Developing a Location-Based Service to Reduce Smartphone Notification Distractions

Submitted APRIL 2016 in partial fulfilment of the conditions of the award of the degree of BSc (Hons) Computer Science.

Harry Mumford-Turner
psyhm1

With Supervision from Dr. Martin Flintham

School of Computer Science
University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated in the text:

Signature ________________________________

Date ______/______/______

Abstract

This dissertation focuses on the construction of an Android app to reduce the distraction of smartphone notifications. Although the app appeals to a wide cross-section of potential users, for the purpose of this project the focus is on University students during their study period. An observational study combined with research into similar systems identified the problem of smartphone notification distractions and determined how a system might solve this problem. Several findings from the research were identified. The types of distractions from smartphone notifications change in different environments. People are more likely to respond to a personal notification than one that is not direct and impersonal. Thus, users of this system want to see some notifications and block others when in different areas. An Android app was constructed to block impersonal notifications in different study areas, defined as zones. Each zone has a set of preferences that detail what app notifications to block and what notifications to deliver to the user. These preferences are specific apps to block, e.g. *WhatsApp* and a list of keywords, e.g. the user's family name. If a keyword is found in a notification, it will be delivered to the user's device, even if the app is on the blocked list. Zone data about each user of the app is sent to a central server, where useful crowd-sourced statistics are generated and displayed on their device in addition to a website. These statistics include the most popular blocked app and the most popular location to study. An evaluation of the project through a usability study revealed an overall positive assessment of the software with all participants confirming the usefulness of the project and several others stating they are less distracted.

Contents

1 Introduction

People spend a significant amount of time on smartphones and for most of us, they have become an essential part of our lives. *Ofcom*'s 2015 twelfth annual Communications Market report [1] states that *'The UK is now a smartphone society'* with *'young people ... ten times as likely as older people to say their mobile phone is the device they would miss the most'*. But when does smartphone use become a problem? If it distracts us from other activities, such as work, our performance for those activities is likely to decrease.

Smartphone usage is shown to be more popular with younger people, than the older generation. A 2014 *SalesForce* report [2] tracked 470 smartphone users, revealing the average person in education aged between 18 to 24 spent 5.2 hours a day on their smartphone. The more time students spend on their smartphone, the more likely it is to impact their study performance because of the distractions smartphones present.

Smartphone usage in classrooms was analysed [3] with students using them for course-relevant actions and another group using smartphones for course-irrelevant actions. The results revealed that *'sending messages unrelated to class content negatively impacted learning and note-taking, while related messages did not appear to have a significant negative impact'*. This proves that students are affected by smartphone distractions while studying, however, not every smartphone interaction is distracting if it is relevant to the current task at hand.

Although when smartphones are completely removed, effects are mostly positive. A study by the London School of Economics [4] found schools that banned phones had student test scores increase by more than 6%. The improvement is likely related to the attentional cost associated with receiving a smartphone notification. A study in 2015 [5] found that *'receiving and reading a notification might be a short affair, but can prompt task-irrelevant activities and promote mind wandering'*. If a notification is task-relevant, it would not distract the user and make the notification meaningful. Blocking task-irrelevant notifications is the key to improving task focus.

An unrealistic solution to these issues would be to remove the smartphone completely while working, but this is an impractical solution to lose a key part of people's lives. [1] A more realistic solution to this problem would be to use a system to track smartphone usage altering the behaviour so a user can toggle distracting apps on or off throughout the day. However, this method can be cumbersome and requires a significant amount of effort from the user, putting them off blocking smartphone apps.

Other systems save configurable lists of apps to block notifications from, enabling the user to toggle between these different lists. However, these are not tied to a specific area and block all notifications from the specificed app. Blocking all notifications from an apps is not practical for everyday use because the types of apps users want to block varies too often in different locations.

The aim of this project is to tackle the above problem by blocking different types of smartphone notifications by using keywords to identify which notifications are important to them. This will reduce the number of bad distractions and still let the user receive notifications important to them, even if they are from the same app. Lists of apps to block and keywords to let through can be easily changed and configured to accommodate different types of distractions changing in different environments.

The project consists of three components, an Android app, a web-service API and a website, that when combined solve the above problem, but can each be used individually for different purposes. The creation of these components was influenced by existing systems, literature, and an observational experiment.

2 Research

The following research was undertaken to help define the system specification and understand the design space of the project. Existing systems were first investigated to discover how other competitors approached the problem of notification distractions and key influential solutions are discussed. User's privacy is approached with care, discussing the ethical issues of monitoring smartphone activity. The disadvantages of smartphone notifications, methods of blocking the distractions and the correlation between high smartphone usage and unproductive study are discussed. An observational study is undertaken to understand the design space of the project. The study highlighted that unimportant, impersonal notifications are very distracting, and the type of distraction changes when smartphone usage changes to a different location or work environment.

The following research concludes that a system this project aims to build does not currently exist and will be a project that is valuable and highly useful.

2.1 Existing Systems

Existing systems with similar aims to this project were researched to better understand how other competitors tried to solve the problem of notification distractions. The following systems impacted the design of the project, their key components influencing the creation of a unique system that solves the problem. Using a combination of their ideas, such as notification blocking profiles, and new approaches like filtering keywords in notifications.

Different types of systems on various devices were looked at to find suitable software that could be used to prevent distractions. Systems that had smartphone apps proved to be better because they were easily available to configure, suitable for the user to interact with and easier to implement. Location-Based Services[1], such as *IFTTT* [6] use geofencing to trigger various actions, such as muting the smartphone when the device enters a specified location, but no apps were found that further enhances this to change types of notifications blocked.

The Android smartphone operating system has extensive API availability [7] that enables users to build apps to block smartphone notifications. Other operating systems such as *iOS* and *Windows Phone* have limited functionality for tracking notifications. Some solutions for *iOS* have been created such as Moment [8], that tracks how much time is spent on a selection of smartphones, but these solutions rely on the app to never be closed and is impractical for most users. Thus, several Android apps are discussed over iOS to focus on blocking notifications.

The Android app *Notifiy Block* [9] is one of many available for Android that creates profiles for different sets of app notifications to block. Although statistics such as the number of notifications blocked are recorded, the profile blocks all notifications from the selected app and gets rid of the notification information once blocked, meaning the user cannot see the notification again after it has been blocked. Switching between configurable notification blocking preferences is quick and easy, granting the ability for users to create a 'work' profile to block notification distractions while working. However, although easy to use *Notify Block* is also simple in use. Important notification information gets lost when blocked, so arguably a better solution could be to turn off all noise that occurs when receiving notifications, so distractions are removed and notification information can be seen afterwards.

Another Android app called *Rescue Time* [10], monitors smartphone app usage by tracking the amount of time a user spends on their smartphone and labelling each app used as *productive* or *unproductive* time spent. Instead of directly preventing notification distractions, *Rescue Time* encourages the user to reduce distractions by displaying information after a working session, and

[1] A service that is dependant on location.

compares your working day with your previous days and another user's day. This report can show users how they can improve, motivating them to work at a productive level that is higher than other people. Displaying statistics to users about how they work and showing information about how other users work is a useful service that encourages users to choose certain factors, because they know others have selected them and been successful. *RescueTime*'s crowd-sourcing aspect heavily influenced the construction of this project and inspired the need for popular user statistics.

The Android OS has some functionality to block notifications and in later versions of Android [11] the *do-not-disturb* function gets quite complex. Users have the ability to select calls or messages from specific people as priority interruptions that they want to receive to their smartphone, while blocking others during certain configurable hours. Apps can be manually added to this priority list or set to always to have their notifications blocked. The settings for this feature can be accessed from the notification bar by every Android user (from version *Lollipop* onwards) making it easily accessible to everyone. In addition to this, because this feature is a part of stock Android, the user experience has been heavily refined and tweaked, so it's very easy to use. Although there are lots of configurable settings, it doesn't take into account types of notifications from the same app (except for calls or messages from specific people), or location-based notification blocking settings.

2.2 Privacy

There are several privacy issues that arise with this project, they are discussed below with strategies on how to mitigate the problems.

Possible Privacy Issues

The system parses the notification content to detect if the text content of a notification contains a keyword. Although not always, a notification could contain sensitive information that users may not be willing to give up easily. Users might not even be aware that this information is sensitive, or even care that it is revealed to the system.

The keywords shared to the central server are written by users and have the possibility to contain personal information, such as the name or surname of the user, and if there aren't large amounts of data sources for the system, it is likely that this information could be used to identify an individual user. Posner [12] approves this reasoning, when discussing intelligence information gathering. *'The volume of data collected cannot invade privacy because a machine reads most of the information'.* However, in this case, the information gathered could be low in volume around areas of reduced use of the system, e.g. small town, resulting in some undesirable sensitive keywords about a few users, e.g. name of the user. The *Nothing to hide* argument [13] can be used whereby people might rationally assume the confidential information would not be seen by human eyes and only the analysed patterns revealed.

Users might not be aware of the amount of information they are giving the program access to. Any notification received during a defined study period is read and parsed by the system. The user is letting the system read their current notifications in addition to future information from new notifications that they might not be entirely aware the system will read. Is it morally right for a computer system to read every notification message that the user might receive? The *Nothing to hide* argument states *'if an individual is only undertaking legal activities, they have nothing to fear'* [13] and that it is morally right for computer systems to read personal messages. Daniel J. Solove reviews this argument discussing the *Nothing to hide* argument is better defined as *'The right to conceal discreditable facts about themselves'*. Although this project is reading concealed information about an individual, it is returning a positive experience back to them.

Therefore, yes. there is a certain value to this private information, but without handing over the notification information, the system would not function to the best of its ability.

Solutions to These Privacy Issues

Daniel's review of the *Nothing to hide* argument discusses *'the value of privacy the argument provides is low because the info is often not particularly sensitive'*, [13] and in this case, most notification information is not that private, so the cost of giving up this information is also low.

Smartphone device permissions can be confusing to some users. To mitigate the sensitive information revealed if these permissions are used incorrectly, as many researchers in this field agree, [14] data minimization should be used to collect the least amount of data to help ensure the privacy of a user and still achieve the same goal.

Users need to be aware that their keywords can be shared and be given the option to stop this action. *'People do not expect their televisions to listen to every word in the house. They want transparency and risk communication'*. [14] Therefore, the system aims to store little data about a user, only the notification text analysed by the system and not sent to the central server. The app is transparent in sending information to the central server to respect people and respect their privacy.

Possible Cyber-Behaviour Issues

Users of the app send the location of zones they have created to the central server to be displayed on the website map. This information has the potential to track a user if they studied at different locations throughout the day. Other users of the app could have an ulterior motive, tracking users movement across multiple study locations and aggregating other services with this location data to find out more information about a user. Respecting People and Respecting Privacy [14] describes this motive as *'Secondary Use'* of the geo-located data. Although the users have to give consent for their location to be sent to the central server, they could be unaware that this information could be used to track their movement when moving to a different study area. Pairing a user location along with other services, e.g. Twitter, and aggregating data about a specific location could reveal more information about a user who is studying. [15]

Daniel writes, *'the combination of little bits about an individual, reveals something they want to hide* [13], which may cause harm to a user of the system. Another article states a penalty in revealing information, *'while the user enjoys the Location-Based Service (LBS), they pay the penalty of disclosing their private data to these LBS services'*. [14] That penalty should not result in stalker-like behaviour from other users. However, perhaps because this Cyber-Behaviour does not involve face to face social interaction, users would care less about the feelings of others and their sense of identity could be transformed.

