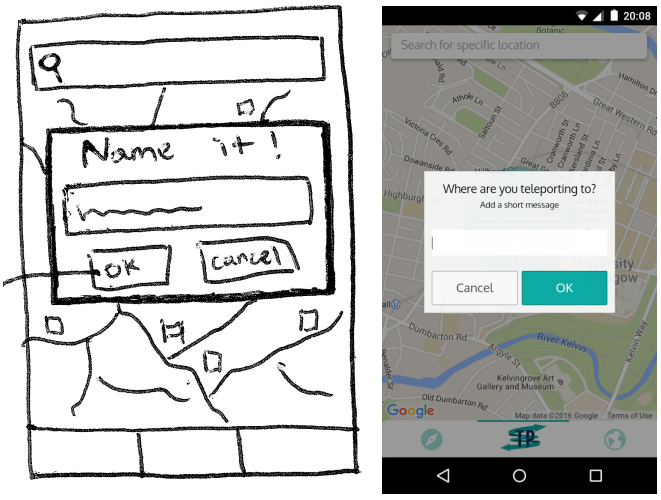
with the Teleport logo on it allows one to send requests. The button on the right takes users to “Request in my location” view and the button on the left takes users to “Your requests” view. The button on the right has also a small indicator which shows how many requests are currently active in the “Request in my location” view.



(a) Sketch (b) Implementation

Figure 4.6: Map View

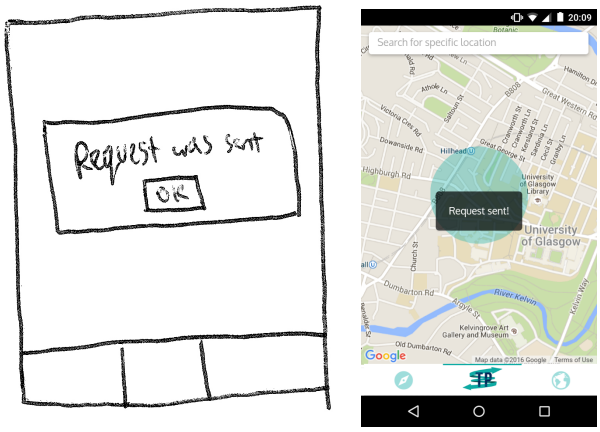
Once a user clicks on the Teleport button, the popup window appears where users can type in a message to describe requests. See Figure 4.7. The OK and Cancel buttons switched their positions because of the default buttons positioning suggested by the development framework Ionic. Once users click on the text box a keyboard appears.



(a) Sketch (b) Implementation

Figure 4.7: Add short message View

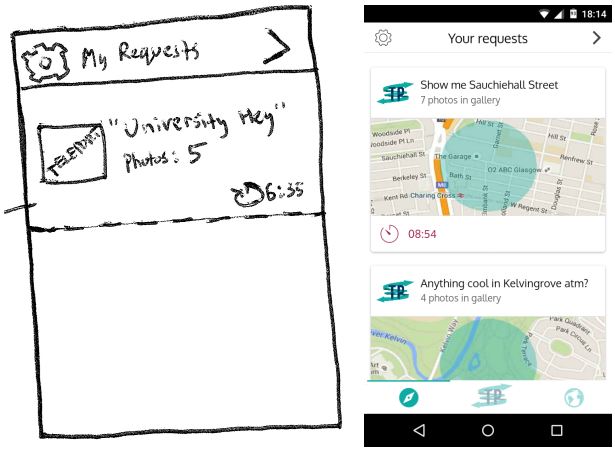
When users click the OK button, the request is sent, the popup window disappears and a toast notification is displayed in order to give users feedback about their action. See Figure 4.8. If the Cancel button is pressed, the toast notification displays “Request not sent!”. The toast notification was chosen in the implementation over another popup window with an OK button to decrease the number of clicks users have to perform in order to send a request, resulting in general better user experience.



(a) Sketch (b) Implementation

Figure 4.8: Confirmation of sent request

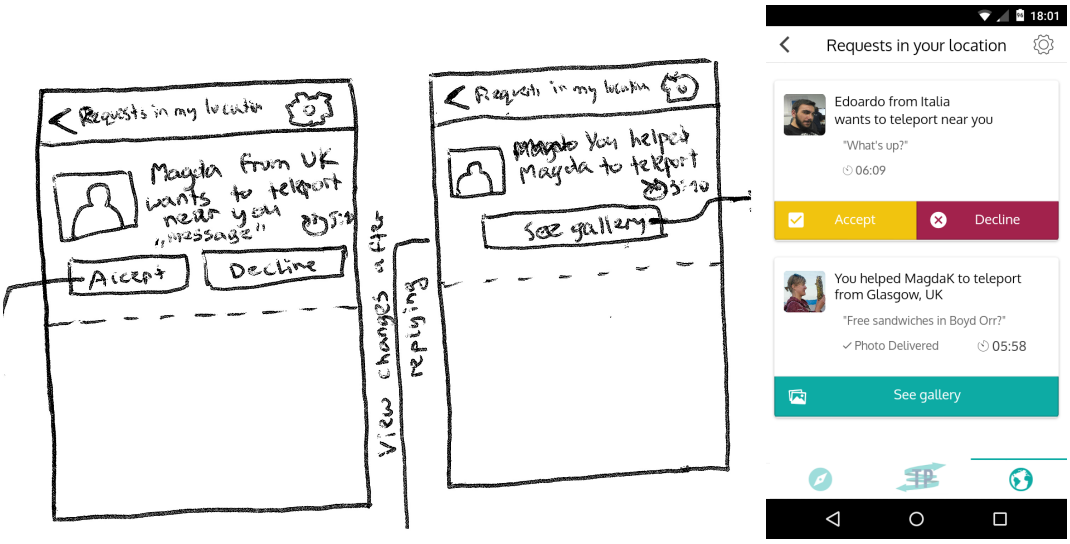
When users create a request, they can view them from the "Your requests" view. See Figure 4.9. The view consists of a list of cards, each card representing a request. It contains the heading, which is the message user entered while creating the request, the indicator of the number of photos in the gallery, a screenshot of the place in the map user decided to teleport to and finally, a timer. The screenshot feature was added after the final sketch had been created. We wanted the users to remember exactly the location they sent a request to and to be aware what photos they might expect to receive from that area. The button on the top left takes users to Settings View. Users can pull down the view to refresh the list. Swiping the view from right to left results in switching to the map view.



(a) Sketch (b) Implementation

Figure 4.9: My Requests View

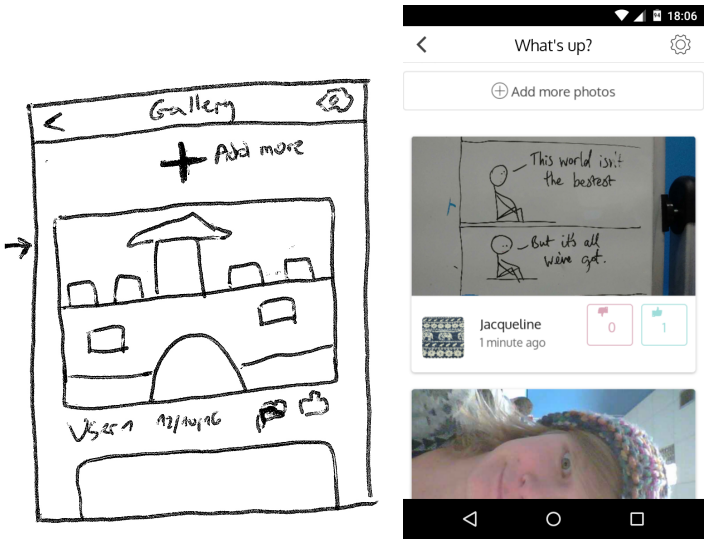
When users navigate to "Requests in your location" view, they see active requests in a 200m proximity from their location. The view consists of a list of cards. Each card represents a request. A request that has not yet been answered by the user yet shows title "UserX from Y wants to teleport near you". The message of the request and time left until it expires are visible below. Two buttons allow users to either accept the request (yellow Accept button) and take a photo or decline it (red Decline button) and remove the request from the list. Once a user replies to a request, the corresponding card changes the title to "You helped UserX to teleport from Y". The message of the request still can be seen, along with the "Photo Delivered" indication and a timer. The buttons change from Accept/Decline to a single "See gallery" button. Profile pictures of the requesters can be seen in the top left of every card. Users can pull down view to refresh the list. Swiping the view from left to right results in switching to the map view.



