

UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE
CSC4000W – HONOURS IN COMPUTER SCIENCE



Co-located Mobile Media Experiences Through Localized Cloud

Author:

Jarvis Mutakha (MTKJAR001)

Supervisor:

Professor Edwin Blake

Other Team Member:

Zola Mahlaza

Co-supervisors:

Pierre Benz

Thomas Reitmaier

A report presented to the Department of Computer Science, UCT, in partial fulfilment of the requirements of the course CSC4000W, Honours in Computer Science.

October 2014

	Category	Min	Max	Chosen
1	Requirement Analysis and Design	0	20	15
2	Theoretical Analysis	0	25	0
3	Experiment Design and Execution	0	20	15
4	System Development and Implementation	0	15	5
5	Results, Findings and Conclusion	10	20	15
6	Aim Formulation and Background Work	10	15	10
7	Quality of Report Writing and Presentation	10		10
8	Adherence to Project Proposal and Quality of Deliverables	10		10
9	Overall General Project Evaluation	0	10	0
Total Marks		80		80

Abstract

This report investigates the idea that cloudlet computing technology as a viable option for file sharing between co-located people using mobile devices. It goes through the problem of conveying useful interface metaphors for users to understand the available functions of the cloudlet. The design and implementation process of creation of an Android application is described and discussed and finally usability testing is performed on the finished application. The conclusions are that cloudlets are a viable option for file sharing for people within physical proximity and an effective interface can be implemented. Future work includes potential to expand the use of cloudlets other than file sharing.

Acknowledgements

I would like to acknowledge Professor Edwin Blake of the Computer Science department at the University of Cape Town for his continuous advice and coordination of meetings throughout the project.

Secondly, I would also like to thank Pierre Benz and Thomas Reitmaier for their expert advice and time they have put into helping the project come to fruition and also for acquiring the necessary hardware for the project.

Finally, I would like to thank all the students who participated in the tests carried out without their input, the final implementation would have been lower quality. I would also like to acknowledge our co-supervisor Mr Gary Stewart of the Computer Science department at the University of Cape Town for his input in the initial demonstrations.

List Of Figures

Figure 2.1	8
Figure 2.2	9
Figure 2.3	12
Figure 3.1	22
Figure 3.2	22
Figure 3.3	22
Figure 3.4	23
Figure 3.5	23
Figure 3.6	23
Figure 3.7	24
Figure 3.8	24
Figure 3.9	24
Figure 3.10	26
Figure 3.11	26
Figure 3.12	26
Figure 4.1	28
Figure 4.2	28
Figure 4.3	28
Figure 4.4	29
Figure 4.5	29
Figure 4.6	29
Figure 4.7	29
Figure 4.8	30
Figure 4.9	30
Figure 4.10	30
Figure 4.11	30
Figure 4.12	32
Figure 4.13	32
Figure 4.14	33
Figure 4.15	33
Figure 4.16	33
Figure 4.17	34
Figure 4.18	34
Figure 4.19	34
Figure 4.20	35
Figure 4.21	35
Figure 4.22	35
Figure 4.23	36
Figure 4.24	36
Figure 4.25	36

<i>Figure 4.26</i>	38
<i>Figure 4.27</i>	38
<i>Figure 4.28</i>	38
<i>Figure 4.29</i>	39
<i>Figure 4.30</i>	39
<i>Figure 4.31</i>	39
<i>Figure 4.32</i>	40
<i>Figure 4.33</i>	40
<i>Figure 4.34</i>	40
<i>Figure 4.35</i>	41
<i>Figure 5.1</i>	44
<i>Figure 5.2</i>	45
<i>Figure 5.3</i>	45
<i>Figure 5.4</i>	46
<i>Figure 5.5</i>	47
<i>Figure 8.1</i>	56
<i>Figure 8.2</i>	56
<i>Figure 8.3</i>	57
<i>Figure 8.4</i>	57

Contents

Abstract.....	i
Acknowledgements.....	ii
List Of Figures.....	iii
1. Introduction	1
1.1. Motivation.....	1
1.2. Work Allocation	1
1.3. Research Questions	2
1.3.1. Can we create an effective common sandbox for data?	2
1.3.2. What interface conceptual metaphors are effective in conveying the properties of the cloudlets as ephemeral data stores?	2
1.4. Research Approach	2
1.5. Conclusion.....	2
2. Background	4
2.1. Overview	4
2.2. Cloud Computing	4
2.3. File Sharing Between Smartphones	5
2.4. Cloudlet Computing	6
2.4.1. Using Cloudlets for External Information Processing	6
2.4.2. Using Cloudlets for Information Exchange and External Information Storage.....	7
2.4.3. Psychosocial Aspects.....	8
2.5. Interface	9
2.5.1. Metaphors.....	9
2.5.2. Users' Mental Models.....	10
2.5.3. Explicit and Implicit Actions	12
2.5.4. Direct Manipulation	13
2.5.5. User Control	13
2.5.6. Feedback	13
2.5.7. Consistency	13
2.5.8. Forgiveness	13
2.5.9. Perceived Stability.....	14
2.5.10. Aesthetic Integrity.....	14

2.5.11.	Visual Element Minimisation	14
2.5.12.	Visual Object Prioritisation	14
2.5.13.	Miscellaneous	15
2.6.	Usability Inspection.....	15
2.6.1.	Cognitive Walkthrough	15
2.6.2.	Heuristic Evaluation	16
2.6.3.	Other Techniques.....	16
2.7.	Testing.....	17
2.7.1.	Qualitative Methods	17
2.7.2.	Usability Testing.....	17
2.8.	Conclusion.....	18
3.	Design.....	19
3.1.	Overview	19
3.2.	Cloudlet.....	19
3.2.1.	Experimental Design	19
3.2.2.	Initial Anticipated Outcomes	19
3.2.3.	Brief Overview Of The Back End	20
3.2.4.	Interfacing With The Back End.....	20
3.2.5.	Front End.....	20
3.3.	Usability Inspection.....	22
3.3.1.	Design 1 – User-Based Design.....	23
3.3.2.	Design 2 – Service-Based Design.....	24
3.3.3.	Re-Design After Heuristic Evaluation	25
3.4.	Conclusion.....	26
4.	Implementation and Testing.....	27
4.1.	Overview	27
4.2.	Participant Observation	27
4.2.1.	Testing.....	27
4.2.2.	Test Sample.....	27
4.2.3.	Procedure.....	27
4.2.4.	Results.....	30
4.2.5.	Discussion.....	31

4.2.6.	Re-Design After Participant Observation: User-Based Interface	31
4.2.7.	Re-Design After Participant Observation: Service-Based Interface	32
4.3.	Final Paper Prototype Draft	33
4.4.	First HCI Expert Evaluation.....	36
4.4.1.	Feedback	36
4.4.2.	Re-Design After Feedback	37
4.5.	Second HCI Expert Evaluation	41
4.6.	Usability Testing.....	42
4.6.1.	Purpose	42
4.6.2.	Ethics	42
4.6.3.	Test Sample	42
4.6.4.	Test Equipment	42
4.6.5.	Testing Procedure	43
4.6.6.	Evaluation Procedure.....	43
5.	Results From Usability Tests	44
5.1.	Overview	44
5.2.	Timing.....	44
5.3.	Required Help	45
5.4.	User Comments.....	46
5.5.	Results From Questionnaire	46
5.6.	Discussion.....	47
6.	Conclusion.....	49
6.1.	Evaluation	49
6.2.	Future Work.....	50
7.	Bibliography	51
8.	Appendices.....	56
	Appendix A: Figures	56
	Appendix B: Participant Observation Questions.....	57
	Appendix C: Cloudlet Android Application Screens.....	58
	Appendix D: Consent Form	60
	Appendix E: Usability Test Tasks.....	60
	Appendix F: Usability Test Questionnaire.....	62

Appendix G: Paper Prototype Participant Observations 72

1. Introduction

In 2002, Kraut et al [1] coined the term “strong ties”. This refers to the things which strengthen human relationships such as frequent contact, and deep feelings of affection and obligation. This project aims to investigate the use of mobile devices and cloudlets that can also be used to strengthen human relationships. Cloud computing has been a powerful computing paradigm which has changed the way we deliver services to users. We make use of cloud services because they provide us with unique ways to collaborate, stay in touch, and engage with media socially. These services have been used to connect and make file sharing easier for people across the globe. The same technologies that power cloud service providers, namely web servers, data stores, and databases, can also run at a smaller scale on embedded systems, such as the Raspberry Pi. Computer hobbyists have revealed that the Raspberry Pi could make as an attractive solution for a small lightweight server because of its solid state storage, no noise, small form factor and low power consumption [2]. This project aims to make use of a Raspberry Pi extended with a battery, WiFi radio and local storage to provide a group of co-located friends with opportunities to share and engage with media amongst themselves. These cloudlets can be used to exchange files, and document collaboration. In essence, they can be used to provide all cloud services on a lower scale. The advantages being that there is less network latency and ownership of the data remains with the co-located people (not a company providing cloud services). This research is also interested in the human-computer interaction aspects, that is, which services can be offered to enhance co-located device use.