Solutions to These Cyber-Behaviour Issues

The location of a user is private information that should be made anonymous before it is shared with others. There is a moral obligation with the storage of users' locations. Developers must follow *fair principles* when handling a user's location, following abstract privacy rules. *'Intensive tracking might betray a pathological desire for data collection, but it might also be a result of the easiest default to set'*. [16]

The ease of collecting large amounts of data makes it easier to increase the quality of the product because of the vast pool of information available. However, studies have shown that the *'quality of service generally decreases when increasing location privacy'*. [17] The quality of the product is determined by the users. They will reduce their trust in the product if they believe it is misleading them and if they have privacy concerns about their location. The *Nothing to hide* argument discusses harm with the aggregation of data, *'for the person who truly has nothing to hide, aggregation is not much of a problem'*. [13] Therefore the point is valid with this project and shows aggregation is not a privacy issue in this case, because places of study would usually be crowded locations where aggregated data from multiple sources would be unlikely to cause harm.

This projects system stores only the necessary data about a user, with location privacy as a top priority because this information is not displayed to all users of the system in an identifiable manner.

2.3 Smartphone Distractions

Little research has been conducted into combating the distractions smartphone notifications present, but the following literature influenced the project by highlighting issues that could occur and give an insight into the market.

The following research shows that turning off a smartphone and blocking all interaction with it, prevents the distraction it creates. But it is not a realistic solution as they are a necessary part of everyone's life. We can see that people are easily distracted and find it tough to ignore phone notifications so they are a particular problem during periods of study. People's attention span is not very high and can be broken by task-irrelevant notifications that heavily distract the user. Although some phone usage is beneficial, the environment is a key factor that decides what a distraction is. People do not mind a system knowing their location and monitoring their smartphone activity if they receive a suitable reward. However, this location information can be used against them if it is not dealt with in a careful way.

Smartphone Impact on Learning

Blocking smartphone usage in a working environment has been shown to improve students study ability. A study that analysed the impact of smartphone usage on student learning was conducted that blocked smartphones from a class lecture, [18] comparing the results with smartphone usage allowed. Students wrote more concise notes and were able to recall more information from the lecture when smartphones were blocked. The results suggests that blocking smartphone usage improves students workability to focus on the work at hand, give their full attention to a task and therefore increase their learning capability.

Smartphone Usage

A 2014 annual report that tracked 470 smartphone users [2] found that (between people aged 18 to 25) 5.2 hours are spent on a smartphone in a day. In addition to this, *'85% of people say a smartphone is a key part of their life'*. Are people willing to give up a *'key part of their life'* to reduce distractions, if they are, how long will that last if they are spending over 5 hours a day using this device? Removing the distractions smartphones present by turning off their phones is not feasible as it is a vital part of their life. The system needs to target the real problem, of continual notifications from the same app where not all are of high significance to them.

Cost of a Smartphone Notification

There is an attentional cost associated with receiving a smartphone notification. A study in 2015 [5] found that *'receiving and reading a notification might be a short affair, but can prompt task-irrelevant activities and promote mind wandering'*. People could be getting distracted by their smartphone but not realising that notifications are the problem. All types of notifications may not be distracting, if a notification is task-relevant they would not distract the user and become very meaningful. Therefore, blocking task-irrelevant notifications is the key to improving task focus.

Length of Peoples Working Span

An educational research report [19] states that *'managing our attention, requires impulse control'*. When a smartphone user receives a notification they have a strong impulse to view the information about that notification and see what it is. The thrill of finding out new information about our lives is something innate to humans, similar to how mechanical slot machines work. Smartphone interaction keeps people wanting more and places them into a *'machine zone'* satisfying a slot machine like sensation. [20]

The educational report also discusses a study on secondary school children where *'students averaged less than six minutes on a task before succumbing to other digital distractions'*. If the length of a student's attention span is already low, receiving a notification during those six minutes would reduce the time even further! The report does not highlight many methods to combat these solutions, other than to completely remove the distractions by turning off notifications or taking regular breaks. Using techniques such as, segmenting time into short work and break time into blocks, similar to the Pomodoro technique [21] that aims to *'manage distractions and eliminate burnout'*.

Types of Smartphone Distractions

Not all smartphone use is seen as a distraction. Some classrooms ban smartphone use completely, but others actively encourage students to take advantage of the features on their smartphones, such as the calculator or other educational apps. An interview with a teacher [22] highlights how they let students listen to music from their smartphone device while they work. *'The noise level in the classroom goes down, and the work amount goes up when you let them listen to their music'*. This shows that the context of work matters highly when labelling smartphone activity as a distraction.

Environment of Smartphone Activity

Smartphones are seen as distracting devices in one environment and useful devices in another. A study was undertaken [23] to understand how students work in study spaces, such as the library, and found that *'the context of learning takes precedence'* with the type of work to decide if a smartphone is distracting. *'Students using a smartphone in a social study location would promote relaxation and social bonding [distracting actions], but in the Library it would often aid with studying - [Students] wanted quiet areas to get away from others so they could concentrate better'*. Although the type of location students chose to study changed based on the context of their work, the type of notifications received wouldn't necessarily change and be distracting (if not more distracting) in quiet areas. What people perceive as a distracting event during study, changed based on their environment and location of work.

Location-Based Services

A system to track a user and check what environment they were in would need to keep checking their current location for comparison. A Location-Based Service (LBS) provides a service that is dependent on location. But there are ethical considerations for this location data. A user of the service may not mind their current location being tracked if the reward is great. Or if the service removes a problem, such as the distractions of smartphone notifications, they more likely to hand over more location information. People do not imagine their location-based data being abused by applications. But if a company builds up data about an individual, they can combine their resources and use it against them. A BBC programme [24] discusses that *'data can be analysed and people can be found based on their location actions, e.g. find me people who like to go to coffee shops'*. This can lead to targeted advertising and less anonymity, where companies exploit this big data and build up a portfolio about an individual. Therefore, location-based big data has benefits by providing Location-Based Services, but care needs to be taken to properly protect the users privacy.

Privacy Preservation

A smartphone will regularly send a user's location to a LBS (Location-Based Service) while the user is using it, but what precision should be used and how often should the location be updated? Users want *'to use a Location-Based Service without revealing private location information'*, [25]. There is a *Service-Privacy* trade-off between a LBS and the user. If a user wants to experience 100% of the features a LBS offers, a user must sometimes provide fine grain location information to that service.

There are many highly technical solutions to better preserve a user's location [26] but basic techniques can still increase the anonymity of a user, such as using an appropriate level of accuracy when asking for the location of the smartphone. If the service needs real-time location updates to function, is the user aware that this is happening, do they know the implications of this, and what implications will this have on the smartphone battery? *'While it's unavoidable that the smartphone company already knows the rough location of the customer it's another matter if the customer's precise location can be tracked over time through pattern[s] of location-dependent queries to [a] remote database'*. [27] These factors should be taken into consideration when creating a Location-Based Service as it is the users right to know this information and they are more likely to trust the service if it is honest about its actions. Strategies recommended by Android [28] suggest requesting location information at a coarse level of granularity and reducing the interval for requests to preserve a user's privacy.

Smartphone Monitoring

Monitoring smartphone activity needs to be handled carefully as it may contain sensitive information about a user. Although a study on user' privacy [29] found that *'people are less concerned about their location being tracked, as long as they find the service useful'*. People may be willing to trade their location information for a useful system, but what is the implication of monitoring notification information that contains potentially sensitive information? The study goes on to state that systems should look at *'what level of privacy is actually needed and desired by users'*. Are people happy with being monitored by a system that parse every notification received? Another study [30] reviews a company that monitored employees and found that *'employees have strong feelings of disliking monitoring, as they perceive privacy violations and unfairness of the practice. Disclosure of policies does little to alleviate the lack of support for monitoring'*. The study revealed that although people are aware they are being monitored and are aware of the rewards they serve to gain (employee praise). They disregard that reward and show negative feelings towards the practice. Perhaps because the reward is not high enough or the relationship between the reward and monitoring is not worth the negatives of monitoring.

2.4 Observational Study

There are a lack of systems available to block smartphone notifications. None of these systems block notifications that are irrelevant and impersonal to users. Therefore, an observational experiment was conducted to find qualitative data to understand the design project space, identify the user requirements and to investigate how distracted users felt after receiving a notification during a working period. See Appendix 9.2 for an example information sheet and consent form.

2.4.1 Overview

Five participants were separately observed during an hour study period. The observer monitored their smartphone interaction. Afterwards, participants discussed their activity and how a system could reduce any distractions smartphone notifications presented.

The observer went with the participant to a location where the participant had previously studied and felt happy to work. The participant placed their smartphone in a place where they would normally position it while working and whenever their smartphone made a noise or vibrated, the time was noted by the observer. If a participant interacted with their smartphone, the start and stop time of interaction were also noted. After an hour of observation, the participant stopped working and was questioned by the observer about their activity, using the noted times of interaction to guide the interview. The questions were asked to guide the conversation to why they got distracted, how they felt their work went and if this was a representable working period.

2.4.2 Key Results

The small sample size of the study meant qualitative results proved more worthwhile as an evaluation and to help define the user requirements. The key results found are listed below and were useful to the project, with people adopting different tactics to reduce distractions. Some results were surprising, with several participants thinking they received a notification when they, in fact, did not. A single participant had no smartphone usage at all and other less surprising facts showed participants reacting differently to some notifications but not others.

All Participants Read Every Notification They Received, but Only Responded to Those Personal to Them

Three of the participants would normally have their phone out of their pocket face up during a normal study period, so phone notifications could be seen and heard straight away. The rest felt the notification vibration in their pocket. A participant set their phone on the desk, so they could respond instantly to a direct message. They would analyse the notification on their lock-screen and see if it was a personal message e.g. an SMS message. If it wasn't, e.g. *'a random Facebook notification'* they would dismiss it and carry on working. This meant any notification could be seen and distract from their study session, even if the notification was not a direct message and they just dismiss it straight away.

Several Participants Left Notifications on Their Phone Lock-Screen as a Reminder to Act on Them Later

Participants reacted differently to different types of notifications, with all participants receiving notifications that did not require an immediate response. Four participants would read the notification and leave it on their lock-screen so they could come back to it after they had finished their study period. A participant instantly dismissed a calendar reminder notification, but said that it was useful information to know at that time, but they did not want it to stay on their lock-screen and did not need that information after their study. With another type of notification, they swiped down to read the full message and based on the content decided to respond to it. If users had a way of ensuring they could see notifications that did not require an immediate response after their study period, instead of when they arrive, it would not distract them when the notification appears on their phone.

A Participant Got Distracted by Lots of Notifications from the Same App

A participant did not want to interrupt the flow of work when receiving notifications. They only looked down and checked their phone when they got a few notifications in quick succession, they would wait until they were at a natural pause before checking. They described a notification limit that had to be reached before they checked their phone. The more notifications they got from a single app in a short time implied a greater importance to read those notifications.

2.4.3 Design Influence

Four results that directly influence the requirements of the project are detailed below.

Block Different Types of Notifications

Participants did not want to interrupt their flow of work when new notifications arrived on their phone. Stopping notifications during a study period is a solution to this interruption, but all participants described some notifications that they wanted to block and others they did not. Therefore, the design should be able to block different types of notifications.

Do Not Block Personal Notifications

Some notifications were personal to participants with some of these coming from the same apps as impersonal notifications. Participants analysed the notification on their lock-screen when they received it to decide what category it fell under. Most of these personal messages had a similar structure throughout different apps, for example the participants name would be in the notification content. It would be useful to set a list of keywords, that if mentioned in the notification then it does not get blocked. There may also be need for more complex grammar to identify if a message is personal, e.g. message ends with a question mark.