(a) Sketch (b) Implementation

Figure 4.10: Requests In My Location View

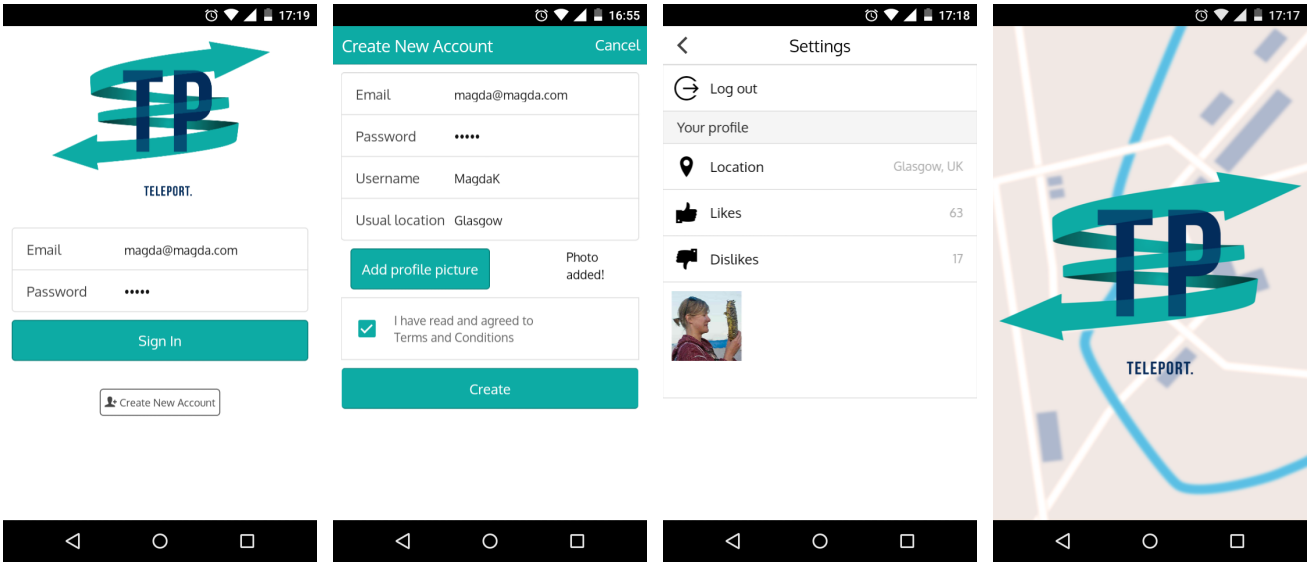
There exist two gallery views: one that can be seen by the author of the request and one that can be seen by users who have replied to the request. Figure 4.11 shows the latter one. The difference between these two is the button on the top allowing users to add more photos. In the final design, the gallery resembles the style of an Instagram gallery where each photo is shown in its full size. Under each photo the profile picture and username of its author can be seen along with an indication on how many minutes ago the photo was taken. The likes buttons are placed to the right from the information about the author. Likes buttons are explained in more detail in section 5.4.3. The latest photos appear at top of the list and the list updates itself when a new photo arrives. Users can also pull down to refresh the gallery.



(a) Sketch (b) Implementation

Figure 4.11: Gallery View

Views that can be seen in Figure 4.12 were not part of the final sketch however they were always considered as part of the application. Figure 4.12a shows the login screen where users have to provide email and password details in order to login. Figure 4.12b shows a modal registration view that can be accessed by clicking the "Create New Account" button. In the registration view, users provide their email, password, username, usual location, profile picture, agreement to the Terms and Conditions in order to create a new account. The Settings view visible in Figure 4.12c was designed in the earlier stage when the Balsamiq wireframes were created. In this view, users can update their usual location and view how many likes and dislikes their photos collect. At the bottom of the view users' profile pictures are visible. Figure 4.12d shows the splash screen of the application. The splash screen is displayed for a few seconds when the application starts.



(a) Login (b) Registration (c) Settings (d) Splash Screen

Figure 4.12: Other views

Chapter 5

Implementation

5.1 Technologies Used

5.1.1 Ionic

We chose to develop Teleport using Ionic [18], a mobile application framework. Ionic allows you to create mobile applications using Javascript and HTML. It offers a library of tools, gestures and CSS components optimised for mobile devices. Ionic provides cross-platform development, i.e. a single code of an application that can run on both Android and iOS devices. It is built on top of AngularJS [3] and Apache Cordova [4]. AngularJS provides the application's structure and supports communication between Javascript and HTML. Cordova is the base of the mobile development framework. We decided to use Ionic and develop a hybrid mobile application rather then writing a native Android or iOS code because it is easier to change and update, which was necessary for the iterative nature of the project. It also provides a set of predefined UI elements, which makes the application look great with minimal effort in terms of graphic design.

5.1.2 Firebase

Firebase [13] was chosen as the main backend provider and data storage. Firebase provides a real-time cloud database, works as Backend as a Service (BaaS) and stores data in a JSON format. It does not require writing any server side code since all its functionality is accessible through a Javascript API. It is also highly compatible with AngularJS and secure due to 2048 bit certificate SSL connection. Having little experience with databases, Firebase seemed like a highly powerful and easy-to-use solution.

5.1.3 Node.js and Amazon Web Services

A simple HTTP server was deployed using Node.js [19] on an Ubuntu instance running of Amazon Web Services [1] to handle the notifications feature. Node.js is an asynchronous event driven framework. We used it to listen to changes in the Firebase database and trigger Parse methods in order to send notifications.

5.1.4 Parse

Parse is a Mobile Backend as a Service (MBaaS) provider. In Teleport, we used one of its many functionalities - a push notifications system. It is one of the most popular in this category and handles a monthly quota of 1 million push notifications for free. It is highly compatible with Ionic through various customised plugins.

5.1.5 Plugins

A number of custom plugins were used to support application's features. These can be seen in Figure 5.1.

Plugin	Purpose	License
Cordova-screenshot [10]	To take screenshot of the location in the map users sends a request to	MIT License
Cordova-plugin-camera [7]	Provides an API for taking pictures and for choosing images from the system's image library	Apache License
Cordova-plugin-crosswalk-webview [8]	Makes the application use Crosswalk WebView instead of the System WebView	BSD License
Cordova-plugin-geolocation [9]	Provides information about the device's location, such as latitude and longitude	Apache License
Cordova-parse-plugin [6]	Enables communication with Parse and provides notifications functionality in native Android and iOS code	Apache License

Table 5.1: List of custom plugins used in Teleport

5.2 Frontend Architecture

Ionic provides a Model View Controller structure. The Figure 5.1 shows how it is applied in Teleport. The configuration and the main class of the application is implemented in App.js class. Communications with the server and the syncing incoming requests are handled in Services.js class. Every view has it own controller which is defined in Controllers.js. HTML provides view of the UI elements. Additional Log.js class is a logging system provided by the project supervisor which was incorporated inside the Controllers.js to track users' interactions with the application for evaluation purposes.

5.3 Backend Architecture

The backend of Teleport constitutes of a number of technologies described in section 5.1. Security was one of the priorities when developing Teleport. Application uses users' location in order to send requests to appropriate users. The requirement was to design the backend in a way so that the application does not track users, i.e. no information about users' location is stored in the database. To achieve this, the following approach was taken: notifications about newly added requests to the database are sent to all users, and their phones compare their current location with the location of the request. If the distance is smaller than 200m, notification is displayed, and if not, the notification is discarded. The diagram of the backend architecture can be seen in Figure 5.2. Communication between various elements of the architecture is marked with numbers in the diagram. The list below describes each of them:

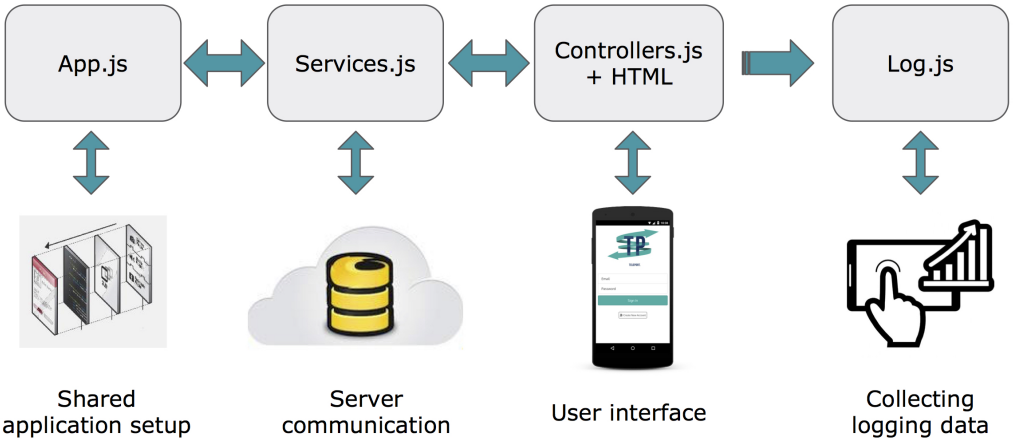


Figure 5.1: Diagram of the frontend architecture

1. **Client - Firebase - Client** User interacts with the frontend and sends requests, which are stored in the Firebase database. Firebase syncs data across all devices in real-time. Location comparison methods only allow certain users to see appropriate requests synced from the database. This is described in more detail in section 5.4.3.