1.1. Motivation

The cloud is generally used for data storage and service provision. Cloudlets can also be used to provide the mentioned things. The provision of a large number of cloud services in a cloudlet scenario would be beneficial, however, for this project a single service will be offered due to time constraints. There are numerous aspects the current cloud computing paradigm is lacking in, these limitations carry over to cloudlets. The following are the motivational factors of the project.

- **Information Ownership, Control and Security:** Data centres for cloud computing service providers can be located in any country in the world. Users may be concerned about where their information is stored. Users may want the ability to control where their data is stored and who has access to it. This may be due to privacy laws and government organizations in different countries.
- **Cellular networks:** Connecting to the Internet via cell phone networks is generally slower than connecting to local WiFi networks. They also introduce high charges from cell phone providers in some cases. These two factors therefore limit the use of cloud services by users of mobile devices.

1.2. Work Allocation

Development of the project has been broken down into two parts: the front and back end. The front-end is the development of an android application to interface with the server and testing it's

usability through focus groups. The back-end is the creation of a server, data storage and development of data management protocols i.e. the cloudlet platform and the file sharing service. This report will focus on the front end.

1.3. Research Questions

The project focusses on the following two research questions:

- Can we create an effective common sandbox for data?
- What interface conceptual metaphors are effective in conveying the properties of the cloudlets as ephemeral data stores?

1.3.1. Can we create an effective common sandbox for data?

A common sandbox in this case is a platform that people on different devices (potentially running different operating systems) can manipulate files. We would like to know whether there be a public conceptual table in which people connected to the cloudlet can place and remove files. This 'common sandbox' can be public and should allow for easy access, placing and drawing of files. It should also allow for the removal of shared files by the sharers. In the case of this project, since the users of the cloudlet will be co-located, making the sandbox public may be a more practical choice because it is assumed that the people would know each other and they can easily see what others are doing within their physical location (hence they are more trusting). To explain the conceptualisation better one can picture a meeting room with a conference table (which represents the cloudlet) where objects that are placed on the table can be seen and interacted with by all the people in the room while people outside the room cannot.

1.3.2. What interface conceptual metaphors are effective in conveying the properties of the cloudlets as ephemeral data stores?

Creating an interface where the conceptualisation of the cloudlet (i.e. conveying the public sandbox) easily depicts its purpose and use to new users. The cloudlet should allow users to share files hence files should only be stored on it temporarily. For example, the interface would have to convey or at least hint as to where the files are stored and whether the cloudlet is storing anything. To use the conference room example, the owner of an object can place an object on the table for everyone to interact with and the owner can go away with the object when they are done.

1.4. Research Approach

A qualitative research methodology will be followed to investigate the research questions. An Android application will be developed with heuristic and expert evaluations carried out on the design of the interface, and finally user testing will be carried out on the final implementation. In order to perform tests on users ethical clearance had to be obtained from the UCT faculty of science research committee.

1.5. Conclusion

Due to the limited scope of the project this report will focus on the file sharing aspect of the cloudlet. Other potential uses discussed in the background section will not be implemented. Thereafter,

the design process for the front end application will be presented, followed by the implementation and finally its user testing and evaluation.

2. Background

2.1. Overview

This chapter will go through aspects the researcher believes affect cloudlet computing. First, cloud computing and file sharing on smartphones will be discussed briefly. Cloudlet computing will then be discussed showing potential uses and the psychosocial aspects that could affect mobile co-located media sharing.

Finally, aspects of the interface that may be important to the design of the front end will be discussed. There are other questions the project would also like to answer for example how cloudlets will actually work and which structure & features are more likely to be preferred by users. Obviously, this question cannot be answered until the system is implemented and deployed for practical testing.

Since the researcher carrying out this experimentation is dealing with the front end of the project, the interface and its design are likely important aspects of the research. There are some aspects of interface design that are considered important in general. According to Apple Inc. [3], some of the fundamental principles include the following.

2.2. Cloud Computing

Cloud computing is a global phenomenon. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [4]. We make use of cloud services because they provide us with unique ways to collaborate, stay in touch, and engage with media socially. These services have been used to connect and make sharing easier for people across the globe. Cloud computing has enabled WhatsApp [5], a small company of 55 employees that operates a proprietary, cross-platform instant messaging subscription service for smartphones that uses the internet for communication, to scale to 18M users and handle spikes in demand. This is because instead of messages being transferred directly from mobile device to mobile device, messages are stored in the company's servers using the internet and passed along thus allowing messages to be accessed at anytime and anywhere (provided there is an internet connection). The company primarily uses cloud storage to store any sent files like images, video and sound clips and it was estimated to host 50 billion messages daily as of January 2014 which some mobile analysts claiming that it rivals global SMS volume [6].

Cloud computing is broken down into 3 parts: first, software-as-a-service where this type of computing delivers a single application through the browser to multiple users using a multitenant architecture; second, platform-as-a-service where there is a development environment provided by the cloud which allow users to build their own applications that can run on the cloud provider's infrastructure; third, Infrastructure-as-a-service where computation and storage facilities are provided that can be accessed from anywhere [7]. The main advantage of cloud computing is it allows for massive scalability because services, computation and/or data have been moved off site. This allows for ubiquitous access from nearly anywhere in the world with an internet connection. In turn this allows for possible economic advantages if the resources and purchasing power of very large-scale multitenant

data centers are shared [8]. A good example of this was an engineering services company's internal cost to for provision of a gigabyte of managed storage was \$3.75 per month, while Amazon was charging between 10 and 15 cents per month. Originally, ISP costs at the firm were \$3500 per megabyte per month. After examining the cost structure of companies such as YouTube, the engineering services company assumed that YouTube's costs were in the teens. Taking advantage of network peering arrangements and consolidating the company's interfaces to a place close to the ISP's POP (point-of-presence) have brought costs down to YouTube levels [8].

Cloud computing technology also affects the design and implementation of wireless access points, whose purpose was to interconnect devices but now also connects them to the internet and the cloud. Currently it is not possible to create a wireless access point to share files locally on current mobile platforms without also sharing your 3G internet connection. Using cloud services may not be the most efficient way of sharing files with the person physically next to you. This is mainly due to the fact that network latency could slow the whole process down considerably.

2.3. File Sharing Between Smartphones

Transferring files between smartphones currently is not a simple task. The main reason behind this is there are various different ways to transfer files which are based on the fact that there are many types of smartphones available and each method depends on the smartphones involved. Some of these methods aren't interoperable because Windows Phone, iPhone and Android all have their own methods [9].

If one wants to share files easily on Android they could use Android Beam where files can be transferred by pressing the two devices back-to-back. Unfortunately, they would need an Android device running Android 4.1 or later and that device would also need to have an Near Field Communication (NFC) chip installed. It also works fine for sending image files but not for every type of file. iPhones don't have NFC chips installed and although Windows Phone and Blackberry devices do have the hardware, Android Beam isn't compatible with them [9]. As for the technology, the data transfer actually happens via Bluetooth. NFC is used to create a short-lived Bluetooth connection without any slow pairing process because Bluetooth and other hardware related processes don't need to be manually enabled, Android handles everything automatically [10].

Most smartphones have Bluetooth hardware installed however file transfers only occur within the same platform with Android supporting file transfers of different types and Windows Phone allowing a number of limited file types [11] to be transferred. Apple's iPhone doesn't currently support Bluetooth file transfers. To be able to transfer files cross-platform using Bluetooth third party apps are usually needed, with several apps available for different platform combinations available freely on the app markets [12].

Since hardware-based techniques are so incompatible across different devices and platforms another method for sharing files can be done using web-based services like Dropbox. Up to this point there have not been any services that work well with iPhone's. Dropbox seems to be the most popular multiplatform file transfer method currently and it uses a cloud computing architecture [13].

Unfortunately, individuals who want to use this method would need internet access and fast internet access not available everywhere at an affordable price [14].

Sending files via email is another method. Email is platform independent and doesn't need specialized apps. The only real downside to email is file size can be limited depending on the email service provider used and both parties require an email account.