Display Blocked Notifications after Study

Participants left some notifications on their lock-screen to serve as a reminder so they could look at them after a study period. If a system blocks notifications from appearing on the lock-screen during a study period, that system should return the blocked notifications back to the lock-screen, so no information about a notification is lost.

Notification Blocking Preferences per Location

All participants had multiple places to study every week. Each area was a different study environment. A participant discussed that in the library they would not touch their phone, setting it to *do-not-disturb*. But in a common room, it would be on the table and they would interact with the incoming notifications. It would be useful for users to have a different set of notification blocking preferences for each location they studied in. The project could encourage productivity by displaying detailed statistics about the user's study period for each location, so they can identify what study area is best for them.

3 System Specification

The goal of the project is to reduce the distractions of smartphone notifications, so users they can be more productive during a work period without completely removing the smartphone device.

The following narrative explains how a user of this project reaches the goal and how they reduce their distractions while using the proposed system. In addition to this, a more concrete set of requirements is listed to state what a user would need from the system in order to solve the goal. The requirements serve a purpose of evaluating the software by comparing them with the finished system.
Constructing the specification from the research conducted will mean if the design follows the specification then the system should meet the project goal.

3.1 Written Narrative

A written narrative was constructed (see Appendix 9.1) to help realise how the project would work from a users perspective. It tells a story about a regular user of the project and highlights their interactions with the system that reduces the distractions from their smartphone notifications.

3.2 User Requirements

The following requirements list what users need from the system for the project to solve the goal.

Encourage User Productivity

The research conducted shows that users do not care about impersonal notifications and get distracted by other apps.

1. The project should block notifications from apps the user has specified.
2. The project should still send notifications that include specified *don't block* keywords.

Display User Statistics

The research shows us that users want to leave some types of notifications for after their study period.

3. The project should display statistics about the study period, including the blocked notifications and study session details.

Map User Notification Preferences for Each Location

From the research, users discussed different study locations and different phone settings set for each location.

4. The project should display each location where the user had studied and list the notification and *don't block* keywords for each one.

Consider Privacy

Each zone a user creates will be their location of study and part of the information will be sent to a central server then displayed on other devices. The project should take care to ensure data transmitted to the central server is anonymous, so a particular user cannot be identified to a specific location.

5. The project should ensure user data sent to the central server is anonymous.

4 Design

4.1 Overview

Constructing this project to meet the specification, required a design step before the implementation process. The design stage ensured the solution was feasible because a trivial example of all features was implemented in the form of a prototype. The system design was refined and visualised as early as possible using simple sketches to iterate quickly through design changes and improve the user experience.

The system contains three components that work together to achieve the system specification. An Android app, a Web-Service API and a Website all combine to reduce the distractions of smartphone notifications.

The app creates locations known as zones to block different types of notifications and let through specific notifications containing keywords. The server contains the web-service API and the website. The API creates useful insights from all users of the system and pushes these to the app and website. The API interacts with a database containing information about all users of the system, that is used to store new data from a user and return information about all users. Although not a design requirement, the website displays useful statistics to everyone, such as the most popular blocked app in a particular location and is seen as a design opportunity.

4.2 Prototype

A prototype was constructed to test the feasibility of the solution. A technical solution was built to prove the user requirements can be implemented into an Android app.

The prototype can monitor the Android smartphone activity, read new smartphone notifications, giving the option to block specific notifications so they do not appear on the user's lock-screen and can re-send those blocked notifications at a later time.

The core implementation builds on top of the features developed for the prototype, meaning any issues and shortcomings were discovered early on which sped up the implementation process.

4.2.1 Feasibility

A large amount of exploratory work was needed to find suitable solutions for each requirement and locate the best route to achieve them.

Several interesting factors were considered in this process. The most surprising were the amount of data that could be gathered when targeting an Android device, compared to another platform. Although there were some limitations to this, with later versions of the Android API [11] providing different mechanisms to track smartphone activity. [31]

Permissions needed to be granted by the user in the settings menu to start monitoring notifications. If a user was not prompted to accept app permissions, monitoring issues arose which influenced a need to re-prompt users if this permission was ever turned off, or not accepted at app launch. For example, to monitor smartphone activity an *Accessibility Service* [32] was used. A particular type of component that requires different permissions that users see less often and so could be more distrustful.

4.2.2 Components

The main components of the prototype are the *NotificationListener* that listens for incoming notifications, blocking them if necessary, a location tracker that polls for a user's location detecting if it has changed. Finally, a user interface that displays the notifications that were blocked and enables interaction with the prototype.

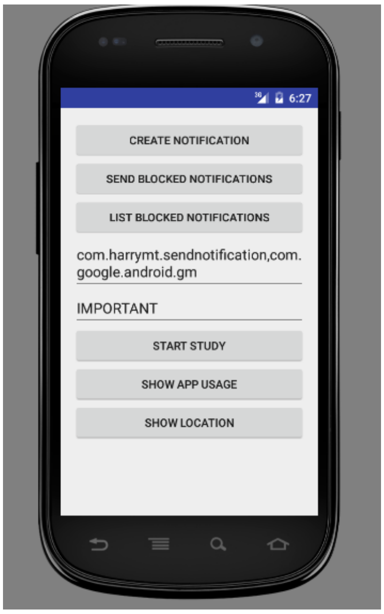

Figure 1: Screenshot of the prototype

Prototype Detects Notifications

An Android *NotificationListenerService* [7] was needed to listen to all notifications the user receives, and once the user has acknowledged the permissions, the listener starts monitoring notifications posted.

Prototype Blocks Notifications Sent to the Phone

The *NotificationListenerService* cancels the notification immediately after it has been posted. The event that is called when a user receives a notification has a long enough delay in between itself and when a notification is displayed to the user lock-screen. The prototype prevents this alert from happening.

Prototype Blocks Notifications from Specific Apps

The *NotificationListenerService* compares an incoming notification with a list of apps the user wants to block. The full name of each apps is used, referred to as a *package* in Android [33]. If the package name matches the incoming notification package, the notification is cancelled, and the notification object is saved to a database ready to be constructed and displayed at another time.

Prototype Lets Through Notifications Containing Keywords

The *NotificationListenerService* receives information about a notification and searches through the text elements of the notification comparing it with a list of keywords and blocking the notification if it finds a match.

Prototype Displays a List of Blocked Notifications

Information about a notification is stored in a database. After the period of blocking notifications, information such as the package name of the app that issued the notification, can be read and displayed to the user.

Prototype Pushes Blocked Notifications Back to the User

The chosen mechanism saves characteristics of a notification such as the title of the notification, then creates the notification later and pushes it to the user's lock-screen. Although certain notification characteristics were unable to be recreated, such as the image icon of the notification. These had to be removed due to security constraints, as the prototype cannot read the image path that points to another apps directory.

Prototype Detects a Change of Location

Using the *Google Location API* [34], the prototype finds the most appropriate location by checking various sensors to get the most accurate location possible. These are preferred over the *Android location API* as encouraged by Android. [35]

Prototype Prepares Data Ready to Send to the Central Server

An object gets saved after the study period containing a unique ID for that the user with information about the study period such as the study location and notification preferences.

4.3 App Design

The app design decisions were heavily influenced by the research found and the construction of the prototype. Because a basic implementation of all features could be achieved in the prototype, there were few limitations in design.

The design of the app was approached with care and to spend less time programming, more time was spent refining the designs. A well thought through design helped remove the bad aspects of each feature, understand the value proposition that each feature held and reduced the implementation time, so the deadlines were met.

Brief sketches were drawn to illustrate how a user would interact with each feature of the app. Each component of the prototype was selected, working out how it would best fit into the app using the written narrative (see Appendix 9.1) as a guide. These final sketches were used during the construction of the app, replicating the designs in program user interface code. Android widgets were also reviewed and revised before the design stage, so the design of the app could match Androids recommended native style layout.

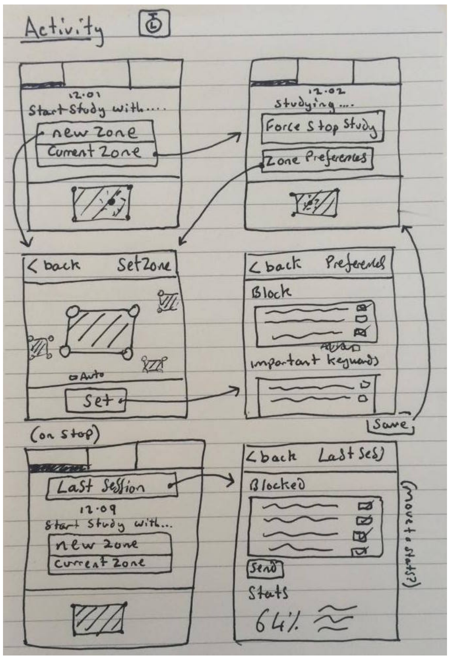

Figure 2: Example of how the user flow through the app was better understood using quick sketches

Proposed Design

To well-structure the app, the system specification was sorted and separated into different sections, each representing a part of the app.

- Managing the starting and stopping of notification monitoring.
- Listing all zones.
- Displaying statistics.
- Creating new zones and editing existing zones.
- Setting blocking preferences.
- Viewing blocked notifications.

The parts were defined into logically separate actions as Activities, per Android's style guidelines [36] with the most common components in easy reach for the user. A single Activity, labelled *Main Activity* was used with Android's Swiped Tab Views [37] as an effective navigation for managing notification monitoring, editing zones and displaying statistics. The rest were separated into individual Activities.

4.3.1 Activities

- Edit Zone Activity: Create or Edit Zones
- Edit Zone Preferences Activity: Set or Edit Zone Preferences
- View Last Session Activity: View information about the last study session
- Main Activity: Manage notification monitoring, edit zones and display statistics.

Edit Zone Activity

Early sketch designs described how zones were manipulated. The wireframe below is the finished design for this Activity.

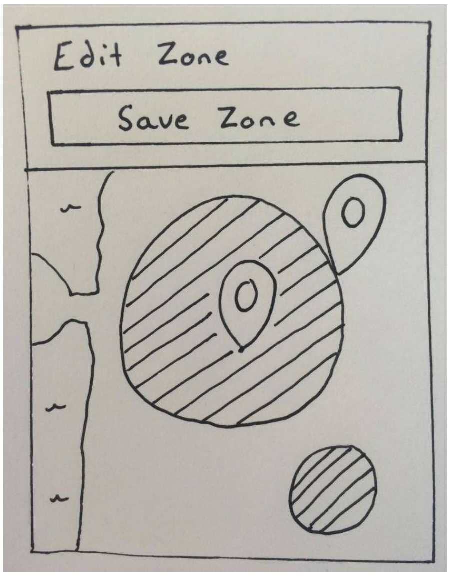

Figure 3: Wireframe sketch design for the Edit Zone Activity

The Edit Zone Activity is set out to make it easy for the user to create a zone or edit an existing one. The map is easy to navigate, and existing zones are drawn onto it, that highlight the name of each zone to make each one more recognisable.

A circle was the most appropriate shape to use as a geo-fence because it requires fewer amounts of taps to change its size, compared with a rectangle. Also, it is suitable for the size of location users require for a geo-fence that provides the best user experience.

The *Set button* confirms the created zone shape and launches the *Edit Zone Preferences Activity* where the user can set the notifications they want to block for this zone. Setting the preferences for the zone appeared to be the next logical step after creating a zone but did not seem right to place before a zone location had been chosen.