2. **Firebase - Node.js** The HTTP server contains a reference to the Firebase instance and listens to any changes that are being made to it using Firebase API.

3. **Node.js - Parse** The HTTP server also contains a reference to the Parse instance. If changes in Firebase were detected, i.e. new request added or new photo added, Parse API methods are called in order to send a push notification. The notification is sent to all registered devices running Teleport. This is described in more detail in section 5.4.5.

4. **Parse - Client** The parse plugin on the client side compares the location of the request with the current location of the device. This is happening in native Java code because this operation runs in the background. This is described in more detail in section 5.4.5.

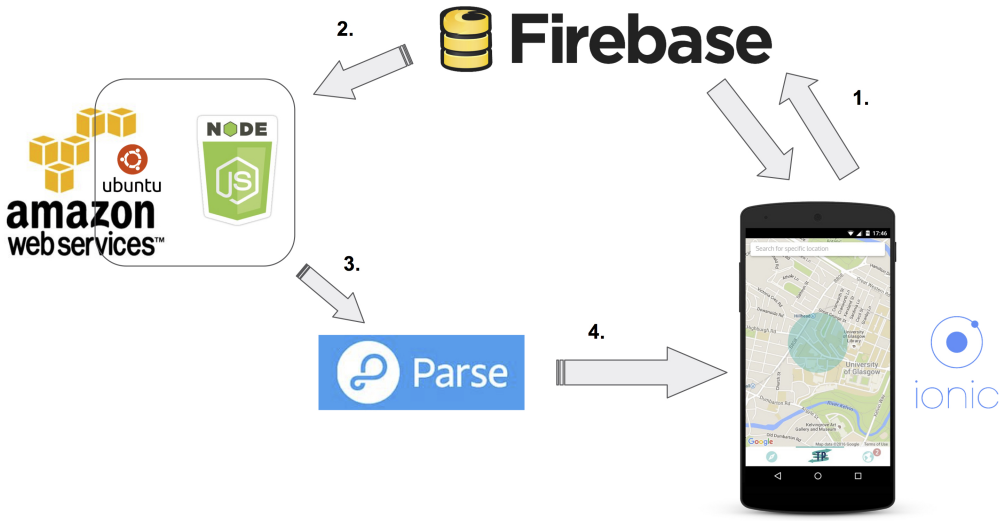


Figure 5.2: Diagram of the backend architecture

5.3.1 Database structure

Firebase stores data in a JSON format, which can be seen in Figure 5.3. For Teleport's purposes, 4 main parent nodes were created in order to store data that users create while interacting with the application. These are likes, photos, requests and users. Users store information about their profiles such as usernames, emails and profile pictures. Every user has a unique ID automatically assigned during registration. This ID is used as a key when storing the list of users. Requests, photos and likes are stored separately and use the timestamp of when a request is created, to allow for identification of data belonging to the same request. Photos are stored separately from requests because there is no point in loading image data while filtering requests. Images are only loaded when a user enters a gallery of a request. Likes are stored separately from photos and request because of the notification system which is described in section 5.4.5.

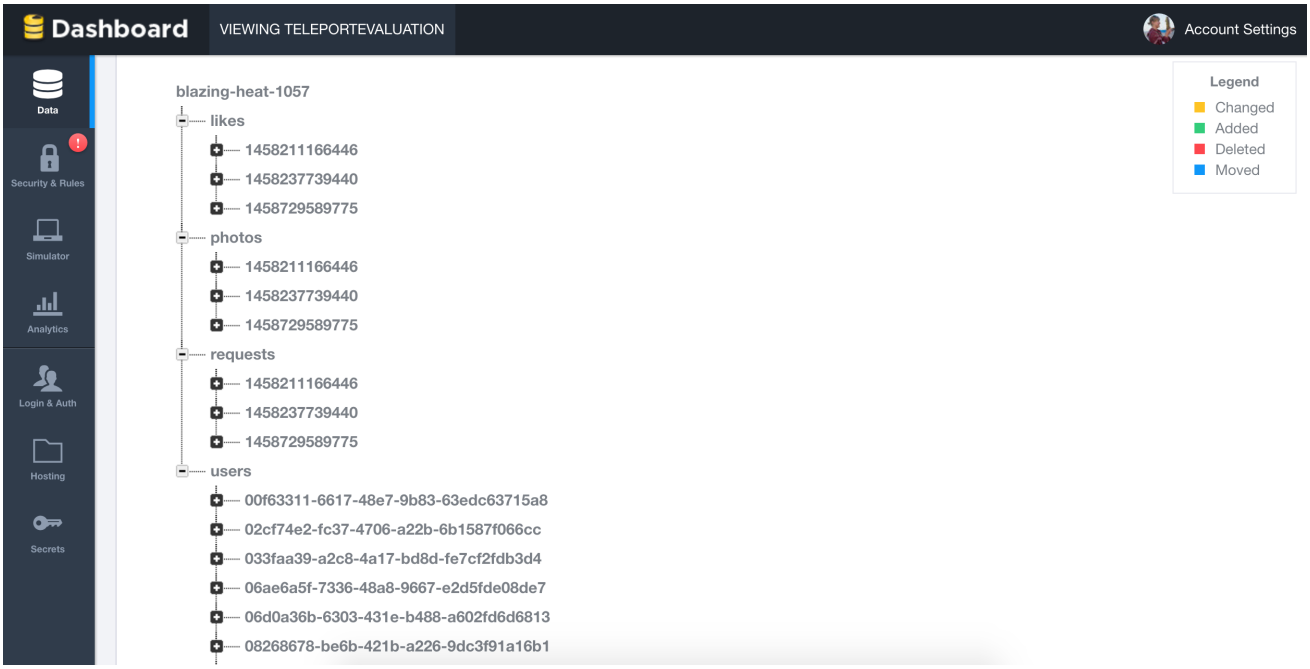


Figure 5.3: Screenshot of Firebase dashboard

5.3.2 Limitations

This solution provides a maximum level of security because no data about users' location is stored in the database. However we are aware of its possible limitation once Teleport is used by thousands or millions of users. It would be infeasible to send notifications to all users about all incoming requests when there are hundreds of requests made every minute. As this is outside the scope of the project, we did not have to consider this scenario. However there exists a simple way to decrease the number of requests sent to every device. Simply store information about users' location in terms of Northern/Southern and Eastern/Western hemisphere. This would decrease the amount of incoming notifications by 4 but locating users by this information would be impossible. Therefore, the system would still remain secure.

5.4 Functionalities

5.4.1 Registration and Login

In order to start using Teleport, every user has to create an account. The registration process requires a user to provide their email, password, username, usual location and a profile picture. Usual location can, for example, be Glasgow but there are no restrictions about this information and the user has freedom of choice on what he wants to write there. This information is only used while a user sends a request and someone receives it seeing "User X from Y wants to teleport near you". This can be seen in Figure 4.10b. The registration process is intended to be very simple, without any need of email or phone verification since this is not in the scope of the project.

Firebase provides a built-in functionality for authenticating users with email and password. The credentials are kept in a secure database behind the Firebase Authentication servers and stored securely using bcrypt.

5.4.2 Map

One of the most important features of Teleport, the ability to browse the map and precisely choose a location in order to request photos from there, uses Google Maps API. Google Maps was chosen because it provides a high compatibility with Ionic and Cordova location plugin. It is also the most well-known map provider therefore, users are familiar with how it works and looks. The API's Geometry library was used to draw the circle representing the 200m radius area that a request can be send to and to calculate distance between requests in the database and users' devices. To keep the circle always placed in the center of the map, a listener of the map object's idle state was added. Every time the map is moved, the central location in the map is updated and the circle's center is updated to that value. The map also contains a search box which allows to browse for specific locations. Google Maps API is available for free until exceeding 25000 map loads per 24 hours, which was enough for the project.

5.4.3 Sending and receiving requests

Creating a request

A user creates a request by navigating in the map to the desired location, pressing the Teleport button and adding a short message. The timestamp of this action provides a unique identifier of the request. A new entry is stored in the database and the timestamp is used as a key. Every request object contains the following fields: latitude, longitude, requester's ID, requester's name, requester's usual location, message, screenshot of the map, number of replies (initially set to 0), array of users who replied to the request (initially set to "none") and array of users who declined the request (initially set to "none"). The timestamp is also used to calculate the remaining time users have to view the request. Latitude and longitude are used to calculate distance between the request and user's phone.