On the newer Apple devices running iOS 7 AirDrop is a service that's available to transfer files between them. Airdrop uses Bluetooth to create an ad-hoc peer-to-peer Wi-Fi network between devices [15]. The devices create firewalls around the connections and files are encrypted which makes it a little bit more secure than email file transfer. AirDrop automatically detects nearby devices that are compatible and connects. The use of Wi-Fi make it faster than Bluetooth and NFC methods but this method isn't cross platform or backward compatible.

There are many other ways to share files and most of them require downloading apps from the app stores of the various platforms. All the methods mentioned above are not well suited for large file transfers.

2.4. Cloudlet Computing

The same technologies that power WhatsApp, namely web servers, data stores, and databases, can also run at a smaller scale on embedded systems, such as the Raspberry Pi [16]. Computer hobbyists have revealed that the Raspberry Pi could make as an attractive solution for a small lightweight server because of its solid state storage, no noise, small form factor and low power consumption [2]. So it seems even a Raspberry Pi extended with a battery, WiFi radio and local storage can provide a group of co-located friends with opportunities to share and engage with media, not with the public at large, but amongst themselves. Hence, the cloud is transformed from a global dynamo to a hyper-localized, ad-hoc instantiations of the cloud: a cloudlet in short. To clarify, a cloudlet is a cloud that is not running on the internet but on a server(s) that are close in terms of number of network nodes a user has to cross (ideally the server would also be in close physical proximity, i.e. less than 100 metres, to the users of the cloudlet).

There are a handful of advantages cloudlets have over cloud technology which include: lower network latency; lower potential cost of maintenance; and users having full ownership of the data shared. These formed as motivation for this project. This project is trying to implement cloudlet technology on computers with low processing power and storage capacity (e.g. Raspberry Pi) and enable file sharing between co-located mobile devices.

2.4.1. Using Cloudlets for External Information Processing

Mobile computing devices generally have less processing power and storage compared to their desktop and laptop counterparts. This is a major disadvantage for mobile devices especially if users want to run software that requires relatively high processing power and/or storage capacity. In 2009 a group of researchers proposed that mobile computing devices could be used to carry out "compute-intensive capabilities" for example, natural language processing; speech recognition; computer vision; machine learning; and augmented reality, but not at their state at the time of writing [17]. The reason being that

mobile computing devices don't have the computing capacity to carry out such tasks at the speed that would make them practical. According to the researchers they were "resource-poor" compared to less portable computing devices, namely desktop computers. They went on to suggest a new architecture where mobile devices use virtual machine technology to run on cloudlets that are within physical proximity of the device and are access via wireless local area network. Each cloudlet would contain at least one computer with a relatively high processing and memory that is connected to the internet (with reasonable speed) and is available for use by mobile devices within the area of the LAN.

Computationally intensive tasks currently are comparable to human quality if the tasks are done on powerful computers these include language translation [18] and facial recognition [19]. Both can be practical applications if they could be used on mobile devices as well instead of non-portable computers. Both applications can be run on the cloud but they are only as effective as the speed of the connection the user has to the servers running them. Hence cloudlet technology could be a contender as a solution. They also saw the limitations of traditional web-based cloud solutions (the main limitation being the latency). The latency in wide area networks could upset the usability of systems that require cloud computation on the fly. Interactive response was shown to be inversely proportional to latency [20]. In this example, the main advantage of a cloudlet over the cloud is minimized latency due to a one hop jump between the mobile device and the network that does the computation.

Lagar-Cavilla et al [20] came to similar conclusions about using cloudlets to reduce these problems; specifically the speed of cloud-based applications could be increased if they were run on cloudlets that are closer to the user. The major differences being that they proposed using cloudlet technology to carry out computations and the cloudlet technology is implemented on relatively more powerful computers.

The pattern in reasoning seems to follow on with other researchers proposing the implementation of the computationally expensive MapReduce framework on Virtual Machines via cloudlets [21]. The main reason here was MapReduce had large overheads in I/O virtualization and managing storage and computation proved difficult. The virtual machine is built on multiple physical machines that carry out complex computations that are shared amongst them.

2.4.2. Using Cloudlets for Information Exchange and External Information Storage

As mentioned earlier there are other questions project would also like to answer for example how cloudlets will actually work and which structure & features are more likely to be preferred by users. Obviously, this question cannot be answered until the system is implemented and deployed for practical testing. But there is some literature that asks similar questions pertaining file and media sharing amongst users on mobile networks locally. Empirical evidence shows us those centralized cloud infrastructures are vulnerable to massive failures due to all the data being stored in one metaphorical location (the servers) [22]. Since the Raspberry Pi doesn't have a lot of space even with an external memory card added to it, it seems practical to minimize the number of files stored on the device.

Since accessing cloud services through mobile devices suffer from slow network speed and battery lives some researchers suggest using cloudlets to reduce battery consumptions and network latency by using the rapidly increasing memory capacities on mobile devices when using traditional cloud services [23].

2.4.3. Psychosocial Aspects

A group of researchers looked into various methods in which co-located people share information with each other. The conclusion that was found was an intuitive albeit a subtle one: people's sharing patterns changed according to the situation they were in even if they were sharing with the same people [24]. This point was supported with another study conducted whereby photos are automatically shared to people in a predefined group. The people's photo sharing behaviors changed according to who was in the group [25]. Psychosocial behavior like this could be taken into account when designing the interface.

Events that take place in co-located areas can be viewed as F-formations (according to Adam Kendon [26]) where it can be used to describe the spatial locations and orientations of people in small groups (some examples include face-to-face, side-by-side or L shape arrangement) [27]. The space used during an activity is called a transactional segment [28]. In this conceptualization, Kendon defines a few terms: sustained spatial arrangements are called formations; formations with shared transactional spaces are called F-formations (Figure 1); F-formations are made up of the o-space (internal sharing space), p-space (where participants are located), and r-space (the space between the interaction and the rest of the world). These segments usually overlap during close distance social interactions and according to Kendon space and orientation of people in social interactions reflects how they could be involved with each other [29]. It seems plausible that since people's interactions with each other changes according space and orientation it may affect how they also use co-located mobile applications. This concept is simple to understand and has been used to analyse physical spaces and how they affect social interactions and how technology could be used to augment these interactions [30].

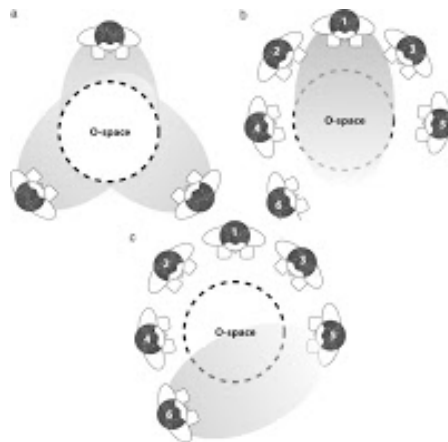


Figure 2.1: F-formations between 3, 5 and 6 people with their respective o-spaces circled in the middle [26]

Another way to look at co-located social interactions is through proxemic theory (developed by Edward Hall) which is a subcategory in the study of non-verbal communication. Proxemics can be defined as "the interrelated observations and theories of man's use of space as a specialized elaboration of culture" [31]. The theory is split into two parts: personal space and territory. Personal spaces defines the immediate space surrounding a person (figure 2), while territory describes areas which people may lay claim to and 'fight' for. Personal space is further broken down into four parts depending on the

distance from a person: intimate distance (under 18 inches) for touching and whispering, personal distance (1.5 to 4 feet) used for interactions between friends and family; social distance (4 to 12 feet) for interactions among acquaintances; and public distance (over 12 feet) for public speaking. Due to the advent of many interactive devices this theory is being used to study and create new and more effective interfaces and devices [32].

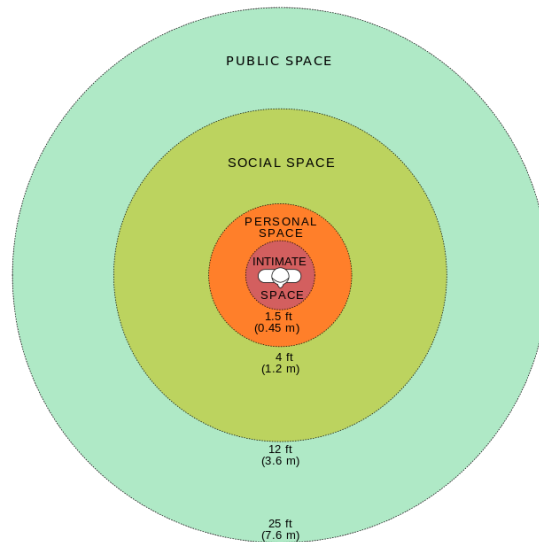


Figure 2.2: Diagram of Edward T. Hall's personal reaction bubbles (radius in feet) [33]

Another important social issue that has to be considered is privacy. Some studies on young people's sharing habits on a Bluetooth based application pointed out the importance of privacy and identity management in mobile co-located media sharing [34]. Participants in the study mentioned some issues which included: who sees what; who has control over who sees what; how they limit or specify what they are sharing and with or to whom; who owns the content. Although this particular study was based on Bluetooth technology, some of the privacy, security and ownership issues can easily be applied to wireless local area network cloudlets. It seems that cloudlets could provide an alternative to cloud-based technology to users who want to maintain information ownership.