Edit Zone Preferences Activity

Different sketch designs experimented with different layouts to show keywords and apps blocked. The wireframe below is the finished design for this Activity.

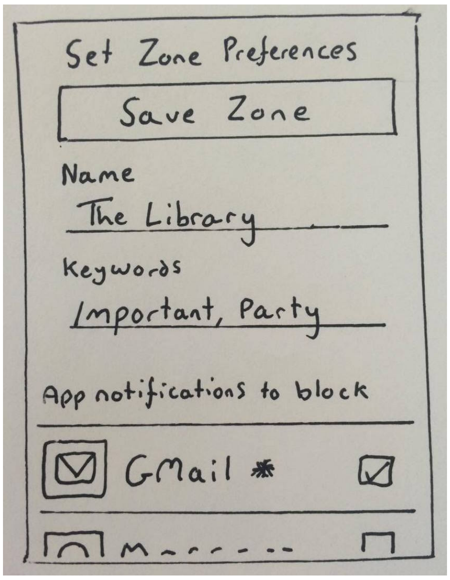

Figure 4: Wireframe sketch design for the Edit Zone Preferences Activity

To make for a good user experience, every app on a user's smartphone are shown in an easy to select list with images and checkboxes, but some Android system apps are left out, such as the clock, because the user is unlikely to receive a notification from them. Popular app statistics are retrieved using the web-service API and are placed at the top of this list. These are marked with a star icon to denote a popular app that they should be more inclined to select as per the recommended method of choosing icons. [38] Displaying popular apps that others have used during the selection of apps to block was the most appropriate time and would be most useful to the user.

The name of the zone is a user-friendly method to identify each one, and is asked for at this stage, rather than during zone location selection as it seemed more sensible to ask for information in groups.

Keywords are displayed in a single input box rather than a list view, because a user may not have many keywords to add, and also may be overwhelmed with lots of lists.

To move to the next step the user presses the set button and returns to the *Main Activity* where the new zone can be seen on the map.

View Last Session Activity

To get a clear design that lists the blocked notifications to the user, different designs were tried and experimented with, before the below finished wireframe design for this Activity.

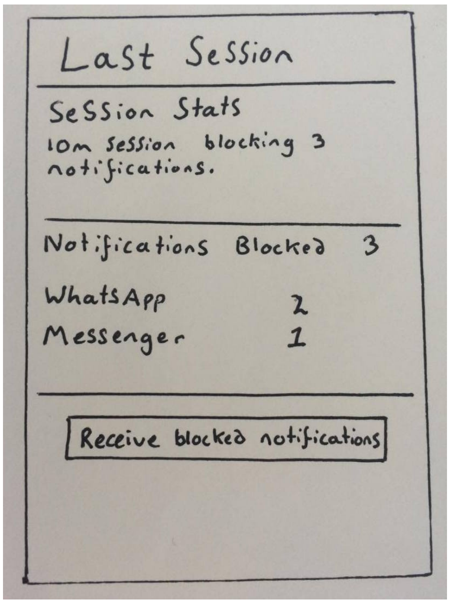

Figure 5: Wireframe sketch design for the Last Session Activity

Statistics are shown to the user about the last time they blocked notifications. Keywords and apps blocked are listed to make the user aware of what notifications were blocked.

Notifications are listed in a grouped list so the user can see exactly how many notifications they want to send back. The button to post notifications back to the lock-screen is labelled and needs to be explicitly pressed so a user can send the notifications once, or choose not to send them at all.

Main Activity

Below lists the main design overview for each section of the Main Activity, Track, Zones, and Stats.

Track

- Links to the Activity to Create New Zones.
- Starts and stops a study session when in a zone.
- Links to the Activity to display a users last study session.
- Shows a map to display nearby zones to the users current location.

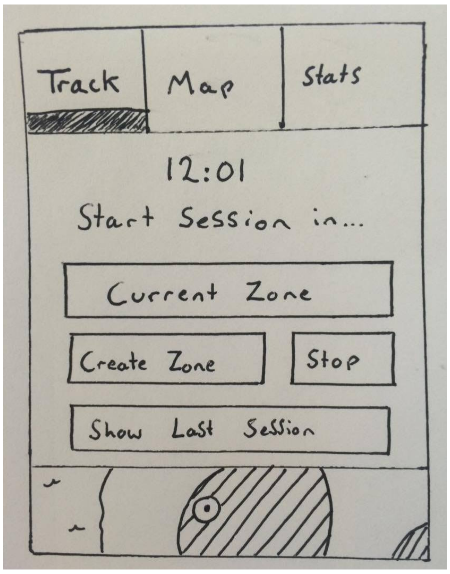

Figure 6: Wireframe sketch design for the Track section in the Main Activity

This section of the app is seen first by every user, so needs to be clear about what the app as a whole aims to achieve. The other tabbed sections are less important, so they are not displayed first, with the other parts of the app are linked within them.

The *create zone* button links to the *Edit Zone Activity* to create a new zone and will be the first button a user should press to start a session. To help guide the user, copy text is added to make this button more attractive to press first.

Start and stop buttons are added so the user can be in full control of notification monitoring, explicitly labelling the start button as *'start in zone'* to make it more attractive to click for users. The *last session* button is only enabled if a user has previously started a session to encourage the user to press it after they have finished studying.

A map is a way for users to instantly see if they are inside a zone by highlighting their current location and showing the name and location of previously created zones. As soon as the user opens the app, they will be able to see how far away they are to a zone and if they have created one at this location.

Zones

- Lists each zone a user has created.
- Links to *Edit Zone Activity* for each zone in the list.
- Has the ability to delete a zone from the list.
- Displays a summary of keywords and apps set for each zone.

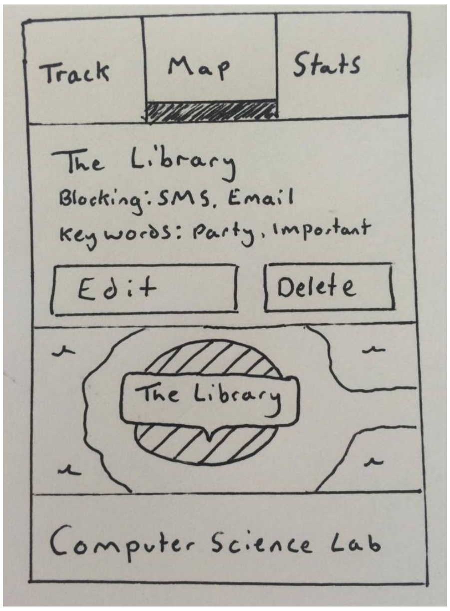

Figure 7: Wireframe sketch design for the Zones section (previously named map) in the Main Activity

A list displaying every zone was preferred over a big map of all zones because comparing the location of the zone with other zones seemed of little importance. A list of static map images combined with information about each zone proved more worthwhile because users want to see what information the zone has, at a glance, before they choose to edit it. The name of the zone is labelled on the static map to make the map on the Track section consistent with this.

Buttons to edit and delete the zone were added next to each zone, enlarging the edit button because users are more likely to press it more often. The *edit* button opens the *Edit Zone Activity* to edit that zone and the delete button requires a confirmation box before the zone is deleted, so accidental deletions are out of the question.

Stat

- Displays statistics about the user.
- Displays popular blocked apps and keywords from all users using the web-service API.
- Has the ability to post this device's data about apps and keywords chosen to the server.

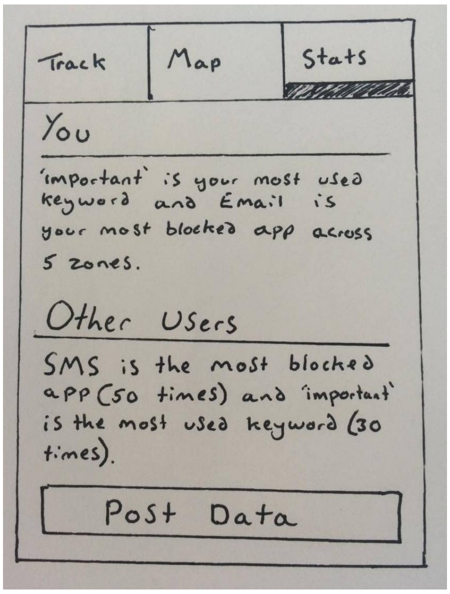

Figure 8: Wireframe sketch design for the Stat section in the Main Activity

Users need to see statistics about their overall usage, so these are displayed, along with their most popular apps blocked and keywords used. Users want to compare their usage with others, so the crowd-sourced data is requested from the server and displayed here. So the most popular apps and keywords from all users are displayed.

The post data button is added explicitly, rather than automatically posting data, so users are fully aware they are sending their data to the remote database.

4.4 Web-Service Design

A central server with a database to hold data from all users was required to create crowd-sourced statistics from every user's data.

The process of pushing data from a device into this database needed to be easy and lightweight because the information was being sent from a smartphone that has reduced Internet connectivity. Android guidelines recommend [39] sending the least amount of data as little as possible, so the decision about what data to send and receive was important to achieve this.

The system specification states that users want a list of the most popular apps and most used keywords, so the functionality to get this data was needed. To provide useful insights about certain locations, latitude, and longitude along with a radius will be stated to request zones, popular apps or keywords in that area.

An API is needed to send and receive data to the website and the app. The chosen set of calls to interact with the server are minimal but provide enough information to give insights about users, sending the data to the app and website in addition to saving new zones from the app.

- Request the top number of keywords and apps in a location.
- Request the closest number of zones to a location.
- Send zone data.

4.4.1 Website Design

The website displays the most popular place to study for a particular location. Users can view the zones around this location to find what are the most popular apps blocked and keywords used. Users want to be able to read useful insights and statistics, so the most blocked apps and keywords used across all users are listed below.

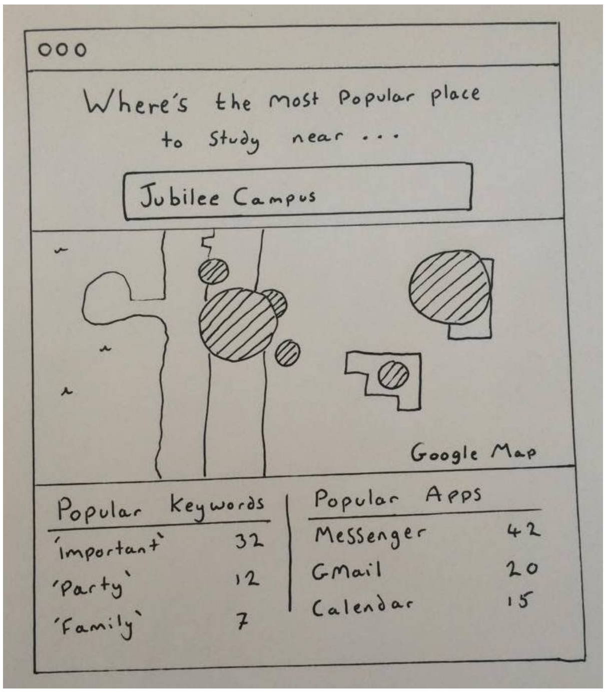

Figure 9: Wireframe sketch of the website

The website compliments the Android app by showing a large map with every zone in that area so the user can view the most popular zones for a certain location to find the most popular place users of the app study.

5 Implementation

The designs were used to construct the three parts of the system, the Android app, the web-service API and the website.