Calculating the distance

Two ways of calculating the distance were taken into consideration - calculating the distance inside the code using Haversine formula and Google Maps API call. The Javascript representation of Haversine formula can be seen in Figure 5.4. On the other hand, the Google Maps API Geometry library call only takes one line of code and eliminates computational scope:

```
var dist = google.maps.geometry.spherical.computeDistanceBetween(l1, l2);
```

It is infeasible to perform Haversine formula calculations on the client side every time a user refreshes the list of requests. Therefore the API call was chosen as the final solution for calculating the distance between locations.

```
var rad = function(x) {
  return x * Math.PI / 180;
};

var getDistance = function(p1, p2) {
  var R = 6378137; // Earth's mean radius in meter
  var dLat = rad(p2.lat() - p1.lat());
  var dLong = rad(p2.lng() - p1.lng());
  var a = Math.sin(dLat / 2) * Math.sin(dLat / 2) +
    Math.cos(rad(p1.lat())) * Math.cos(rad(p2.lat())) *
    Math.sin(dLong / 2) * Math.sin(dLong / 2);
  var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
  var d = R * c;
  return d; // returns the distance in meter
};
```

Figure 5.4: Haversine formula in Javascript

Populating list of received requests

When a user enters or refreshes "Requests in my location" view, a number of methods are run in order to sync data from the database and display it on the user's phone. Firstly, received requests are stored as an array in a scope variable in order to make them accessible by HTML code:

```
$scope.receivedRequests = ReceivedRequests.all(myLat, myLng, myID, initImages);
```

ReceivedRequests is a service that takes care of syncing data from Firebase. Its *all* method takes user's ID, current location and the callback *initImages* function as parameters and returns an array of filtered requests. Four factors are taken into consideration while filtering the requests. The first being the distance between user's current location and the request's location. The second validates if the ID of the requester is not equal to the receiver's ID. The third checks if a request is still active, i.e. it was created less than 10 minutes ago. The fourth checks if the user have not declined this request before. Once filtering is completed, the callback function loads profile pictures of requester for every request in the list. The "Requests in my location" view gets populated using ng-repeat directive which creates a card in the list for every request in $scope.receivedRequests array.

Timers

Every request in the list has a timer indicating how much time a request has left until it stops being active. To implement timers angular-timer [2] directive was used. To calculate the value of the timer the value of the timestamp of a request is subtracted from a current timestamp. This value is subtracted from 600 seconds (10 minutes). The result is the amount of seconds left until a request expires.

5.4.4 Sharing photos

A user accepts requests by taking photos and sharing them with requesters and other people who replied to the same request in a single gallery. The Cordova camera plugin allows user to take a photo and returns it in a

Base64 binary format. The string representing the photo can be easily stored in a database of JSON format. The timestamp of a request is used as a key for the gallery of photos. Every photo's timestamp is used as its key within the gallery. This way, the request is synced with the gallery. When photos are loaded to a gallery, the Base64 string is loaded to the HTML image tag ({{img.image}} being the Base64 string):

```
<img ng-src="{{img.image}}" width="100%">
```

5.4.5 Notifications

Notifications are one of the most important features of Teleport. They instantly inform user about new requests and photos. Implementing the notifications was one of the biggest challenges of the project. To start with, notifications can arrive to the phone at any time, even when the application is closed. Therefore they could not be implemented in Javascript application's code. As this functionality runs in the background, a native code had to be created in order to handle notifications.

Extension of Parse plugin

Parse plugin [6] provides a code that establishes connection with a Parse instance and displays an Android notification. The plugin had to be extended so that the native code of the application is aware of the user's ID in order to determine if an incoming request is made by them or not. We added an extra parameter to ParsePlugin.initialize function so that the user's ID gets stored in Shared Preferences and can be accessed through the application's native code. Then ParsePluginBroadcastReceiver, a class that is responsible for receiving push notifications from Parse, got customised so that when it gets a notification about an incoming request, it starts a MyLocation Service.

MyLocation Service

In Android, a Service is an application component that can perform operations in the background. MyLocation service retrieves user's ID stored by the plugin and checks a device's current location using Google Play services location API. It retrieves data from the notification, i.e. request's latitude, longitude and author ID. It creates two location objects - the first being the user's current location, the second being the device's current location. If they are less than 200 meters away and the request was not made by the user of the current device, it displays a notification saying "UserX wants to teleport near you".

Notifications about incoming photos

Notifications about photos as opposed to notifications about request are not sent to every device. Instead, a Parse functionality called channels is used. Parse allows for the sending of notifications to a group of users who subscribe to the same channel. In Teleport, a user creating a request subscribes to a channel named after the timestamp of the request to provide uniqueness of the name. Then, every user who replies to that request subscribes to the channel too. So only this group of users receives notifications about new incoming photos to the request's gallery.

Server side of the notifications system

Firebase API provides a method *onChildChanged()* which listens to any changes made to the children of the specified node. In the Node.js server, two listeners were created - one for the Request node and one for the Photos node. When a change is registered to the Request node, information about the request, such as latitidue, longitude and requester's ID, are passed to Parse notification builder and sent to all registered devices. However when a change is registered to the Photos node, the notification containing a message "New photo added!" is sent to all devices subscribed to the channel of that request.

5.4.6 Likes

Every uploaded photo has its own "Likes" entry in the database. Likes have two fields - "thumbsUp" and "thumbsDown" to track how many likes and dislikes a photo receives and which users react to the photo. There exist three possibilities in regards to a likes scenario and they are handled on the client side in order to display appropriate buttons and these can be seen in Figure 5.5. Figure 5.5a shows how the buttons look if the current user is not the author of the photo and did not use like functionality for that current photo. The buttons are active and the user can either click the "thumb up" or the "thumb down" button. After the user clicks one of them, they both change colour and become inactive, which can be seen in Figure 5.5b. The third option is that the current user is the author of the photo. In this scenario, buttons are inactive and faded out so that the user cannot click on them. These can be seen in Figure 5.5c. Likes sum up to a total score of a user's profile so that he or she knows if other users liked their photos or not. This reward system encourages people to take relevant photos.

(a) Not the author, can click

(b) Not the author, already clicked

(c) Author, cannot click

Figure 5.5: Likes buttons

Chapter 6

Evaluation

In order to evaluate how Teleport was used, data logging, field trials and interviews were conducted. Firstly, a logging system was added to the application in order to gather quantitative data on how the application was used. Appropriate logging events tracked how participants of the evaluation were using the application. On the 12th of February an advertising website [25] was launched where people could read information about the evaluation and sign up for it. On the same day, the application was published on Google Play Store [26] so that everyone could start using it. The information about launching the application was shared on social media websites such as Facebook and Twitter. After a week, 15 people were signed up for the evaluation and about 35 people installed the application in total. This was enough to make the evaluation valuable. The field trial officially started on the 21st of February and participants agreed to use the application for a week and to be interviewed afterwards. This chapter explains in detail the process and the results of the evaluation.

6.1 Field Trial

From all people that signed up for the evaluation, 15 participants were selected; 12 male, 3 female, aged between 19 and 24. 13 of the participants were students at the University of Glasgow and 2 were young professionals. All of the participants lived in Glasgow. The criteria which identified them as appropriate participants were the age, the place of living and a compatible mobile device. More specifically this meant, young people currently living in Glasgow with Android phones were selected. The terms and conditions of the evaluation were made available to them on the website and they agreed to them by filling out a Google Form and singing up for the evaluation. As they used the application for at least 7 consecutive days, logging data was collected and interviews were scheduled afterwards using a Doodle poll [12]. The next sections cover the logging and interview outcomes.

6.2 Logging

The logging system recorded data on what views were entered, what requests were sent and whether a user replied by taking a photo or declined a request. Figure 6.1 shows when and how often users were interacting with the application. According to the data, participants of the evaluation made a total of 387 requests and sent 287 photos. Assuming that every photo belonged to a different request, there were over 100 requests that did not get any replies. Only 8 requests were declined, which indicates that this functionality was not very popular and can lead to a conclusion that requests were not answered because users did not want to reply to them, but because they did not notice them on time or because there was nobody in the location they sent a request to. Users liked photos 95 times and disliked them only 22 times.

Thanks to this data, we know that participants used the application a lot. 387 requests divided by 15 participants and 7 days of evaluation gives almost 4 requests made per day. Although this data gives an interesting overview on how users interact with the application, the size of the experiment was too small in order to perform an adequate quantitative analysis. Our main focus is therefore on the qualitative data gathered during the interviews that followed the aforementioned field trials.