Proxemics may be extremely useful in designing interfaces (and their metaphors) for the services that use ephemeral data [35].

2.5. Interface

Thirteen principles were selected that will be used for the interface design of the front end. What follows are the principles and the reasoning for their selection in this project.

2.5.1. Metaphors

What is a metaphor? According to Aristotle, a "metaphor consists in giving the thing a name that belongs to something else." [36]. This definition is still practical today. According to the Oxford English

Dictionary a metaphor is a thing considered as representative of some other thing [37]. What one can take from these definition is metaphors are supposed to be representations of concepts using other (usually similar) concepts.

Computers are complex pieces of equipment that use metaphors everywhere. As an example, people refer to the capacity of data that can be held by a computer as ‘memory’. For the purpose of this project the definition of a user interface metaphor is a set of user inter face visuals, actions and procedures that exploit specific knowledge that users already have of other domains [38].

User interface metaphors are applied to almost every type of graphical user interface and although they are ubiquitous in the world of interfaces some researchers believe it is still poorly understood [39]. User interface metaphors include the fundamental concepts, terms, and images by which and through which information is easily recognized, understood, and remembered [40]. Metaphors are the building blocks in the user’s mental model of a task. So it is advised to use metaphors that represent familiar concepts that make using an interface intuitive. For example, iTunes playlists and iPhoto albums, which represent real-world music playlists and photo albums. It seems appropriate metaphors will have to be designed and implemented for the application to be more user friendly with a shorter learning curve.

2.5.2. Users’ Mental Models

The mental model that this project aims to convey to users is one where the cloudlet is viewed as a virtual table that people can place files and those files can be retrieved publicly. It is assumed that users already have certain mental assumptions that mold the way they view any software they are using. The mental models are built from a melding of their previous experiences with software, hardware and other real world experiences that don’t involve computers directly. A good example is given by Apple within its human interface guidelines [3] where they point out that users have real world experience writing and mailing letters and most users have used email apps to send and receive email. Based on this, users have conceptual models of the task and they have assumptions, like the ability to create new letters, choose recipients and posting letters. An email app that ignores the user’s mental model and does not meet at least some of the user’s expectations would be difficult and even unpleasant to use. This is because such an app imposes an unfamiliar conceptual model on its users instead of building on the knowledge and experiences those users already have.

Many software companies advise software developers to find and evaluate target users’ mental models of tasks their apps help them do. This allows the developers to discover potential metaphors which could help simplify certain components of the tasks.

Apple Inc. go on to advise software developers to add a few characteristics to interfaces so as to support mental models of users which include:

- **Familiarity** – Since the user’s mental model is based purely on experience the software developers should augment user interfaces with lingo and symbols that are used in the task at hand. Many media-playing apps use the traditional forward, play/pause, stop and back buttons that have been used in radios for decades worldwide. This makes it easier for users from

different parts of the world (who don't speak the same languages) to intuitively understand how to use the players (See appendix A, Figure 8.1).

- **Simplicity** – Since the user's attention is limited, the interface shouldn't display too many attention grabbing elements for any given task. Streamlining the interface allows for quick manipulation of objects. Thus most interfaces display the basic and most used features first and the optional features (for more advanced users) are placed in menus out of the way. Zipf's principle of least effort applies to this area very well where it is postulated that animals, people and even well designed machines will naturally choose the path of least resistance [41]. Zipf's principle can be rewritten as make frequent things easier to do and unlikely things harder to do. This is easily applied to user interfaces by placing the most commonly used features of an app first while supplementing with menus for performing additional tasks (See appendix A, Figure 8.2).
- **Availability** – This involves simply by making key features and options more available depending on the needs of the user. Important components shouldn't be hidden too deeply in menus and submenus. Zipf's principle of least effort also applies to this. Another theoretical principle that applies to this aspect is Pareto's principle which states that, for many events, roughly 80 percent of the effects come from 20 percent of the causes [42]. So for interfaces 20 percent of the functionality should ideally account for 80 percent of the usage [43,44]. The obvious difficulty is identifying the 20 percent which is of highest priority when completing certain tasks.
- **Discoverability** – Elements of the interface should portray cues about how the user should interact with them. If an element is clickable, it helps if it appears that way. This easily applies to affordance design theories by Norman where affordances of an object are those properties of the object which give users clues as to how the object is used or handled [45]. This applies to tangible objects, like the handles on a tea set which provide an obvious affordance for holding for individuals with a hand, as well as intangible objects, like software buttons that have the texture of real buttons so that user feel as though they can touch them.

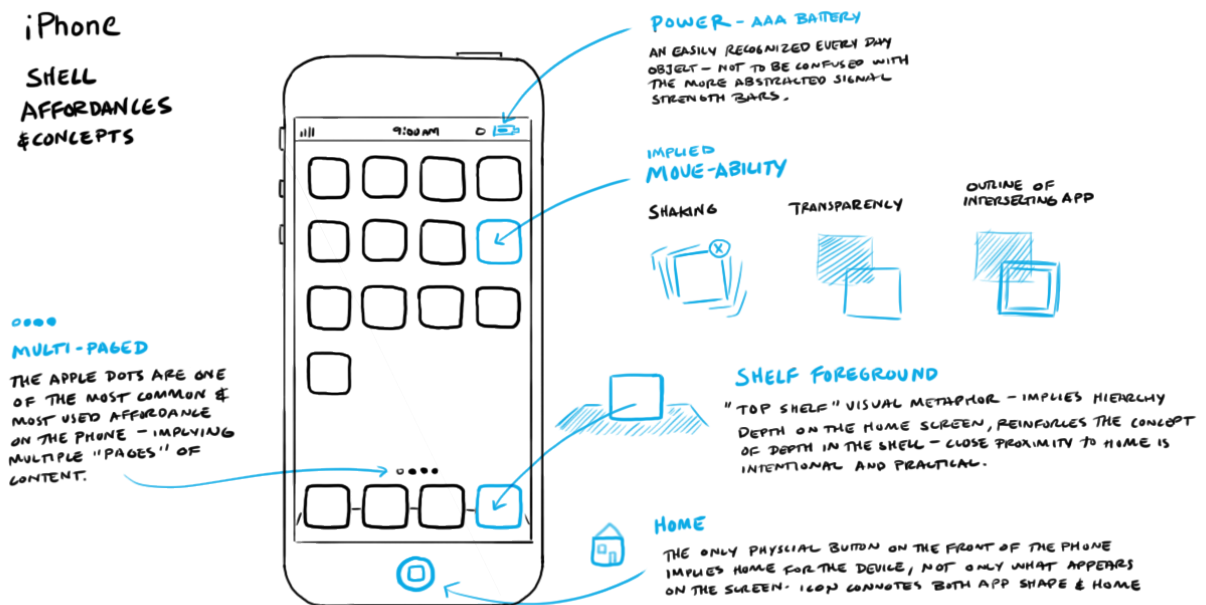


Figure 2.3: Affordances on the iOS 7 software on the iPhone [46]

2.5.3. Explicit and Implicit Actions

Every operation on every graphical user interface involves the manipulation of objects using actions. These actions usually involve two steps. As an example, the first step of manipulation of a user using a smartphone app is actually seeing and identifying the desired object onscreen; the next step the user interacts with that object (usually by selecting or assigning it); finally, the user performs an action by manipulating the object either directly or with another device like a mouse or game controller. From this example we can see that there are two types of actions: implicit and explicit actions.

Explicit actions are, just as their name implies, actions that clearly show the result of manipulation. These actions don't require any memory on the user's part because everything is displayed. Implicit actions express results of actions via visual cues and context. The pinch action to zoom in or out on most smartphone image software is a common example of an implicit action. Pinching inward implies that the image is a physical object that can be squashed into a smaller area and spreading the fingers implies that the image can be 'spread out' to take more space. For implicit actions to work, users must be able to recognize the objects involved, the manipulation to be performed and the ramifications of said action. Of course this type of action requires more memory from the user. It seems necessary for the developers of applications to take this into account because the number of implicit and explicit actions will definitely affect the learning curve and the amount of time it takes to perform tasks.