5.1 Android App

The Android app handles the blocking of notifications from specific apps and can let through notifications containing specific keywords. Locations can be defined in the app as zones that block notifications from a pre-defined list of apps and let through notifications that include keywords when a user is inside a zone. After a period of study, a user can receive any notifications that were blocked while inside the zone, so no information is lost. Useful insights about other users of the system, such as the most blocked app, are sent to the remote database using the web-service API to create crowd-sourced statistics. These insights are pushed back to all users of the app and are displayed to everyone via the website.

5.1.1 Development

Development of the Android app used Android Studio [40], Android's official Integrated Development Environment (IDE). The app was built using Gradle [41] the recommend build system for Android apps. GitHub [42] is used as version control to store and manage changes to all program code. The app was deployed and continually tested throughout development using a Nexus 5 Android phone.

A series of 3rd party libraries assisted the development of the app. Google's official Map and Location API libraries [34] along with Google's Map Utility library [43] provided easy location and mapping functionality to the app.

To assist with API requests to the central server, a HTTP library called Volley [44] was added following Android's recommend guidelines to make *'networking for Android apps easier and most importantly, faster'*. [45]

An API key was needed to make requests to Google API's which needed to be kept private on a per developer basis. All the code in version control is publicly facing, following good practice, the API key file is kept out of version control, and other developers need to add this file to make the project work. A key for the web-service API is also kept secret from version control to prevent unauthorised calls to the API.

5.1.2 Permissions

Android handles permissions in a similar style to other smartphone operating systems. This means similar issues arise when asking a user for these permissions. When the app is first opened after the install, the permissions settings dialogue is displayed, asking the user if they want to grant permissions to monitor notifications and track the device location. Edge cases were identified during this phase and programmed for. For example, if a user revoked permissions during a study session or chose not to grant the permission on start-up, parts of the app would fail.

5.1.3 Architecture

An Android app is typically split up into several components. The graphical user interface is split up into lots of single focused tasks a user can do, called *Activities*. Each Activity contains lots of input and output interfaces called *Views* that provide user interaction, such as a button. An Activity can also contain a series of *Fragments* that behave like nested Activities within an Activity. These *Fragments* are used to create a multi-pane user interface and are used in creating the *Tabbed View* navigation in this app.

Another fundamental Android component is a Service that performs long running tasks usually in the background. This app uses a service that runs in the background to listen for incoming notifications to the device.

Monitoring Notifications

Listening to notifications requires the *NotificationListener* component that builds on top of the core Android *NotificationListenerService* [7] class, extending it's functionality by overriding the event that occurs when a notification is posted to the lock-screen.
When a notification is sent to the device, this event is fired and the component checks to see if it should block the notification before it is sent to the lock-screen. It checks if the app package name is on the list of blocked apps and parses the notification content for any keywords. If it finds a match, a cancel request is issued, and the notification content is saved to the database via the *DatabaseAdaptor* component.

The *NotificationListener* component makes it easy to monitor any incoming notifications and cancel them before they appear on the lock-screen.

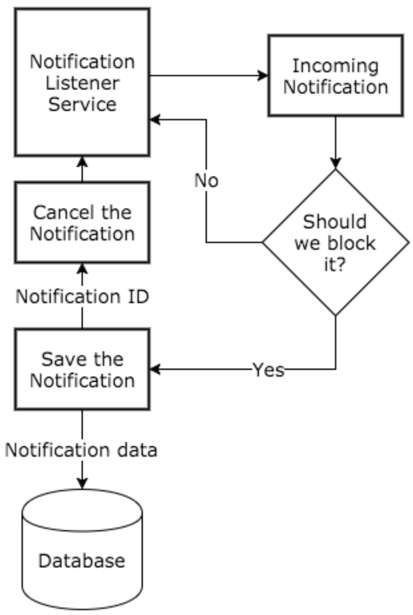

Figure 10: Flow chart illustrating how the NotificationListener component works

Polling for Location Updates

To detect if a user is inside of a zone, the app must continually poll for the current location of the device to check if it has changed since the last request. The component *LocationPoller* uses the Google Location API to easily start requesting location updates. This is the recommended method supplied by Android [28], because the Google location APIs are preferred over the Android framework location APIs. [35] An event is fired when a user's location changes, the co-ordinates are requested and compared with the user's previous location via the *DatabaseAdapter*. This means the app has a record of the users last known location to use with other components, and can detect if a user is inside of a zone by help from the *DatabaseAdapter*.

Database

A Database is used to store information about a user's zone, their current session, and to generate statistics about the user.

SQLite [46] is the relational database management system in Android and is used with a *DatabaseHelper* class to read and write data in the app. The database layout is defined separately in a *DatabaseSchema* class that define the database table names and SQL create and delete tables string constants. The database on the device contains information about a session that does not need to be sent to the central server. So there are differences between the structure of the database on the app and the database on the central server.

The *DatabaseAdapter* class contains various database methods to manipulate data, such as editing, deleting and saving a new zone. But some of the database methods simply read all data from a particular table, such as the zone, keywords and apps tables. The session table keeps a track of information about the current session, stores the data from the notifications that have been blocked and tracks the start time of the session. Other methods are used as simple number statistics, for the total number of unique apps being blocked, the keywords used across all zones and the total number of zones created.

To handle the transfer of data between the server database and the device database, only certain information is selected to be transferred. Each new zone a user creates is flagged as not having been sent to the server. When the button is pressed to post the new zone data, each zone is marked as sent in the app database on a successful transmission.

To detect if a location is inside of a zone, for when the device location changes, the distance between the centre point of the zone and the current location point needs to be compared using the radius of the zone. The Haversine formula is used to calculate the distance between two points on a sphere. However, a modified version is used [47] to take the shape of the earth into account. The simplified formula is as follows:

$$c = 2 \times atan^2(\sqrt{a}, \sqrt{(1-a)})$$
$$distance = R \times c$$

Where R is earths radius (of 6,371km), c is the angular distance in radians between two coordinates, and a is the square of half the chord length between the points. This formula is used on the server and the app to calculate a rough distance between two coordinates.

5.1.4 User Interface

The user interface was constructed in Android Studio, defining the elements of each Activity in XML files. Some of the layout was tough to implement and issues arose that required slight tweaks to replicate the design.

The *Main Activity* required a series of nested components called *Fragments* [48]. To create the Tabbed View each Fragment had a different layout that represented each page of the Tabbed View and required extra work to handle switching between each tab. A small amount of padded space was added to the edge of each fragment to stop the user from accidentally scrolling the Google Map on the Track Fragment rather than switching to different tabs with left or right swipes.

Users interaction when creating a zone in *Zone Edit Activity* was tricky to implement. Integration with the Google Maps API was easy, but there were some limitations involved. During development it was tricky to drag a marker around because the size of the marker was tiny.

Google Maps forced custom markers to be added to the project because their API did not allow the re-size of their default markers.

When the app is opened, statistics are requested from the server using the web-service API and are stored in memory because they are used in multiple places around the app. Edge cases were identified during development and detect if the request fails, if it is invalid or if the device is not connected to the Internet. The app can be used without the Internet, but suitable messages are displayed to the user that encourages them to connect to the Internet to get more app features, such as statistics.

5.1.5 User Flow

The user requirements for the project are listed below paired with each implemented feature of the app, to show an example user flow throughout the app.

Encourage User Productivity

The project should block notifications from apps the user has specified.
The project should still send notifications that include specified *don't block* keywords.

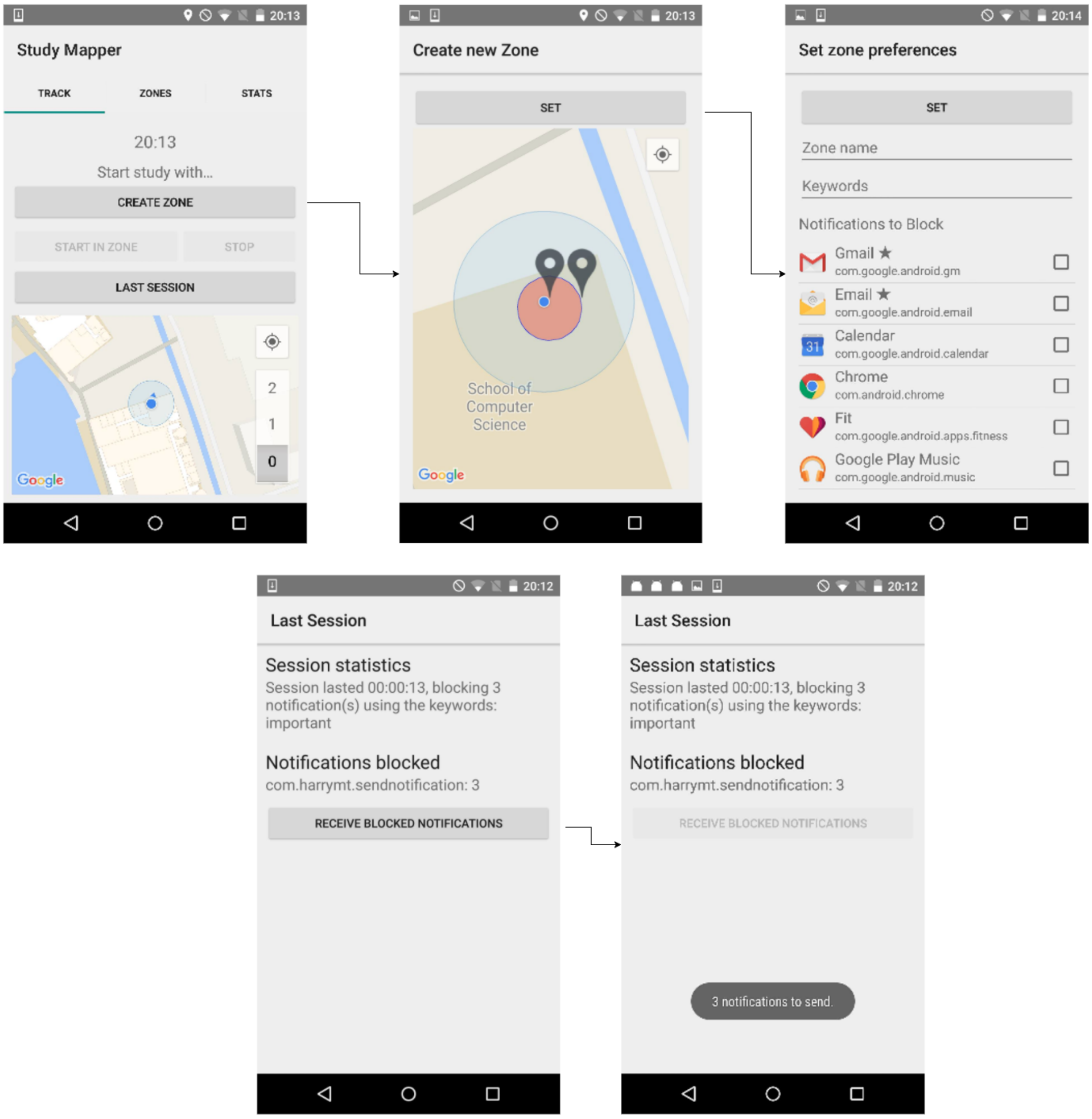

Figure 11: User flow to show how the app can block specific notifications, let through ones with keywords and send the blocked notifications back to the user

Display User Statistics

The project should display statistics about the study period, including the blocked notifications and study session details.