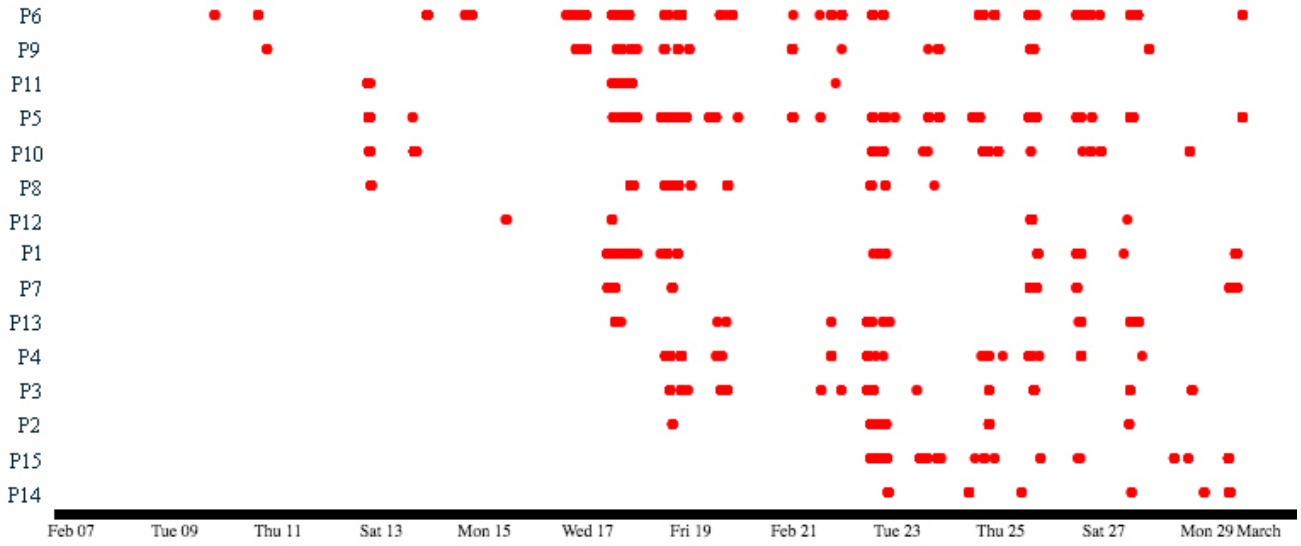


Figure 6.1: Data visualisation from logging

6.3 Interviews

Each interview was about 15-20 minutes long and was carefully transcribed in order to perform an analysis of the data. A spreadsheet gathering all the data together was created in order to identify trends and see differences in users' experiences while using the application. Topics we were interested in while analysing the data are linked with the background research and motivation of the project. Therefore we focused on these key issues:

- Why did they use the app? What was the motivation behind its use?
- How did they use the app? What did they use it for?
- What was their opinion on the ephemerality aspect?
- What did they think about the locality aspect?
- Did they use the short message functionality? How?
- What was their experience? What was different about communicating with others through the application?
- How would they use the app in the future, providing that there are many users?

Overall, participants were given a lot of freedom in the interviews, which gave us information about different topics, such as unexpected ways of using the application or great ideas on how to improve the application. Subsequent sections cover our findings on all of these.

6.3.1 Reasons and motivation for using Teleport

First of all, the majority of participants noticed that there was no other application like Teleport - or that at least they have never heard of any. Everyone liked the novelty of the idea. Participants strongly appreciated the possibility of being able to request desired content. P10 noticed that all other available social media applications provide content in a from of a stream of consciousness and in Teleport you only receive content when you ask for it. People were interested in seeing places or how some of them named it "to virtually go places". This is exactly what we had in mind while making the application. Another recurring theme was the ability of to see what is happening somewhere in the moment or when someone their mind about going to a place. People also reported that Teleport allowed them to see events they could not currently take part in. Another reason was the ability to get real pictures of places. Users valued authentic photos they could get via Teleport more than images they could find on Google or in other stock images collections. These, according to some participants, are always edited or filtered out by media and do not show how a place really looks, as opposed to photos you can get in Teleport. Furthermore, people also wanted to send photos to other people and provide them with information about the place they are in. The will to interact with people was another reappearing theme. Participants wanted to share what they were up to and see if other users would like their photos. People were interested in showing what they are experiencing in the moment and also in getting people to share their experiences. Moreover, many of them used Teleport with their friends to share pictures throughout the day. Some of them got a feeling of instantaneous connection with the world while using Teleport. Others described it as a very down-to-earth way of communicating with people.

6.3.2 Ways of using the application

Most of the participants of the evaluation were disappointed by the size of the experiment which did not enable them to fully use the application. They wanted to see places like New York or Paris but knew it was impossible since all the current users were based in Glasgow. P14 realised about this fact in a very interesting way. Without pointing the circle on the map to the area around the Eiffel Tower, he sent a request with a message "Pariiis!". He then received back a photo of the famous French monument and was excited that someone in Paris was using the app. However as it turned out during the interview, the location of the request was set to an area of university campus and someone had taken a photo of a picture of Eiffel Tower he found online.

P3 was sending a lot of requests in random areas of Glasgow but could not get any replies. Therefore he decided to start sending requests in his exact location to see what people around him were doing. He later stated that Teleport could be used to get people from your area to do something together. For example he would send a request to his current location and ask "Would anyone like to go for a coffee?". This way, he could meet new people in his area he had not previously met. Other participants were also of the opinion that using Teleport gets you to meet new people. Because the user group was small, users were often getting photos from the same people. From the content they shared they could learn about people, what they do and what they like.

P5 sent a request on Saturday afternoon to the Sir Alwyn Williams building but did not know about the Cyber Defence Exercise that was taking place there that weekend. He got a reply with a picture of free food and decided to join the hackers and sample the free food.

P1 while being in the library got a request saying "How busy library is?". He decided to respond instantly, feeling responsible to deliver answer to that question, since library was full of people and he did not want that person to arrive and be disappointed. Not knowing how to show that in a photo, he replied in a way that can see in Figure 6.2.

P8 was mainly using Teleport in the evenings because he did not want any distractions during the day while he was at work. One evening he decided to look for his flatmate using Teleport. He sent out requests to the West

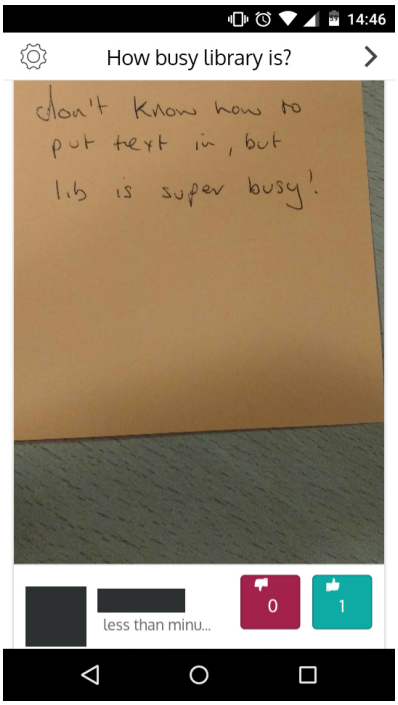


Figure 6.2: Reply provided by P1(profile picture and username were removed to provide anonymity)

End area hoping his flatmate was somewhere close by. After a few attempts, one of the requests got answered by the flatmate. This is how P8 figured out where his friend was.

Overall, some participants happened to use Teleport in unique ways. However there were some common themes in using the application, which were recurring for a number of participants as well. In general most people when sending a request to a location, were not interested that much in the place itself but in the way someone present in the area would show it to them. Sending random requests to see how people would reply to them it seemed like a common motive. People sent requests to see what was happening in the selected area. Participants that used the application with their friends would communicate with them asking what are they up to with the intention of joining them later if they were for example at a pub. Making jokes and sending funny pictures was popular both between friends and strangers who were using the application together. Some of them were interested in a specific content such as view out the window or how a place looks like at night. Others admitted that they used Teleport for procrastination purposes or to just have a break from studying. Finally, two groups were identified; one of them preferred to send requests while the other was mostly interested in replying with photos.