2.5.4. Direct Manipulation

It seems important to mention this principle because it is heavily tied to smartphone interface design because most smartphones use touchscreens. This form of manipulation is implicit because it gives the user the feeling of handling an object. According to the Android Design Principles, allowing people to directly touch and manipulate objects in apps reduces the cognitive effort needed to perform tasks while making it more emotionally satisfying [47].

2.5.5. User Control

Another popular design principle that will probably apply to this project is in understanding what actions to allow the user to initiate and control. Some apps constrain the user to simple actions because the developers feel that they are protecting the user from having to make complicated decisions. This may work well for novice users but may not work well for an audience that's well versed with the task and app at hand. Constraints are recommended in design to only limit the user to using interface elements the correct way (for example Lego) [48].

2.5.6. Feedback

Feedback is considered one of the most important design principles for GUIs and user interfaces in general [49,50]. Feedback in this case involves more than just error messages. Feedback also includes seemingly 'unnoticeable' interactions which people take for granted but are none the less extremely important. An example of this is buttons on screens that change their texture to a darker texture when they are pressed or clicked on. This quickly lets the user know that the action of pressing the button has been interpreted by the computer. Another good reason to provide feedback is during lengthy operations to let the user know that the app is running fine and it hasn't hanged so that they don't kill the activity.

2.5.7. Consistency

This involves designing interfaces that have similar actions and operations and use similar elements for achieving similar tasks. According to Jakob Nielsen a usability consultant "The more users' expectations prove right, the more they will feel in control of the system and the more they will like it." Users shouldn't have to relearn certain actions if they are easily transferrable to other actions in different apps [51]. Consistency also applies to actions within the same app users find it easier and faster to learn a new interface if it is consistent internally. This bodes well with the principle of least astonishment whereby users learn to use something faster if it has a lower 'astonishment factor'. This principle not only applies to graphical user interface design but all interface design in general [52].

2.5.8. Forgiveness

To encourage people to discover new things about the app actions should be made easily reversible. This gives people the feeling of freedom where they can do more without jeopardising the system. This is usually done by posting warning messages before an irreversible task is carried out. For example most games that already have a saved game and don't allow for more than one saved game (as with most mobile device games) warn the player when they are about to start a completely new game that the old save will be lost. People usually learn by trial and error and it is widely accepted that interfaces that are too sensitive to erroneous inputs discourage from learning new things because

people become too cautious and work slower to avoid more mistakes. Due to this productivity usually suffers [53].

2.5.9. Perceived Stability

This refers to giving users a predictable and familiar environment. This shouldn't be confused with consistency because perceived stability applies to the conceptual understanding for the user. For example, increased stability is where when an object(s) doesn't apply to an action it is greyed out instead of removed so that the interface only changes minimally. This helps minimise the number of 'modes' that the interface has to be in. The fewer the number of modes the more deterministic the interface appears to the user and this is directly linked to better feedback [54]. The reason being, according to Polya's principle of non-sufficient reason, if users have no reason to believe things are different they assume they aren't [55].

2.5.10. Aesthetic Integrity

This basically means that the information is displayed in aesthetically pleasing and practical ways which take into account the principles of good visual design. Norman's aesthetic usability effect applies to this design principle. The effect describes a phenomenon of how things that are perceived to look better usually perform better in usability tests. Aesthetic designs look easier to use and have a higher probability of being even if they aren't actually easier to use [56].

2.5.11. Visual Element Minimisation

People get overwhelmed when they see too much information at ones. So designers are advised to hide non-essential tasks for different contexts. Hick's law ties in well with this design principle because it states that the time taken to reach a decision increases as the number of choices a user has to evaluate increases [57]. Hick developed functions to describe this:

$$T = bH$$

Where T is the average reaction time required to choose among the choices.

$$H = \sum_i^n p_i \log_2(1/p_i + 1)$$

Where H is the information-theoretic entropy of a decision.

This has been applied internet marketing by advertisement firms around the globe who all say that the more clicks a potential customer has to make before the start buying from a site, the more choices they have, therefore the less likely they are to buy (See appendix A, Figure 8.3 and 8.4).

2.5.12. Visual Object Prioritisation

Actions are very rarely equal in importance for every task. So designers are advised to give the most important actions easier to find and faster to use. For the former Fitts' Law applies directly to it. This law states that the time taken to acquire a target is a function of the distance to and size of the target. Fitts' Law has proven to be a good predictor of pointing performance but it is only mediocre when it comes to

predicting some touch based input scenarios (like modelling small target acquisition on a large touch screen) [58]. It still shows however that distance and the size of objects on the screen are important no matter the input method. For the latter a theory that may apply is the signal-to-noise ratio (SNR). This principle is borrowed from other technical fields like electronic communication. It is summed up by the following: The ratio of relevant to irrelevant information in a display; the highest possible SNR is desirable in design [59].

2.5.13. Miscellaneous

There are many design principles that apply to interfaces in general. The following are those that weren't mentioned above but are still considered important in the design of the interface for the front end.

- **Personalisation** – People seem to like to add personal touches because it gives them a higher sense of control.
- **Preferences** – learning people's preferences or settings over time could increase reuse value because rather than asking for the same actions and decisions again and again, previous choices can be saved and places within easy reach.
- **Briefness** – Using shorter phrases can help with usability because people are less likely to skip if sentences are shorter.
- **Images over alphabet** – pictures, diagrams and icons could be used to convey concepts with less space than using words. These concepts can also be interpreted faster than reading words.
- **Conceptual location** – This ties in well with feedback. It helps if the user knows where they are relative to the home screen for an app. Using transitions between different screens is a good example to let user know they are moving to a different area.
- **Fine-grained tasks** – This requires the designers to break complicated tasks in to smaller more 'fine-grained' tasks to reduce the cognitive load on the user.

2.6. Usability Inspection

This section will discuss the various methods that will be used to evaluate the front end usability before any testing is carried out.

Usability inspection is a general name for a collection of techniques that are all based on having evaluators inspect a user interface. Normally, it is targeted at finding usability issues in the design, though some techniques also address problems like the severity of the usability issues and the overall usability of a whole system. Many inspection methods lend themselves to the inspection of user interface specifications that have not necessarily been implemented yet, meaning that inspection can be performed early in the usability engineering lifecycle [60]. The following are evaluation methods that can apply to the inspection of the usability of the interface:

2.6.1. Cognitive Walkthrough

This method involves focussing on how easy it is for new users to accomplish tasks with a system. This approach is task-specific and it uses an explicitly detailed procedure to simulate a user's

problem-solving process at each step through the dialogue, checking if the simulated user's goals and memory content can be assumed to lead to the next appropriate action.

It starts with an analysis of tasks that specifies the sequence of steps required by users to accomplish certain tasks. The designers then walk through the steps asking themselves a set of questions at each step. An evaluation of potential issues is compiled at the end [61]. For the purpose of the project only three tasks will be evaluated i.e. using the cloudlet app to connect to a nearby cloudlet; sending and receiving media files with other devices.

2.6.2. Heuristic Evaluation

This method simply involves evaluators examining an interface and judging its compliance with recognised usability principles (aka the heuristics) [62]. There are generally three popular sets of heuristics that are used:

2.6.2.1. *Jakob Nielsen's heuristics*

- Visibility of system status; Match between system and the real world; User control and freedom; Consistency and standards; Error prevention; Recognition rather than recall; Flexibility and efficiency of use; Aesthetic and minimalist design; Help users recognize, diagnose, and recover from errors; Help and documentation [63].

2.6.2.2. *Gerhardt-Powals' cognitive engineering principles*

- Automate unwanted workload; Reduce uncertainty; Fuse data; Present new information with meaningful aids to interpretation; Use names that are conceptually related to function; Limit data-driven tasks; Include in the displays only that information needed by the user at a given time; Provide multiple coding of data when appropriate; Practice judicious redundancy [64].

2.6.2.3. *Weinschenk and Barker classification*

- User Control; Human Limitations; Modal Integrity; Accommodation; Linguistic Clarity; Aesthetic Integrity; Simplicity; Predictability; Interpretation; Accuracy; Technical Clarity; Flexibility; Fulfilment; Cultural Propriety; Suitable Tempo; Consistency; User Support; Precision; Forgiveness; Responsiveness [65].