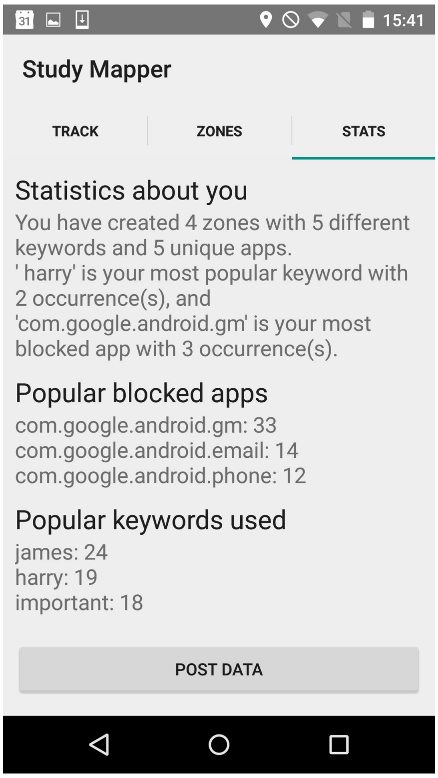

Figure 12: Screenshot from the app, showing the MainActivity Stats Fragment

Map user notification preferences for each location

The project should display each location where the user had studied and list the notification and *don't block* keywords for each one.

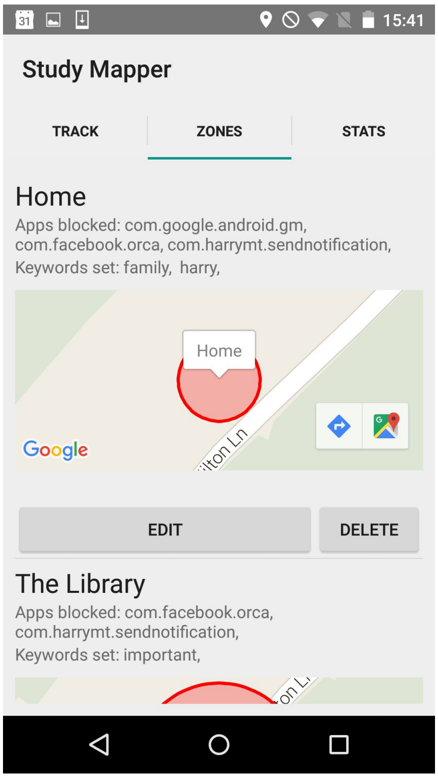

Figure 13: Screenshot from the app, showing the MainActivity Zones Fragment

Consider Privacy

The project should ensure user data sent to the central server is anonymous.

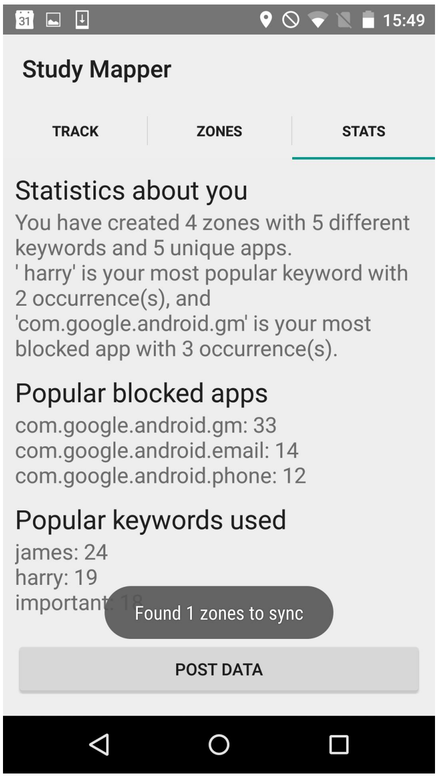

Figure 14: Screenshot from the app, showing the MainActivity Stats Fragment sending data to the server

5.2 Web-Service

The web-service API on the server provides a method of collecting data from all apps to generate useful statistics that can be pushed to each device and displayed on the website. The API was built in *PhP* using PhPStorm [49] as the IDE, again using GitHub for controlling versions and automatically deploying the code to the server along with the website.

5.2.1 API

An API is needed to provide this data to the website and the app. The REpresentational State Transfer (REST) concept is chosen to send and retrieve data from the database over the other popular Simple Object Access Protocol (SOAP) protocol because the REST style has less overhead and is more standardised as it uses HTTP protocol.

The construction of the REST API is separated into different requests. Retrieving data is denoted by the HTTP GET protocol and sending data uses the HTTP POST protocol.

A simple status check was implemented to aid with debugging and development that simply returns a user-friendly status message and optionally the server-side time.

- GET /status/
- GET /status/date/
- GET /status/time/

The following endpoints get the most popular keywords, apps and zones of all time, and optionally, targeted at a specific location. A number is added to help with development so a function knows exactly how many results will be returned. The location also contains a search radius to give the ability to search in a specific zone.

- GET /keywords/number/
- GET /keywords/number/lat/lng/radius/
- GET /apps/number/
- GET /apps/number/lat/lng/radius/
- GET /zones/number/lat/lng/radius/

The POST protocol is used to send zone data using the body of the API response (payload). The payload carries the zone data in JSON format.

- POST /zone/

Only basic zone data about apps and keywords are useful to send to the server. A user identification number is passed along with this data to anonymously link each zone with a particular device. Below is an example payload to use with the API request *POST /zone/*.


```
{
    "user_id": 1,
    "id": 2,
    "name": "The Library",
    "lat": 51.2,
    "lng": 1.19,
    "radius": 23.0,
    "blockingApps": ["com.facebook.messenger", "com.google.gm"],
    "keywords": ["important"]
}
```


The API creates useful insights, stores information about all users of the app and creates statistics that are pushed back to each app and the website. The Haversine formula is also used when detecting if a location with a radius overlaps the radius of another zone. The formula previously listed is replicated in *PhP*, the language of the web-service API.
Extending the capability of the API is easy because it has been built using standardised protocols and secure techniques. A key needs to be passed in with each API call to authorise each request, preventing read and write access from unauthorised users.

The below diagram shows the direction of data between the web-service API, the app and the website.

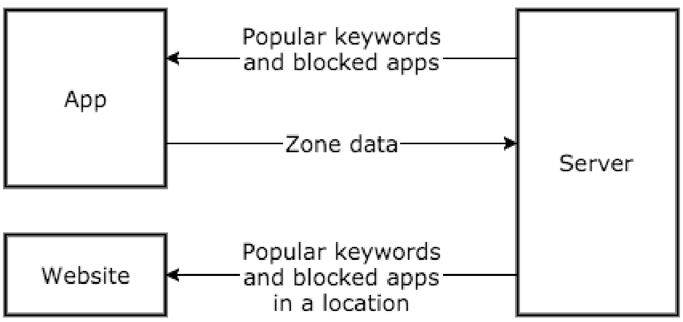

Figure 15: The direction of information between the web-service API and the other components

5.3 Website

The website shows the most used keyword, most blocked app and displays interesting insights to everyone. Users can view the website to find the most popular place to study and see information about what apps are blocked at that popular location.

The website is written using a combination of *PhP*, *Javscript*, *HTML* and *CSS*. The program code is post-processed before it is deployed to the central server. A task runner named *Grunt* [50], runs a series of commands to add additional components to parts of the language and provides several optimisations, such as compression in the form of *minification* [2].

A CSS pre-processor called *Sassy CSS* (SASS) [51], acts as an extension to CSS, adding extra functionality such as variable names in CSS. When Grunt processes SASS to compile it into CSS, an additional step is added that adds support for all modern browsers using a lightweight SASS library called *Bourbon.* [52]

Security is handled in a similar way to the app, when using the API keys (Google Maps Javascript API and the central server API key), by not storing the API keys in version control, but separately in different files.

[2] The process of removing all unnecessary characters from source code without changing its functionality.

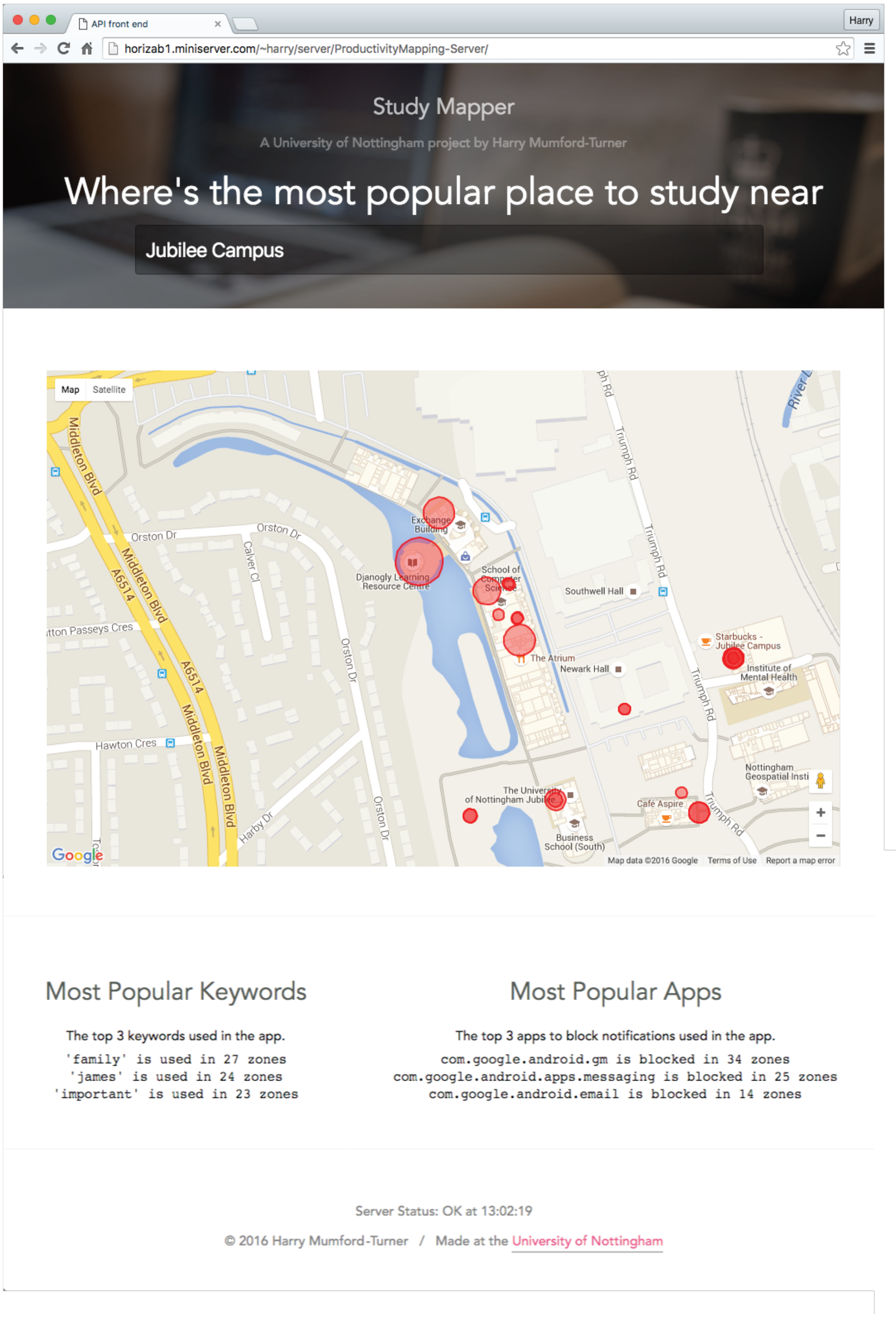

Figure 16: Screenshot of the website

The website makes a request to the web-service API to get all the zones for the particular area specified. The data is serialised into a format that the Google Maps API can use to drawn the zones onto the map. The most popular keywords and blocked apps are also listed in a simple table to provide users with insights about the app.