6.3.3 Opinions on the ephemerality aspect

Interestingly, the participants had considerably discordant opinions about how the ephemeral nature of communication in Teleport affected their experience. On the one hand some people claimed that 10 minutes is not enough because users tend to have their phones muted and therefore could not see notifications on time or missed incoming photos. They also believed that due to the insufficient amount of users at the current stage of the application, the time limit prevented them from effective communication. They also found it frustrating to miss requests due to the timeout and it was annoying not to be able to see photos they received because they saw notifications too late. Some people also believed if the time limit was more than 10 minutes they would get more photos and would be able to see more perspectives of a requested location. On the other hand, proponents of the 10 minutes restriction pointed out that it boosts the creativity and as a result they can get more interesting photos. They also argue that content shared in Teleport by its definition is meant to be temporal and keeping photos for a longer period of time would make it boring. According to them, ephemerality makes the application more fun and distinguishes it

from applications like Instagram. They also noticed that time restriction makes users reply instantly and people would not care to reply if a request was active for days. Ephemerality provides "instantaneous connection"(P12). Most of the participants agreed that the time limit enables you to see how a place looks right now and they named it as one of the most important and unique features of Teleport. However, P13 pointed out that although the time restriction imposes getting the most current view of a place, in his opinion irresponsible users might still send irrelevant photos, e.g. photos of a wall. P9 claims that while using Teleport, users are expecting to see what is happening somewhere in that instant, rather than a standard view of the place. On the contrary, P2 believes that when requesting a photo from a popular place the time limit does not matter because that place will always look more or less the same anyway. P14 goes even further to say that he wants to see places from different people's perspectives but does not care about the time in which the photos were taken. The majority however disagrees, saying that without the ephemerality aspect you could just look up pictures using Google and that with Teleport you can get the most current view of a place. They also believe the time limit would work better if there were many users of the application which would guarantee getting a photo every time they send a request. About half of the participants also suggested they would like to be able to adjust the time limit themselves within certain bounds. However, P10 argued that more functionality would prejudice the novelty of the application and that the fixed time limit is what makes it fun.

6.3.4 Opinions on locality aspect

In general, participants liked the idea of requesting photos from certain locations. They said it provides a unique perspective of a place, something you could not see anywhere else online. Some of them pointed out that there are no other applications which allow you specifically designate an area in order to receive photos from there. They thought it was a great idea that Teleport enables you to do just that. For some of them, the 200m radius fitted perfectly with places they were interested in (P3, P6). Others pointed out that the 200m radius restriction prevents a possible overflow of information. Some people claimed that 200m might not be enough because there is not enough users at the moment and they did not get replies to all their requests. Others said 200m would only work in a crowded area. On the other hand, some participants were concerned about people getting requests in neighboring areas to which the requests might not be relevant. P13, at one time was in the Boyd Orr building and received a request that was aimed at the Sir Alwyn Williams building. He felt disappointed that his reply would not be relevant to what the requester asked for. The majority agreed that any radius above 200m would make people care less about replying with photos because that big an area is too general and they would not know what the requester would want to see. However, P8 believed that the bigger radius would get you more photos. Almost every participant pointed out that they would like to be able to adjust the radius themselves within certain limits. Interestingly, they also said that for places nearby or known to them they would like the area to be more precise and that for distant, unknown places they would like to set a bigger radius. This finding is described in more detail in section 6.4.

6.3.5 Opinions on the messages functionality

Almost every participant agreed that the possibility of adding a short message to a request narrows down what one is interested in. They also said it makes a request more specific and pinpoints what a person would like to see. P9 said it allows to precisely designate a location within the 200m radius area. Some said the message is essential and that a request is meaningless without it. In general, people noticed that the message did influence what photos other users were taking. People were getting creative while making the messages according to P11. For example, by adding messages like "What book are you reading at the moment?" or "Cool thing you did recently". However some participants noticed the message only made a difference if it was specific. It would only influence someone while taking a photo if it conveyed an actual message (P1). However, P13 noticed that sometimes the message might be not understood correctly. Participants who used Teleport with their friends said it was good for making jokes and sending funny messages.

6.3.6 General experience of using Teleport

The general feedback was very positive and participants enjoyed using Teleport. They said it was fun to use it and it was exciting to receive requests and photos. Some of them felt like belonging to a community therefore felt if they reply to a request, other people will reply to their requests in the future. P13 described it as "help a brother out". P10 felt like he let other users down if the photo he sent was disliked. He therefore quickly learned only to take relevant photos. Users felt motivated to reply in an interesting way. P3 described using Teleport as "a conversation without words". People liked the randomness of the communication; "you never know what you are going to get"(P11).

6.3.7 Teleport in the future

Because of the size of the experiment, users could not benefit from what Teleport intends to provide - an instant view of any place in the world. With around 30 users in the UK, 10 in Poland and a few others around the world, participants mainly sent requests in the Glasgow area because the probability of getting a reply was the highest. We asked participants how they would use Teleport in the future, providing that there would be many users all over the world. The answers ranged from simply looking up how a potential holiday destination looks to Teleport being "an invisible superpower". Some said it would be useful to see what events, e.g. parties, concerts, football games, look like before going there. Some would like to see popular places such as New York's Times Square or the Eiffel Tower in Paris (P14). Others would like to see places they might never be able to see, such as the Fiji Islands (P12). P13 would like to see photos of beautiful sunsets around the world. P4 would use Teleport to confirm information shared by other people, which he described as "an invisible superpower". P9 imagined journalists using the application and normal people willing to see information not filtered by media. This finding is described in more detail in section 6.4.

6.3.8 Technicalities

Feedback received about the usability of the application was highly positive. Participants found the application easy to use, they liked the idea of three screens and they found the design appealing. The most enjoyable feature was the map and the ability to navigate to a specific location. P10 liked that the application is "map-centric". Users also gave positive feedback for the likes and dislikes functionality, which encouraged them to take better photos. Participants reported issues with the Teleport button; many did not realise the first time the center tab button with the Teleport logo on it is the button you use to send requests.

Interestingly, none of the participants expressed any concerns regarding how the application knew their location and if they were being tracked. The advertising website said that the application does not track users. However this was not explained in more detail. Therefore participants did not know how the application was selecting certain users to receive requests based on their location.

6.4 Main findings

6.4.1 It is about the people, not the places

One of the most interesting phenomenon we identified while analysing the interviews was that participants' main interest, while sending requests and receiving photos, was not in an actual view of the place they requested but in the perspective that others captured the place from. They described using Teleport as "being able to see world

through other people's eyes" (P12). It was not just about getting a photo of a location but the way someone will interpret the request (P10). Many participants said that the biggest value of the photos they received, was that someone decided to devote a few seconds of their life in order to take a photo and share it through Teleport because they requested them to do so. P4 felt like the requests he received were personal to him. Even if the message said "What's happening in the area?", he interpreted it as "What are you doing at the moment?".

6.4.2 Different perception of locality

Users were repeatedly reporting that they would like to be able to adjust the radius of the location they send a requests to. We identified a recurring theme in their reasoning behind that statement. Most of the users said they would like to be able to set the radius to smaller for places near them or known to them and set it bigger for distant and unknown locations. This shows how the definition of locality changes according to how well we know the place we want to see or how far from us the place is. A similar finding was identified in the Columbus study [42], where users reported that the smaller the area they could see pictures from, the more local they seemed to them. Teleport participants suggested that if they send a request to a distant place, e.g. Berlin, they do not know if anything interesting is happening there, therefore they would like to set a bigger radius and get photos from the whole city. However when they send a request to a location they are familiar with, e.g. Glasgow West End, they would like the area to be more specific and pointing to a precise location.

6.4.3 Serious use in the future

Participants reported they were using Teleport because it was fun. Sharing trivial content with people around campus made them feel happy. In few cases participants reported they used Teleport for procrastination purposes. Others said there is no specific reason for using in the same way as other social media applications, such as Snapchat or Facebook. However multiple participants said that in the future, when Teleport has many users, the application could be used for much more serious purposes. A few suggested that Teleport would provide the most accurate information about events such as riots or demonstrations. They said information we find online is often filtered and altered by media, which is not the case for Teleport. P9 suggested that if Donald Trump's supporters could send requests to Syria and see families struggling with the civil war, they would change their opinion on bombarding these areas. Participants see Teleport as a powerful tool that could change users' opinions on various topics by providing an actual and true view of places.

6.5 Outcome

Great amount of feedback has been collected during the evaluation. Despite the size of the experiment, the quality of data collected was greater than had been expected. Users truly enjoyed using the application and many suggested that Teleport could be the next big thing on the mobile applications market. However certain suggestions for improvement were identified and these are discussed in the next chapter.

Evaluation Documents

The Information Sheet can be seen in Appendix A.
The Consent Form can be seen in Appendix B.
The signed ethics checklist can be seen in Appendix C. The recordings and transcriptions of the interviews can be found in the submission directory.

Chapter 7

Future

7.1 Suggestions from Evaluation Feedback

7.1.1 Teleport Button

Although users were given instructions on how to use the application on the advertising website [25], many reported they struggled with finding out which button allows them to send requests. It was suggested that the tab bar button with the Teleport logo should be redesigned so that it looks more like an action button. Some suggested an additional button could be placed somewhere within the map. Others said sending requests by clicking on the area on the map could be another way to do it.

7.1.2 Adjustable Parameters

The majority of users said they would like to be able to adjust the area and the time limit themselves. As described in section 6.4 the perception of locality changes across locations depending on whether a user knows the place or not. Users suggested they would like to set the radius themselves within certain limits. P9 suggested it would be even better if a user could draw a shape on the map to encompass the area he is interested in. Others suggested choosing the time limit yourself would a be a useful feature as well.