The heuristics chosen will be based off of mainly Nielsen's heuristics with a combination of the design principles that were discussed earlier. These are the final heuristics chosen: Visibility of system status; User control and freedom; Consistency and standards; Error prevention; Recognition rather than recall; Flexibility and efficiency of use; Aesthetic integrity and minimalist design; Help users recognize, diagnose, and recover from errors;

2.6.3. Other Techniques

The following are notable mentions of other inspection methods that were considered but will not be used [60]:

- Heuristic estimation is a variant of heuristic evaluation in which the inspectors are asked to estimate the relative usability of two (or more) designs in quantitative terms (typically expected user performance).
- Pluralistic walkthrough uses group meetings where users, developers, and human factors people step through a scenario, discussing each dialogue element.
- Feature inspection lists sequences of features used to accomplish typical tasks, checks for long sequences, cumbersome steps, steps that would not be natural for users to try, and steps that require extensive knowledge/experience in order to assess a proposed feature set.
- Consistency inspection has designers who represent multiple other projects inspect an interface to see whether it does things in the same way as their own designs.
- Standards inspection has an expert on an interface standard inspect the interface for compliance.

Formal usability inspection combines individual and group inspections in a six-step procedure with strictly defined roles to with elements of both heuristic evaluation and a simplified form of cognitive walkthroughs.

2.7. Testing

The proposed system's architecture will be divided up into two parts; front-end and back-end. This division implies that different research approaches will be employed. The front-end is concerned with providing intuitive user interfaces; more generally, it's interested in user experience (UX) thus a qualitative research methodology will be followed for its development. The commencing section will go through the potential evaluation methods and the reasoning behind the chosen testing methods.

2.7.1. Qualitative Methods

The three most common qualitative research methods are participant observation, in depth interviews and focus groups [66]. The researcher is carrying out participant observation. This is where there is a systematic description of events, behaviors, and artifacts in a setting chosen for study [67]. Participants will be asked to use the app and they will answer a few short questions in the end. The questions asked are in the results chapter.

2.7.2. Usability Testing

This refers to analyzing a good or service by testing it with representative users [68]. The users will be asked to try to complete typical tasks with the app or paper prototypes while they are observed. For the paper prototypes, the users will be asked to speak out loud what they are thinking during the process and they will be recorded. Established questionnaires; for example, System Usability Scale (SUS), Software Usability Measurement Inventory (SUMI), or Questionnaire for User Interaction Satisfaction (QUIS); will be considered for use because they can give more reliable and repeatable results than ad hoc questionnaires [69]. It is more likely that the QUIS questionnaire (or a modification) will be used because it measures attitudes towards certain interface factors.

2.8. Conclusion

It seems that there are a few uses of cloudlets that may be more practical using standard cloud technology especially for co-located tasks. Media sharing among co-located people using smartphones seems limited depending on the method used (for example, Bluetooth may be too slow for larger files). The interface, as with most interfaces, has to be designed with a few principles in mind. This project is not attempting to carry out computations on cloudlets. Firstly, the cloudlets in this project are meant to be used as enabling technology for file transfer between mobile devices. Secondly, one of the research questions is to check if the cloudlets can be implemented on cheaper and computationally weaker embedded systems (namely the Raspberry Pi).

3. Design

3.1. Overview

This chapter will go through the design process which the researcher carried out before the implementation of the applications and the tests carried out. It will also discuss the reasoning to many of the design choices made. An overview of the potential systems designed is described as well.

First how the research questions are to be tackled is discussed followed by the cloudlet design. Next the design of the front end application is discussed which includes the reasoning behind conceptual metaphors used for the cloudlet and the resulting design principles. Finally the design of the experiments and testing is analysed and evaluated.

3.2. Cloudlet

3.2.1. Experimental Design

The first research question wants to know if an effective common sandbox for data can be created. Therefore, the final system should allow the data to be manipulated on the application regardless of the device it is running on. Hence the sandbox is the application that allows for the manipulation of files (uploading, downloading and removing) exclusively from the device i.e. the files are not only available on a single device or a platform. In this case the application should allow any device running Android OS 2.3 or above to use the cloudlet.

As for the second research question, the interface would have to convey an appropriate conceptual metaphor for the cloudlet and its file sharing service. The metaphor we have chosen to go with is a conference table within a meeting room. The meeting room represents the cloudlet and the conference table represents the file sharing service. So to allow others to view or manipulate files they would have to be placed on the table and anyone who wants to access those files would have to be in the meeting room at the time the files are on the table.

The research methods chosen are qualitative due to the limited quantitative ways to evaluate the research questions (i.e. evaluating interfaces and the system as a whole). The research will go through the following phases and the design of the application and its interface will be improved iteratively throughout the process after every phase: first, the initial paper prototypes will be designed; next, there will be a usability inspection (specifically, a heuristic evaluation) done using the heuristics chosen in the previous chapter; next, a participant observation with paper prototypes will be done; then, an expert evaluation will be done on the interface; next, a second expert evaluation; and finally, usability tests will be carried out using focus groups.

3.2.2. Initial Anticipated Outcomes

The expected tangible products are a portable central server that runs on a Raspberry Pi that will instantiate a cloudlet and coordinate media sharing between users. An android application targeted at Android 2.3 (and above) is needed with the key features being: connecting to a nearby cloudlet, storing data on the cloudlet and sharing media with other connected devices.

The researchers expect the system to work with the following minimum requirements. First, the requirement for instantiating a cloudlet using a single Raspberry Pi coupled with the developed software solution. Users can then share media using any phones compatible with Android 2.3 and above. The expected and intangible outcome is a cloudlet which caters solely for personal social interactions. Additionally, a cloudlet which can also be easily extended to provide more cloud services on a smaller scale.

3.2.3. Brief Overview Of The Back End

The back end has a database (either Mariadb or Berkelydb) running on it to store the shared data and metadata about the clients and shared files. MQTT is used as the protocol between the different devices. MQTT is a protocol for machine to machine communication that is lightweight and uses a publish/subscribe model. It was designed for constrained devices and low-bandwidth, high-latency or unreliable networks [70]. The Paho open source project is used as the implementation of the clients for the mqtt protocol [71].

3.2.4. Interfacing With The Back End

This is the server running on the Raspberry Pi with a database store as well. A protocol was developed to help the communication of the front end and the back end. The following is a summary of the protocol. It is broken up into two parts: connecting to the cloudlet and requesting a service.

- Connecting to the cloudlet
 - Client connects to WiFi with the ssid; mac address is retrieved;
 - Client connects to mosquitto (which is an open source message broker that implements the MQTT [MQ Telemetry Transport] protocol v3.1) giving a username and mac address;
 - Client subscribes to an MQTT login channel; connection status is received on this channel;
 - Client subscribes to connected user channel;
 - Client subscribes to available services channel;
- Requesting a service
 - Client registers for receiving service request responses;
 - Client publishes their username, mac address and requested service;
 - Client waits for response on approval;
 - If client is approved, the requested service's channels are available and the client has to subscribe to them.

3.2.5. Front End

The front end was designed with the research questions in mind. To create an effective common sandbox the following design was created. The front end is the Android application.

The literature review reveals that users would like an application with very simple and intuitive interfaces with a short learning curve and a few number of interactions per function. Users also want a system to allow them control of their shared information by dictating: who has access to it; where it is stored; who owns it; and the persistence of it. The application has three main goals. It first has to be

able to connect to the cloudlet server. Next, it should allow users to find other users who are connected to the same cloudlet using the application. Finally, it should allow users to select files of any format, stored in the mobile devices, and send those files or allow visibility of them to other selected devices and also be able to store them in the central server.

As for the second research question i.e. conveying the conceptual metaphors to users, the cloudlet was initially designed to have data that was highly ephemeral. This means that if a user uploads a file to the cloudlet, the file is available to users connected to the cloudlet as long as the uploader stays connected. Once the uploader leaves, other users cannot download the file. However, if a user downloads the file and the uploader leaves, they still have the file because it is downloaded to their phone's storage. After doing some analysis and evaluations of the file sharing related use cases it was decided that this may not be very practical for the following reasons: Firstly, connections are not always guaranteed to be consistent and they may be lost within sharing sessions. If this happens then access to shared files will be unnecessarily removed from the cloudlet and users would have to upload the files again. For obvious reasons this would increase the time it takes for users to share files and this could reduce the users' satisfaction with the system. Secondly, if a user leaves the cloudlet range (which is 10 metres) other users will lose access. One of the possible solutions that were considered for this use case was having the application check if the user disconnects without signing out of the cloudlet. However, this would add complexity to the signing in process by checking if a user signing in was disconnected improperly in previous sessions before letting them connect. For these reasons the cloudlet is designed to have files for set periods of time. Users will upload files onto the cloudlet and those files will have lifetimes (15 minutes; 30 minutes; 1 hour etc.) once those lifetimes are fulfilled the files are automatically deleted from the cloudlet. If the cloudlet is reset/restarted files on the cloudlet will automatically be deleted as well.

The initial designs of the system (figure 1 - 3) had multiple services (like different lifespans of files and having groups of people within the cloudlet who share specific files with each other) which were out of the projects scope so subsequent designs dealt with only the essential service – file sharing.