5.4 Project Management

A series of tools were used to assist with aspects of project management including version control, progress tracking and issue management.

Git was chosen as the version control system, to store the program code for disaster recovery and create a link with *GitHub* to handle issue tracking. These tools were found to be the most powerful and user-friendly to keep a track of changes and revert to later versions if problems with the current version arose.

A Gantt chart was used to meet deadlines, plan the progress throughout the project and ensure a high level of software quality. [53] The chart helped manage the planning of the implementation of the project due to the significant amount of software engineering required.

5.5 Deliverables

Below are the list of deliverable items constructed, that make up this project.

1. Android App
2. Web-Service API
3. Website

5.6 Summary

The diagram shows the interaction between the three components and briefly how data passes between them.

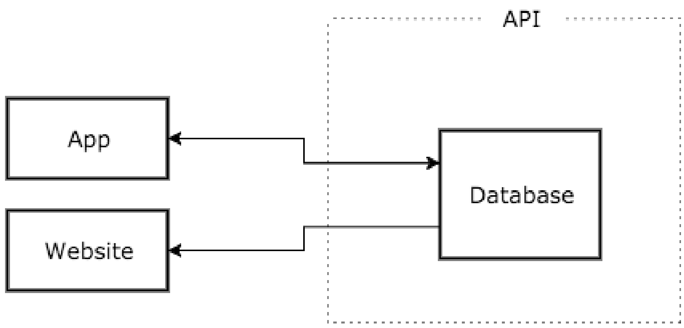

Figure 17: Diagram briefly showing how information passes between the three components

6 Evaluation

An evaluation of the whole system was needed to identify the quality and prove the success of the project. Evaluating the usability of the software is a useful method to measure software quality [54] and to ensure the system has all the functionality users required. Usability testing was conducted in the form of a user study. Four users discussed their experience when using the app and talked through any issues they encountered. A personal evaluation reviewed the system specification against the project, using the usability results from the study to decide if the project meets the requirements.

6.1 Usability Study

The evaluation of the software involved four participants using the app, hearing their experience with using it, and discussing any improvements or issues they found.

6.1.1 Overview

The usability study aimed to evaluate the software by listening to users experience with the app, identifying software amendments and comparing their results with the user requirements. See Appendix 9.3 for an example information sheet and consent form.

The app was built and loaded onto four participants Android smartphones. Each participant was encouraged to use the app during a working period as much as they liked, for up to seven days. After each participant had used the app during a study period, a short interview followed to discuss their interaction with the app, how they felt while using the app and to see if they were less distracted while studying.

6.1.2 Key Results

Similar to the observational study, the sample size was small, meaning qualitative results were focused upon, and the interview was a discussion about the project more than a quantitative survey. The study was guided by the system specification to touch upon all aspects of the requirements.

The user requirements are listed below paired with each result from the interviews to display the link to the discussion and each requirement.

Encourage User Productivity

1. The project should block notifications from apps the user has specified.
2. The project should still send notifications that include specified *don't block* keywords.

All participants stated that the app blocked notifications during their study period. They confirmed these were notifications from apps they specified in the blocking list. Notifications that contained keywords participants had added were sent to the participants smartphone, even if the app was on the blocked list. One participant stated that it was unclear exactly how many keywords they should add and if adding more would benefit them. The lack of clarity could be the result of the size of the input used to enter keywords, increasing the size of the textbox could increase the amount of keywords people added. The same participant wanted to allow notifications from a particular person and added that persons name as a keyword. However, they suggested that letting through individual contacts as key people, would be easier to select and more preferred.

Three participants stated that the app was useful for *'blocking out unwanted messages in group chats'*. However, another participant said they preferred the mute setting in each messaging app as a way of reducing distractions more than this app.

Display User Statistics

3. The project should display statistics about the study period, including the blocked notifications and study session details.

All participants acknowledged the statistics about their study session and commented on the usefulness of grouping the apps. However, two participants stated they were confused about what the apps listed were, and one participant wanted to *'show the list of apps from the preferences screen'*. Perhaps this was because a user was familiar with the look of the blocked apps list, and would be able to better understand the apps that had been blocked, by the visual similarity between the two lists using the name and icon.

Three participants were surprised about the small amount of statistics about their study session. One participant suggested the app displayed the number of notifications they received vs. the number blocked. These statistics are not necessarily directly reducing the distractions of notifications, but are interesting to the user and would provide a more enjoyable experience while using the app that perhaps would encourage the user to spend more time using it.

Map User Notification Preferences for Each Location

4. The project should display each location where the user had studied and list the notification and *don't block* keywords for each one.

One participant commented about how they wanted to see a full map of all the zones they had studied and the number of blocked notifications at each place. However, three participants stated they liked the simple list view of zones, but all participants stated they did not feel satisfied with how the blocked apps and keywords were displayed. Perhaps users want a single standardised interface for displaying a blocked app throughout the design and prefer the layout of the blocked apps list when setting preferences.

Consider Privacy

5. The project should ensure user data sent to the central server is anonymous.

One participant commented about the information sent to the central server via the web-service. The participant questioned, *'does the app need to send the location of each zone to the server?'*. Could the same experience be achieved without sending the location of the zones to the server? If a user creates a zone in a remote location, some people could perhaps infer where that user of this app lives. Some steps could be taken to prevent this, such as, not passing in a user's exact zone location, but instead using a coordinate that could be inside the radius of the zone to preserve privacy.

6.2 Critical Appraisal

The project found that smartphone notifications do cause distractions and these distractions can be reduced using zones and keywords. The project presents a partially successful solution to the problem of smartphone distractions. The system acts as a more advanced do-not-disturb feature tied to different locations. The usability study results indicate mostly positive feedback, highlighting specific notification blocking with keywords as the strongest feature of the app.

The quality of the software was difficult to measure because the number of usability testers was small so a mean failure rate of functionality is impossible to identify, and the small sample size does not guarantee the evaluations were not biased. The results from the study could not be representative of a standard users study period. Repetition of the experiment would make the results more meaningful increasing their reliability. The collected data may not be valid because the participants were all University students from a similar age group. Older people may have different types of notifications and still may be studying or wanting to reduce smartphone notification distractions. The impact the results have on the app development are valid for the age range of users interviewed in the usability study, but may not necessarily accurate or useful for all users of the app. The same applies to the observational study, as that included participants from the same age range. Also, participants knew they were observed as the study was overt. Perhaps if a covert study was conducted, the data gathered would be more representative which may confirm the validity of the original study or produce different results.

Unexpected notification metaphors were discussed during the Usability study. A participant found the app was unable to block notifications from Facebook Messenger's *floating heads*. The solution they found was to turn this feature off in the Facebook Messenger app, but the user was disappointed that these notifications were not blocked. Arguably this type of widget is not the same as a notification. However, it is still a smartphone distraction and will still distract the user, unless they manually remove this feature from the Facebook Messenger app.

The decision to choose Android over Apple's iOS affected how the prototype was built and the development of the entire system. Several development paths were analysed before the prototype was constructed, such as Cordova [55] a system that uses web technologies to create cross-platform apps to speed up construction. Exploratory mechanisms were tried using this method for monitoring app usage and listening to notifications. But during this stage, it was realised that there was little need for Cordova, because most of the app logic needed to be Android or iOS API specific code. Moving the prototype from Cordova to Android resulted in less code required to perform the same action. This process highlighted the value of small, experimental research, such as a prototype, before a development route is chosen, because it saves a significant amount of software development time later. Choosing Android did have some drawbacks, with inconsistent documentation about various Android API's used to monitor app usage and listen to notifications, perhaps due to the unpopular use of these features. The inconsistent documentation led to some unexplained results and slower development time because more research was required to understand the situation. However, if Apple's iOS with its more restricted API was used, the app would have even less functionality than the current system in Android and would not meet the user requirements.

Three key additions to this project were suggested by several participants during the usability study to improve the success of the system.

Firstly, displaying statistics about smartphone usage while in a zone during a study period, such as app usage, could be beneficial to users because they can be warned of how many minutes they spent on their phone. This list of apps could be split into two lists labelled as productive apps and unproductive. *Rescue Time* [10] could be used to look up each app and define what label is for what app. A user could set warnings that send a notification alert if they spend a certain amount of time on an app they have labelled unproductive. These alerts would break the lengthy distraction and help users not get distracted. After the study period, users could reflect on their time spent on distracting apps and identify issues with their behaviour while studying, perhaps removing the distracting apps from their smartphone. This feature was achieved during the prototype stage by extending an Android *AccessibilityService* [56] to detect an event that happens when the user changes their focus from one app to another.

Secondly, advanced personal notification detection could be used to decide if a notification should be blocked. For example, a message with repeated question marks ('harry???') could be seen as more personal, regardless of if the phrase ('harry') is set as a keyword to let through.

Finally, individual contacts across multiple apps could be linked such as SMS and *WhatsApp* messages, so notifications across multiple apps from a particular person could be sent to the phone or blocked. Combining multiple streams for each contact could be useful to select family members or close friends that a user might be expecting a message from, but are unsure what messaging platform they would use.

From the results of the usability study and using comparisons against the system specification, the choice of Android for this project was correct to achieve the user requirements and create a successful system to reduce the distractions of smartphone notifications.

7 Conclusion

The development of the project followed the plan to a great extent, setting realistic milestones for features to finish the project in full. The project shows that smartphone notification distractions were an issue. However, this problem can be reduced by using the Android app. The observational study revealed that there are different types of notifications a user wants to block and sometimes these are from the same app. The Android app was constructed to reduce these distractions by blocking notifications from different apps and letting through specific keywords. A web-service API provides a fascinating insight into what apps users want to block and what keywords in notifications they want to receive. The website displays these statistics generated from all users of the app to show the most popular location where people have created zones to study. The evaluation identified three key improvements that would form the basis of further work. To display statistics about smartphone usage, add more sophisticated personal notification detection and link contacts across multiple apps. Implementing these tasks would enhance the project, improving the solution. However, overall the current Location-Based Service proved useful and in part, successful.

8 References

[1] Ofcom. The Communications Market Report. Technical report, The Office of Communications, 2015.

[2] Salesforce. Salesforce Mobile Behavior Report. Technical report, Salesforce, 2014.

[3] Munz S. & Titsworth S. Kuznekoff, J. H. Mobile Phones in the Classroom: Examining the Effects of Texting, Twitter, and Message Content on Student Learning, Communication Education. *Communication Education*, 64:344–365, 2015.

[4] R. Beland, L. P. & Murphy. Ill Communication: Technology, Distraction & Student Performance. Centre for Economic Performance. *CEP Discussion Paper No 1350*, 2015.

[5] Mitchum A. & Yehnert C. Stothart, C. The Attentional Cost of Receiving a Cell Phone Notification. *Journal of Experimental Psychology: Human Perception and Performance*, 2015.

[6] IFTTT. Inc. If this then that. Android App. `https://play.google.com/store/apps/details?id=com.ifttt.ifttt`.

[7] Android. Notificationlistenerservice. `https://developer.android.com/reference/android/service/notification/NotificationListenerService.html`.

[8] K. Holesh. Moment - track how much you and your family use your phone. iOS App. `https://itunes.apple.com/us/app/moment-track-how-much-you/id771541926`.

[9] Aranea. Notify block. Android App. `https://play.google.com/store/apps/details?id=com.araneaapps.android.apps.notificationdisable`.

[10] RescueTime Team. Rescuetime time management; Android App. `https://play.google.com/store/apps/details?id=com.rescuetime.android`.

[11] Android. Android lollipop. `https://developer.android.com/about/versions/lollipop.html`.