7.1.3 Heat Map

A few users complained about the fact that sending requests is “hit or miss” (P11), because you never know if there are other users of Teleport in the area or not. Some of them suggested a heat map solution. This would result in having all users of Teleport marked on the map with a dot or other marker. Thanks to the heat map, users would know where sending a request will be successful.

7.1.4 Grouping Notifications

Some users complained about the notifications cluttering their notification bars. They suggested notifications about incoming photos should be grouped together. In a situation when a user receives 3 notifications about one request and 2 notifications about another request, he would see only 2 notifications grouped together instead of 5 single notifications.

7.1.5 Private Requests

Some participants suggested that once Teleport becomes popular it would be difficult to get requests only from their friends. Therefore they would like to be able to have a list of friends and designate requests only for them in certain situations.

7.2 Further Improvements

7.2.1 Extending Cross-Platform Support

Ionic provides a cross-platform implementation of applications. Teleport however does not take advantage of this for the purpose of the project because of the native code that had to be written in order to handle the notification feature. This can be easily fixed by writing a corresponding code for iOS devices in Objective C or Swift. This would result in the possibility of accessing a larger number of users and thus making Teleport more popular.

7.2.2 Videos

Sometimes what you want to share with someone who has requested a view of your current location cannot be captured in a single photo or even a series of photos. Short videos could be added as a way of replying to a request along with the existing option of taking photos. The video would have a time limit and could only be 10 seconds long at most, similar to Snapchat.

7.2.3 Sharing interesting photos to a publicly visible profile

Another idea would give users the possibility of sharing great photos they receive to their publicly visible profile, so that if you receive a photo you really like, you can choose to share it. It would appear in your profile and the author of the photo would get notified that someone decided to share their photo. This way, users could create unique galleries of beautiful places they requested to see. As a result, memories of those great photographs would stay alive for longer.

7.2.4 Comments or Chat

Since the evaluation discovered that people felt like creating temporal communities with the people who were responding to the same request, it would be a nice feature to be able to establish some closer contact with this community by adding comments or having a group chat with the people, so that you could ask them questions about what they shared with you in the photos.

Chapter 8

Outcome

8.1 Idea

The most important outcome of the project is the idea itself. The ability of requesting photos from specific locations is something novel and there are no other applications that allow users to do so. This was greatly appreciated by the participants of the evaluation and all other users.

8.2 Advertising Webpage

As a way to promote Teleport and inform people about the evaluation, an advertising webpage was created and deployed using Amazon Web Services. A domain was purchased and the website can be accessed though the following link: http://teleportapp.co.uk/

8.3 Published Application

Teleport has been successfully published to the Google Play Store [26]. Currently, the application has in total 101 downloads. Figure 8.1 shows which countries people installed Teleport in. Without any advertising campaign, Teleport managed to be installed and used by users across five continents within less than two months since it was published. Direct link to the application: https://goo.gl/8Rf3y8

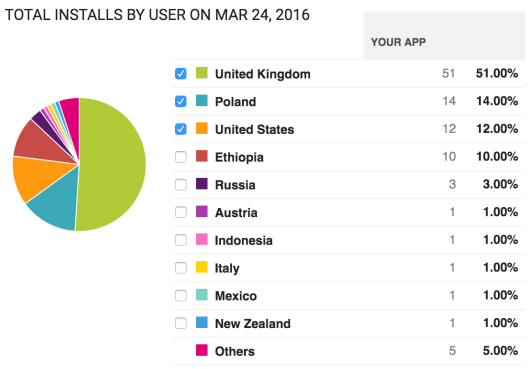


Figure 8.1: Total installs by country

8.4 Evaluation and Analysis

The evaluation of the project was very successful. It has given great hints on future improvements which will potentially make Teleport a very popular application. However, this process would have to be repeated once the user base is larger so that more quantitative data is available and even better improvements could be done to the application.

Chapter 9

Conclusion

9.1 Summary

The objective of this project was to develop a location-based photo request application and evaluate its use. After many iteration cycles the idea was finalised and requirements were set. The final product of the project met all requirements described in Chapter 3. The application was published on Google Play Store and used by over 100 users. The evaluation study demonstrated that the application is highly usable and reliable. However some further work is required in order to make Teleport even better. I strongly believe Teleport has the potential to become as popular as applications it was inspired by, such as Snapchat or Instagram.

9.2 Reflection

Creating Teleport was an extremely interesting and rewarding experience. The thing I liked the most about it was the complexity of the whole process that the project took me through. Firstly, brainstorming about different ideas, creating multiple designs, researching about other mobile applications and communication concepts. This made me look at technology people use everyday, such as Snapchat, from a completely new perspective. Furthermore, learning how to create an application using Ionic has been an extremely interesting process and I will definitely use Ionic for my projects in the future since I have gained this very tangible skill. The evaluation was a completely new experience to me; planning the interviews, recruiting participants, interviewing people, transcribing and analysing their feedback. This was something I have never done before and truly enjoyed. All steps of the process were extremely fascinating and allowed me to learn a lot. Finally, Teleport became a very personal project throughout this year and I am very grateful and happy I had the chance to create it. All things considered, I did not expect such an amount of positive feedback and high interest which Teleport gained in the last months. Being rewarded like that for work I have done myself is by far the best experience of my degree.

9.3 Acknowledgments

I would like to thank Mattias Rost for his invaluable guidance and passionate involvement with the project.

Appendices

Appendix A

Information Sheet

The aim of this experiment is to investigate the usage trends of a location based photo sharing app. More specifically I want to investigate how users interact with each other and what photos they take while using this app. Users are crucial for the app. They use it to communicate between each other, and this is why I need to run the experiment.

The experiment will take 1 week to complete.

This app has been developed at the University of Glasgow Computing Science department in a student project. I have developed this app to study how it is used. For this purpose, quantitative data will be collected from the app and upload it to Computing Science Department's servers for analysis. The data we collect includes: your username; what screens you visit in the app; what times you use the app; what request and photos you sent; and information about your connectivity. The data will not be shared with anyone outside the project. As far as possible, your participation in the study will be kept confidential and the data will be made anonymous. The outcomes and findings of this research will be made available in the form of academic publications and presentations.

At the end of the experiment, you will be interviewed, and you will receive a reward of 10.

All results will be held in strict confidence, ensuring the privacy of all participants. No personal participant information will be stored with the data. Online data will be stored in a password protected computer account; paper data will be kept in a single-occupant locked office.

Your participation in this experiment will have no effect on your marks for any subject at this, or any other university. Please note that it is the app, not you, that is being evaluated.

You may withdraw from the experiment at anytime without prejudice, and any data already recorded will be discarded.

By installing and using this app you are participating in our research. If you are under 16 years of age, or you do not agree to participate, please stop using this app. You can contact me at any time if you would like further information: 2037342k@student.gla.ac.uk

Appendix B

Consent Form

The aim of this study is to investigate the usage trends of a location based photo request application. I am running this study in order to analyse user behavior in regards to this application.

The experiment takes 1 week to complete.

You participated in the study a week ago, by agreeing to the Terms and Conditions of the evaluation, included on project's website, and installing Teleport app on your phone. You then used the app for a week. While you were using the application, some quantitative data was collected about your use of the application, including: timestamps, location of requests you made, requests you replied to, requests you declined, visited screens in the app, likes and dislikes. Photos you took are also stored, but their content will not be used in the evaluation.

Today you'll be interviewed about your experience while using the app. At the end of the interview you will be rewarded 10. Please write I received 10 for my participation in the box below.

All results will be held in strict confidence, ensuring the privacy of all participants. No sensitive personal information will be stored about any participants. Online data will be stored in a password protected computer account and paper data will be kept in a single-occupant locked office.

Please ask questions at any time if you need to, otherwise you are not required to check in with us at any point until the study is over. Your participation in this experiment will have no effect on your marks for any subject at this, or any other university. Please note that it is the application, not you, that is being evaluated. You may withdraw from the experiment at anytime without prejudice, and any data already recorded will be discarded.

If you have any further questions regarding this experiment, please contact:
2037342k@student.gla.ac.uk (the author) or
Mattias.Rost@glasgow.ac.uk (project supervisor)

I have read this information sheet, and agree to voluntarily take part in this experiment:	
_____	_____
Name	E-mail
_____	_____
Date	Signature

Appendix C

Ethics Checklist

School of Computing Science
University of Glasgow

Ethics checklist form for assessed exercises (at all levels)

This form is only applicable for assessed exercises that use other people ('participants') for the collection of information, typically in getting comments about a system or a system design, or getting information about how a system could be used, or evaluating a working system.