The following are paper prototypes for the initial designs:

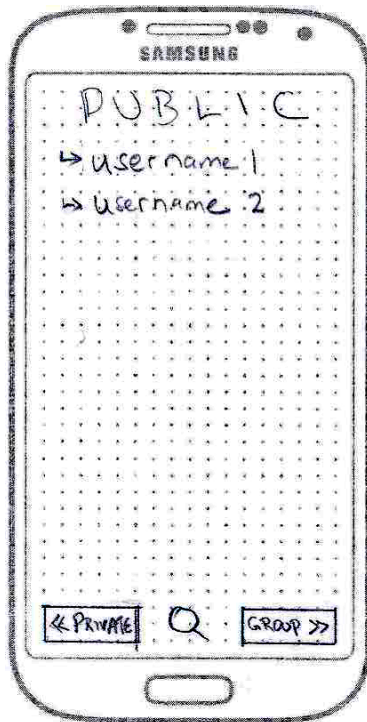


Figure 3.1: Public Screen

User selects other users currently accessed to the cloudlet; the search feature allows for searching of users and/or files; inclusion of a sort and filter feature was considered

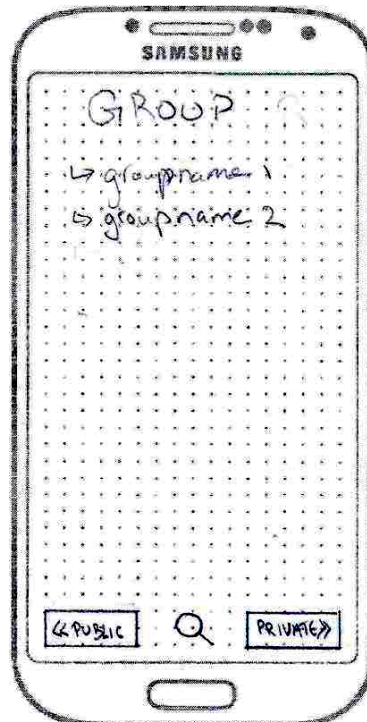


Figure 3.2: Group Screen

The group feature allows for certain files to be available to certain groups.

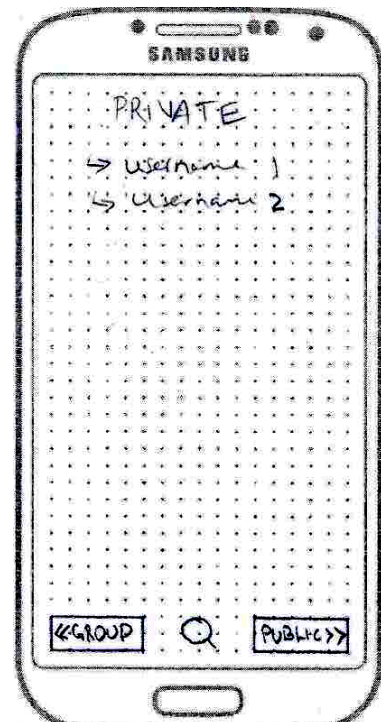


Figure 3.3: Private Screen

The private feature allows a user to open private file sharing streams with other another user.

Sort and filter features were considered as potential additions however it may not be completely needed for small groups of people because there may not be extremely many files shared in each sharing session and since the whole process occurs with users in physical proximity with each other they would be communicating verbally about the files they are sharing. The group and private feature are other features that are potentially useful additions however they may not fit in with the public sandbox conceptualization that the project aims to do. If files are only available to certain people within co-located sharing sessions it may defeat the purpose of the project.

3.3. Usability Inspection

This section will discuss the various methods that will be used to evaluate the front end usability before any testing is carried out.

The heuristics chosen will be based off of mainly Nielsen's heuristics with a combination of the design principles that were discussed earlier. These are the final heuristics chosen: Visibility of system status; User control and freedom; Consistency and standards; Error prevention; Recognition rather than recall;

Flexibility and efficiency of use; Aesthetic integrity and minimalist design; Help users recognize, diagnose, and recover from errors;

Two major designs were made and here are the heuristic inspections. Each bullet point is a heuristic and its analysis for that particular design.

3.3.1. Design 1 – User-Based Design

This design has three screens which are titled at the top. The users select users first before they can send or receive files.

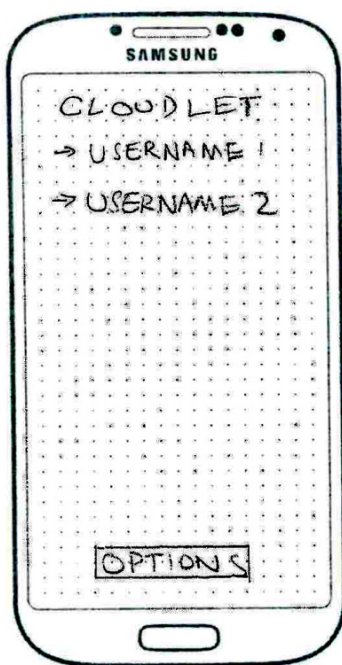


Figure 3.4: Home Screen
List of usernames of people connected to the cloudlet; User can select the users by tapping usernames; Options button brings a popup to allow for username change;

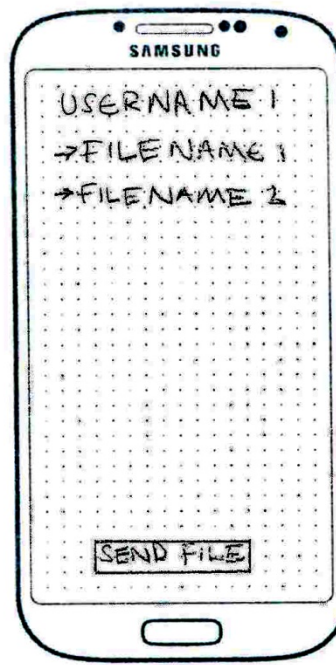


Figure 3.5: User Screen
Username of selected person is title; Shared files are listed; Send File button allows for sending of files from this device to the selected user (this button brings up a popup); Selecting a filename brings up filename screen;

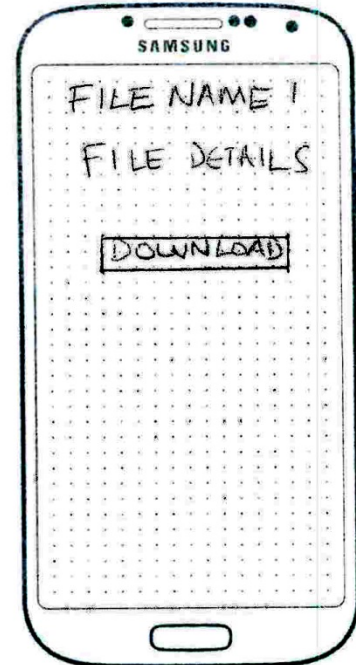


Figure 3.6: File Screen
Selected filename is the title; File details are displayed (e.g. size, date modified etc.); Download button allows for download;

- **Visibility of system status** – Title shows which screen the user is in. But there isn't a clear separation between the title and other text.
- **User control and freedom** – The user can easily go through each screen by selecting or tapping the back button on their phone. Although download is many screens away.

- **Consistency and standards** – Every screen seems to follow similar themes. Back button does what is expected to do.
- **Error prevention** – Errors are well prevented by limiting the users' possible actions to only tapping user names, file names, and buttons.
- **Recognition rather than recall** – Not a lot to remember other than tapping the back button to return to previous screens.
- **Flexibility and efficiency of use** – Due to the limited use cases the flexibility is limited. Efficiency is ok but users have to go through many screens just to upload and download files.
- **Aesthetic integrity and minimalist design** – It is minimalist. But there is only text which could be bland. Separating the different screen locations could be difficult.

3.3.2. Design 2 – Service-Based Design

Here the users select a service to begin with (although the scope of the project only allows for one service, namely file sharing) then select the user followed by other screens that are similar to the user-based design.