[12] S. Kaplow, L. & Shavell. Economics analysis of law. *Handbook of Public Economics, Volume 3*, 2002.

[13] D. J. Solove. 'I've Got Nothing to Hide' and Other Misunderstandings of Privacy. *GWU Law School Public Law Research Paper No. 289*, pages 1, 11, 18, 2007.

[14] L. J. Camp. Respecting People and Respecting Privacy. *Communications of the ACM, Vol. 58 No. 7*, pages 1–2, 2015.

[15] European Network and Information Security Agency. Privacy considerations of online behavioural tracking. Technical report, Enisa Report, 2012.

[16] J. Cooper, A. & Morris. Binding Privacy to Location on the web. In *Proceedings of the Second International Workshop on Location and the Web*, 2009.

[17] V. Iyer, K. B. P. & Shanthi. Study on Privacy Aware Location-Based Service. *Journal of Scientific & Industrial Research*, 72:5, 2013.

[18] S. Kuznekoff, J. H. & Titsworth. The Impact of Mobile Phone Usage on Student Learning. *Communication Education. Vol. 62*, pages 235–240, 2013.

[19] K. Goodwin. Managing Attention Span in the Digital Age. Technical report, The Association of Independent Schools of New South Wales Ltd, 2015.

[20] N. D. Schll. *Addiction by Design: Machine Gambling in Las Vegas.* Princeton University Press, 2013.

[21] F. Cirillo. *The Pomodoro Technique.* (The Pomodoro), 2013.

[22] E. Graham. Using Smartphones in the Classroom. *Advice and Support*, 2013.

[23] D. Suarez. What students do when they study in the library. Using ethnographic methods to observe student behavior. *Electronic Journal of Academic and Special Librarianship*, 2007.

[24] Editor: Vadon R. Producer: Bowlby, C. The Dictatorship of Data, 2015.

[25] M. F. Mokbel. Privacy-Preserving Location Services, 2008.

[26] Jain M. & Padmanabhan V. N. Guha, S. Koi: A Location-Privacy Platform for Smartphone Apps. In *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 183–196, 2012.

[27] K. B. Atallah, M. & Frikken. Privacy-Preserving Location-Dependent Query Processing. *In Proceedings of the International Conference on Very Large Data Bases*, 2014.

[28] Android. Receiving location updates. https://developer.android.com/training/location/receive-location-updates.html.

[29] A. Barkhuus, L. & Dey. Location-Based Services for Mobile Telephony: a study of users' privacy concerns. *Proceedings of the INTERACT 2003, 9TH IFIP TC13 International Conference on Human-Computer Interaction*, 2003.

[30] S. Fountain, P. D. & Coultrup. Effects of Electronic Monitoring and Surveillance on the Psychological contract of employees. *Proceedings of ASBBS*, 2012.

[31] Android. Usagestats. http://developer.android.com/reference/android/app/usage/UsageStats.html.

[32] Android. Accessibilityservice. https://developer.android.com/reference/android/accessibilityservice/AccessibilityService.html.

[33] Android. Packages. https://developer.android.com/reference/packages.html.

[34] Android. Google location api. https://developers.google.com/android/reference/com/google/android/gms/location/package-summary.

[35] Android. Making your app location-aware. https://developer.android.com/training/location/index.html.

[36] Android. Activites. https://developer.android.com/guide/components/activities.html.

[37] Android. Implementing navigation. https://developer.android.com/training/implementing-navigation/index.html.

[38] A. Bedford. Usability Testing of Icons. *Nielsen Norman Group*, 2016.

[39] Android. Performing network operations. https://developer.android.com/training/basics/network-ops/index.html.

[40] Google. Android studio. https://developer.android.com/sdk/index.html.

[41] Gradle Inc. Gradle. http://gradle.org/.

[42] GitHub Inc. Github. https://github.com/.

[43] Android. Map utility library. https://github.com/googlemaps/android-maps-utils.

[44] Android. Transmitting network data using volley. https://developer.android.com/training/volley/index.html.

[45] Android. Transmitting Network Data Using Volley. *Android Training*, 2016.

[46] D. R. Hipp. Sqlite. https://www.sqlite.org/.

[47] Rosetta Code. Havesine formula - in java. http://rosettacode.org/wiki/Haversine_formula.

[48] Android. Fragments. https://developer.android.com/guide/components/fragments.html.

[49] JetBrains. Phpstorm. http://www.jetbrains.com/phpstorm/.

[50] GruntJS Team. Grunt: The javascript task runner. http://gruntjs.com/.

[51] Weizenbaum N & Eppstein C. Catlin, H. Sass: Syntactically awesome style sheets. http://sass-lang.com/guide.

[52] Thoughtbot Inc. Bourbon - a lightweight sass tool set. http://bourbon.io/.

[53] N. R. Tague. *Tagues The Quality Toolbox*. ASQ Quality Press, second edition, 2004.

[54] J. Nielsen. *Usability Engineering*. Academic Press Inc, 1994.

[55] The Apache Software Foundation. Cordova. https://cordova.apache.org/.

[56] Android. Accessibilityevent. http://developer.android.com/reference/android/view/accessibility/AccessibilityEvent.html.

9 Appendix

9.1 Written Narrative

1. Harry wants to study, so he opens the app to see where he has previously studied.
   (a) The app displays a list of locations where Harry has previously studied.
2. Harry walks to a usual study area the Library. He opens the app.
   (a) The app detects that he has entered a previously used study zone.
   (b) The app lists the types of (previously defined) notifications it will be blocking for this study session. i.e. *WhatsApp*.
   (c) The app lists (previously defined) keywords that if contained within a notification, will not be blocked and alert Harry's mobile phone, for this study session.
3. Harry is attending an event tonight and doesn't want to miss notifications about it.
   (a) Harry wants to add two keywords that describe the event tonight, *party* and *tonight*.
   (b) Harry edits the current zone (Library) and adds the two keywords to the list.
   (c) Harry saves the new keywords to the Library zone.
4. Harry starts his study session by telling the app he is studying.
   (a) Harry presses the *Start Study* button.
   (b) The app starts monitoring his notifications, checking through the list of apps to block notifications from and detecting if a keyword is in a notification.
   (c) Harry closes the app, locks his phone and starts working.
5. Harry's friend sends him a *WhatsApp* message that says *'hi'*.
   (a) The app blocks this notification from being sent to his lock screen.
   (b) The app saves this notification for after Harry's study session.
   (c) Harry is not distracted by this notification because his phone doesn't register that he has received one.
6. Harry's other friend sends him a *WhatsApp* message discussing the event later. *'What time are you going to the party tonight'*.
   (a) The app detects this keyword in the notification and doesn't block or save it.
   (b) The notification gets sent to Harry's mobile phone.
   (c) Harry responds to the message, then carries on studying.
7. After Harry's study session he leaves the Library and opens the app to check what notifications were blocked during the study session.
   (a) Harry ends his study session.
   (b) The app displays a list of notifications that were blocked and from what apps they were blocked from.
   (c) Harry sees one notification blocked from *WhatsApp*.
8. Harry receives the blocked notifications.
   (a) Harry presses a button to receive all blocked notifications.
   (b) The *WhatsApp* notification gets sent to his phone.
   (c) Harry replies to his friend.

9.2 Observational Study Forms

Below is an example information and consent form used in the observational study.

Observing smartphone notification interaction

This is an observational study to find out how many notifications you receive and how you interact with these notifications during a set study time. You will be observed for up to 1 hour in a *study* environment familiar to you. During the study period, if you receive a notification, the time and sound or vibration noise will be noted.

The observational period will take no longer than 1 hour. After the study period there will be a short interview about your experience that will be recorded (audio only).

The data collected will be used to analyse how often notifications are received and how people interact with different types of notifications. You may contact me at any time for information about the research or in relation to your consent, my address details are:

Harry Mumford-Turner
47 Teversal Ave
Nottingham,
NG7 1PY
psyhm1@nottingham.ac.uk

Participant ID: 4321

Your data will be stored in accordance with the Data Protection Act 1998, namely on a password protected drive in a secure facility and only for the duration for which it is required. It will only be accessible by those directly involved in the research.

You have been chosen to participate in this study as someone who might be representative of using this type of system. You may withdraw consent from the experiment at any time during or after the task for any reason without penalty by contacting me on the address above. In this event all data will be erased.

The voice recorded interview will be transcribed anonymously by using your participant ID to identify your transcription. The transcriptions will be included in the writeup of my dissertation.

Observing smartphone notification interaction - Consent Form

This is an observational study to find out how many notifications a typical smart phone user receives and how they interact with these notifications during a set study time. You will be observed for up to 1 hour in a *study* environment familiar to you. During the study period, if you receive a notification, the time, sound and vibration noise will be noted.

The observational period will take no longer than 1 hour. After the study period there will be a short interview about your experience that will be recorded (audio only).

Read and understand the attached information sheet, which includes information about the data to be recorded.

You understand that you can withdraw at any time either during the experiment or by contacting the researcher with the participant ID assigned to you, at the address provided in the information sheet, or by the email provided and your personal data will be erased from the records.

This is to confirm that you are over the age of 18 and have agreed to take part in a research study on the date: 20/02/16

Signed: J. Bloggs

Name: Joe Bloggs

Address: 1, Example Lane, City,

ABC 123

Contact Tel no: 07654321234

Email: jbloggs@example.com

Participant ID: 4321

In addition to the data analysis, you give permission for data that could be used to identify yourself (e.g. voice) to be transcribed and discussed in the writeup of my dissertation [✓]

Signed: J. Bloggs

9.3 Usability Study Forms

Below is an example information and consent form used in the usability study.

Usability testing an Android zone mapping app

This is a usability study to test the functionality and user experience of the Zone Mapping Android app.

You will install the Zone Mapping app on your Android device and use the app for up to 7 days.

After this period there will be a short interview about your experience that will be recorded (audio only).

The data collected will be used to improve the functionality and user experience of the app. You may contact me at any time for information about the research or in relation to your consent, my address details are:

Harry Mumford-Turner
47 Teversal Ave
Nottingham,
NG7 1PY
psyhm1@nottingham.ac.uk

Participant ID: 1234

Your data will be stored in accordance with the Data Protection Act 1998, namely on a password protected drive in a secure facility and only for the duration for which it is required. It will only be accessible by those directly involved in the research.

You have been chosen to participate in this study as someone who might be representative of using this type of system. You may withdraw consent from the experiment at any time during or after the task for any reason without penalty by contacting me on the address above. In this event all data will be erased.

The voice recorded interview will be transcribed anonymously by using your participant ID to identify your transcription. The transcriptions will be included in the writeup of my dissertation.

Usability testing an Android zone mapping app - Consent Form

This is a usability study to test the functionality and user experience of the Zone Mapping Android app.

You will install the Zone Mapping app on your Android device and use the app for up to 7 days.

After this period there will be a short interview about your experience that will be recorded (audio only).

Read and understand the attached information sheet, which includes information about the data to be recorded.

You understand that you can withdraw at any time either during the experiment or by contacting the researcher with the participant ID assigned to you, at the address provided in the information sheet, or by the email provided and your personal data will be erased from the records.

This is to confirm that you are over the age of 18 and have agreed to take part in a research study on the date: 10/03/16

Name: Joe Bloggs

Address: 1, Example Lane, City,
ABC 123

Contact Tel no: 07654321234

Email: jbloggs@example.com

Participant ID: 1234

In addition to the data analysis, you give permission for data that could be used to identify yourself (e.g. voice) to be transcribed and discussed in the writeup of my dissertation [✓]

Signed: J. Bloggs