If no other people have been involved in the collection of information, then you do not need to complete this form.

If your evaluation does not comply with any one or more of the points below, please contact the Department Ethics Committee for advice.

If your evaluation does comply with all the points below, please sign this form and submit it with your assessed work.

---

1. Participants were not exposed to any risks greater than those encountered in their normal working life.
   Investigators have a responsibility to protect participants from physical and mental harm during the investigation. The risk of harm must be no greater than in ordinary life. Areas of potential risk that require ethical approval include, but are not limited to, investigations that occur outside usual laboratory areas, or that require participant mobility (e.g. walking, running, use of public transport), unusual or repetitive activity or movement, that use sensory deprivation (e.g. ear plugs or blindfolds), bright or flashing lights, loud or disorienting noises, smell, taste, vibration, or force feedback

2. The experimental materials were paper-based, or comprised software running on standard hardware.
   Participants should not be exposed to any risks associated with the use of non-standard equipment: anything other than pen-and-paper, standard PCs, mobile phones, and PDAs is considered non-standard.

3. All participants explicitly stated that they agreed to take part, and that their data could be used in the project.
   If the results of the evaluation are likely to be used beyond the term of the project (for example, the software is to be deployed, or the data is to be published), then signed consent is necessary. A separate consent form should be signed by each participant.

   Otherwise, verbal consent is sufficient, and should be explicitly requested in the introductory script.

4. No incentives were offered to the participants.
   The payment of participants must not be used to induce them to risk harm beyond that which they risk without payment in their normal lifestyle.

5. No information about the evaluation or materials was intentionally withheld from the participants.
   Withholding information or misleading participants is unacceptable if participants are likely to object or show unease when debriefed.

6. No participant was under the age of 16.
   Parental consent is required for participants under the age of 16.

7. No participant has an impairment that may limit their understanding or communication.
   Additional consent is required for participants with impairments.

8. Neither I nor my supervisor is in a position of authority or influence over any of the participants.
   A position of authority or influence over any participant must not be allowed to pressurise participants to take part in, or remain in, any experiment.

9. All participants were informed that they could withdraw at any time.
   All participants have the right to withdraw at any time during the investigation. They should be told this in the introductory script.

10. All participants have been informed of my contact details.
    All participants must be able to contact the investigator after the investigation. They should be given the details of both student and module co-ordinator or supervisor as part of the debriefing.

11. The evaluation was discussed with all the participants at the end of the session, and all participants had the opportunity to ask questions.
    The student must provide the participants with sufficient information in the debriefing to enable them to understand the nature of the investigation.

12. *All the data collected from the participants is stored in an anonymous form.*
    All participant data (hard-copy and soft-copy) should be stored securely, and in anonymous form.

Module and Assessment Name LEVEL 4 PROJECT

Student's Name MAGDA KOWALSKA

Student's Registration Number 2037342

Student's Signature Magda Kowalska

Date 28.03.2016

Bibliography

[1] Amazon web services. http://aws.amazon.com/. Accessed: 2016-03-24.

[2] Angular-timer directive. https://goo.gl/ypLBqR. Accessed: 2016-03-25.

[3] Angularjs. https://angularjs.org/. Accessed: 2016-03-24.

[4] Apache cordova. https://cordova.apache.org/. Accessed: 2016-03-24.

[5] Balsamiq. https://balsamiq.com/. Accessed: 2016-03-19.

[6] Cordova-parse-plugin. https://goo.gl/WZNaz8. Accessed: 2016-03-25.

[7] Cordova-plugin-camera. https://goo.gl/L0SO4t. Accessed: 2016-03-25.

[8] Cordova-plugin-crosswalk-webview. https://goo.gl/2FQNOl. Accessed: 2016-03-25.

[9] Cordova-plugin-geolocation. https://goo.gl/3IAfx5. Accessed: 2016-03-25.

[10] Cordova-screenshot. https://goo.gl/fTQHHr. Accessed: 2016-03-24.

[11] Design seeds. http://design-seeds.com/. Accessed: 2016-03-25.

[12] Doodle. http://doodle.com/. Accessed: 2016-03-21.

[13] Firebase. https://www.firebase.com/. Accessed: 2016-03-24.

[14] Flickr. https://www.flickr.com/. Accessed: 2016-03-16.

[15] Flickr map. https://www.flickr.com/map. Accessed: 2016-03-16.

[16] Google street view. https://www.google.com/maps/streetview/. Accessed: 2016-03-16.

[17] Instagram. https://www.instagram.com/. Accessed: 2016-03-16.

[18] Ionic framework. http://ionicframework.com/. Accessed: 2016-03-24.

[19] Node.js. https://nodejs.org/. Accessed: 2016-03-24.

[20] Panoramio. http://www.panoramio.com/. Accessed: 2016-03-16.

[21] Periscope. https://www.periscope.tv/. Accessed: 2016-03-16.

[22] Snapchat. https://www.snapchat.com/. Accessed: 2016-03-16.

[23] Star trek transporter. http://goo.gl/Yolo5g. Accessed: 2016-03-25.

[24] Stargate transportation rings. http://goo.gl/tRr5Df. Accessed: 2016-03-25.

[25] Teleport. http://teleportapp.co.uk/. Accessed: 2016-03-21.

[26] Teleport on google play store. https://goo.gl/NDZ1R6. Accessed: 2016-03-21.

[27] Yik yak. https://www.yikyak.com/. Accessed: 2016-03-16.

[28] Louise Barkhuus, Barry Brown, Marek Bell, Scott Sherwood, Malcolm Hall, and Matthew Chalmers. From awareness to repartee: Sharing location within social groups. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 497–506, New York, NY, USA, 2008. ACM.

[29] Joseph B Bayer, Nicole B Ellison, Sarita Y Schoenebeck, and Emily B Falk. Sharing the small moments: ephemeral social interaction on snapchat. *Information, Communication & Society*, pages 1–22, 2015.

[30] Richard Chalfen. *Snapshot versions of life*. Bowling Green State University Popular Press Bowling Green, OH, 1987.

[31] Jennifer Charteris, Sue Gregory, and Yvonne Masters. Snapchat selfies: The case of disappearing data. *eds.) Hegarty, B., McDonald, j., & Loke, S.. K., Rhetoric and Reality: Critical perspectives on educational technology*, pages 389–393, 2014.

[32] James Clawson, Amy Voida, Nirmal Patel, and Kent Lyons. Mobiphos: a collocated-synchronous mobile photo sharing application. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 187–195. ACM, 2008.

[33] Fredrik Espinoza, Per Persson, Anna Sandin, Hanna Nyström, Elenor Cacciatore, and Markus Bylund. Geonotes: Social and navigational aspects of location-based information systems. In *Ubicomp 2001: Ubiquitous Computing*, pages 2–17. Springer, 2001.

[34] Google Developer Console Help. Content ratings for apps and games. http://goo.gl/2mDoHU. Accessed: 2016-03-16.

[35] Ryan Hoover. Whats the deal with snapchat? http://goo.gl/kE8FlL, December 2012. Accessed: 2016-03-16.

[36] Ryan Lawler. The secrets to snapchats success: Connectivity, easy media creation, and ephemerality. http://goo.gl/Oolxf6, 2014. Accessed: 2016-03-17.

[37] Andrew D. Miller and W. Keith Edwards. Give and take: A study of consumer photo-sharing culture and practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 347–356, New York, NY, USA, 2007. ACM.

[38] C. Ye O. Nov. Why do people share photos online? antecedents of photos' quality and quantity. 2009. AMCIS 2009 Proceedings.

[39] MartinWilbourn Partners. Snapchat is now the third most popular social network among millennials. http://goo.gl/VAf9nB, August 2014. Accessed: 2016-03-16.

[40] Christine Roberts. iphone app that deletes photos, snapchat, said to encourage sexting among teens. http://goo.gl/GjTqmb, November 2012. Accessed: 2016-03-16.

[41] Franziska Roesner, Brian T Gill, and Tadayoshi Kohno. Sex, lies, or kittens? investigating the use of snapchats self-destructing messages. In *Financial cryptography and data security*, pages 64–76. Springer, 2014.

[42] Mattias Rost, Henriette Cramer, and Lars Erik Holmquist. Mobile exploration of geotagged photographs. *Personal and Ubiquitous Computing*, 16(6):665–676, 2012.

[43] Chris Speed. Developing a sense of place with locative media: An underview effect. *Leonardo*, 43(2):169–174, 2010.

[44] Karen P. Tang, Jialiu Lin, Jason I. Hong, Daniel P. Siewiorek, and Norman Sadeh. Rethinking location sharing: Exploring the implications of social-driven vs. purpose-driven location sharing. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, UbiComp '10, pages 85–94, New York, NY, USA, 2010. ACM.