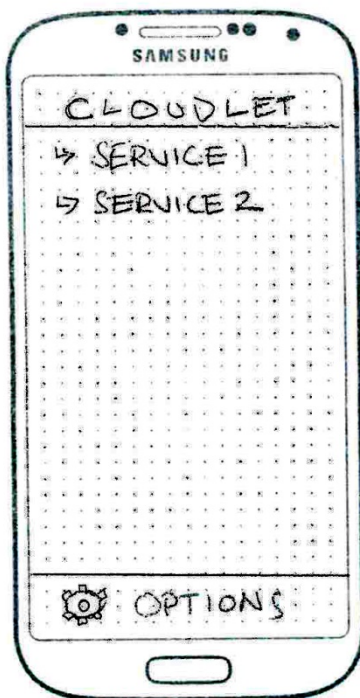


Figure 3.7: Home Screen
List of available services on the cloudlet are displayed; Title of screen is separated on top; Options allows for user name changes; User can select a service by tapping one;

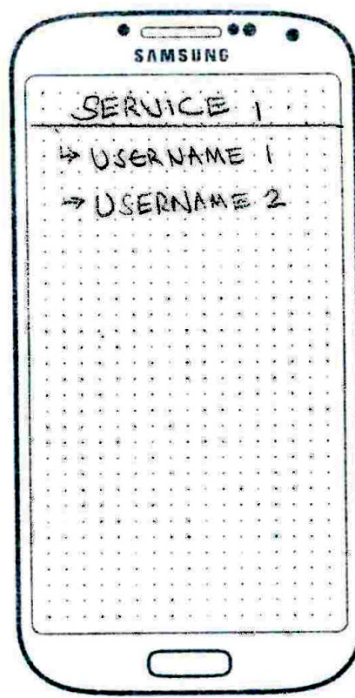


Figure 3.8: Service Screen
Service name on top; Users using the service are listed using bullet points; User can select a username;

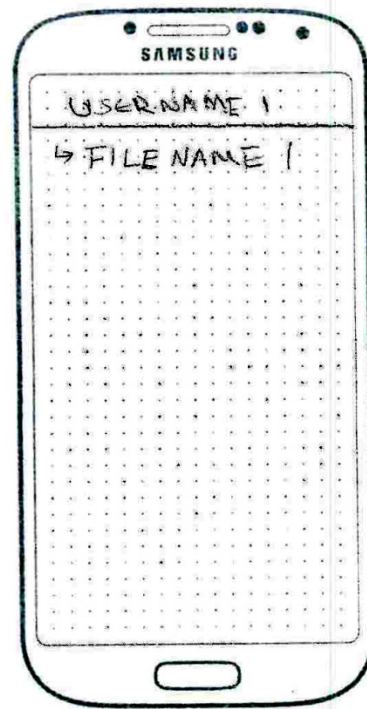


Figure 3.9: User Screen
Screen similar to the user-based design; Selecting file name will take users to a file screen where they can download;

- **Visibility of system status** – Title shows which screen the user is in and there is a clear separation between the title and the rest of the screen.
- **User control and freedom** – The user can easily go through each screen by selecting or tapping the back button on their phone. However, uploading and downloading is even further down the chain of screens.
- **Consistency and standards** – Every screen seems to follow similar themes. Back button does what is expected to do.
- **Error prevention** – Errors are well prevented by limiting the users' possible actions to only tapping user names, file names, and buttons. Having extra screens could increase the number of potential errors.
- **Recognition rather than recall** – Not a lot to remember other than tapping the back button to return to previous screens.
- **Flexibility and efficiency of use** – Due to the limited use cases the flexibility is limited. Efficiency is ok but users have to go through many screens just to upload and download files.
- **Aesthetic integrity and minimalist design** – It is minimalist. Title of the screen is well separated from the rest of the screen and the options button is also large and well separated which makes the screen more organised and appealing. The cog icon also makes it look appealing. But bullet points may not look good with many users and/or files.

3.3.3. Re-Design After Heuristic Evaluation

The major problems with both designs seen from the heuristic evaluations were the aesthetic integrity, the number of screens the user has to go through just to carry out a download/upload. The designer took this into account and redesigned the interfaces to include icons; have upload and download options earlier though the cognitive walkthrough; and separating different parts of the screens more clearly (as was seen with the service-based design).

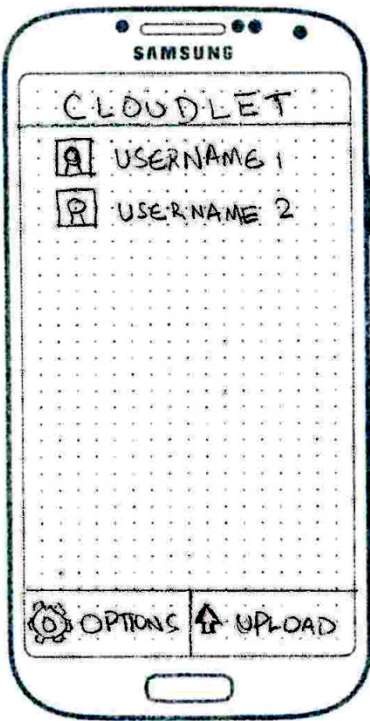


Figure 3.10: Home Screen
Icons added to make users more visible and give the user the sense that they can tap a username; Upload button placed here (popup shows that allows the user to select a file to upload to the public or a user); Icon added to the upload button

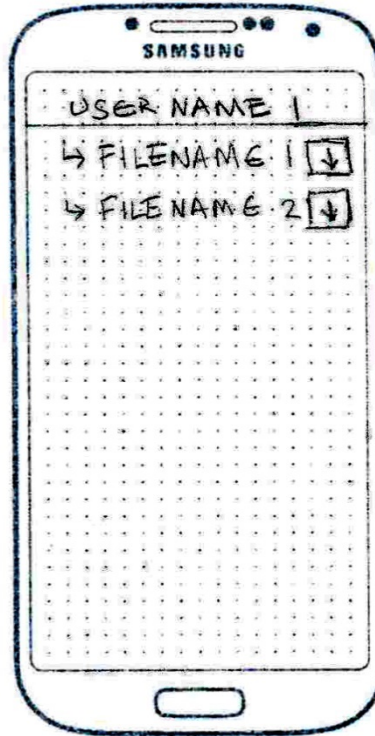


Figure 3.11: User Screen
Download button added next to each file name to allow the user to download without having to go to the next screen;

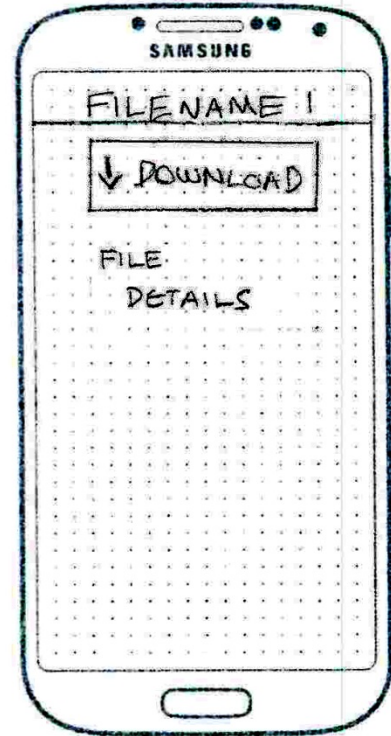


Figure 3.12: File Screen
Download button placed first above the details (therefore it is easier to spot); Icon added to the download button for aesthetic integrity;

3.4. Conclusion

The design of the interface at this point was almost complete; however, a participant observation still needed to be carried out to ensure new users could understand the purpose of the cloudlet quickly i.e. effectively conveying the conceptual metaphor (research question 2). Conveying the conceptual metaphor via the interface seems to be done to some extent with the little information at hand from the usability inspection. However, this cannot be truly evaluated until the other phases of the design process are completed because up until this point there have been no external evaluators or potential users who have used the system or seen the paper prototypes. As for the first research question, it will only be fully evaluated once the application is implemented. The next section carries on with the design process before delving into the implementation, expert evaluation and final user testing.

4. Implementation and Testing

4.1. Overview

To carry on with the final parts of the design process a participant observation done and user feedback was taken into account to design a final paper prototype. Next two expert evaluations were done with implementations of the final designs and the experts' advice was taken into account with the final implementation. Finally a qualitative user test was carried out with the final implementation. This chapter goes through that process.

Up to this point the question as to whether an effective common sandbox for data can be created couldn't be fully evaluated because it requires a working system. The system will be designed to run on Android mobile devices, specifically the Samsung Galaxy Pocket which runs Android 2.3 (codenamed Gingerbread). So the development environment chosen is Eclipse Juno with the Android Development Tool embedded.

4.2. Participant Observation

4.2.1. Testing

Participant observations were carried out using paper prototypes to help in the design of the user interface. In order to perform the participant observation ethical clearance had to be obtained. However, the participant observation did not use UCT students as participants hence no further clearance was required. Testing was carried out in a Cape Town residence.

4.2.2. Test Sample

There were 11 participants between 18 and 25 years old in this participant observation and all of them had previous experience using Android OS, iOS or BlackBerry OS. The participants are not a specialist sample because they had varying cultural backgrounds and experience with phones.

4.2.3. Procedure

Each participant went through the observation process in isolation with the researcher to keep the procedure controlled. Participants were handed a consent form to fill in before the walkthrough began. The user-based and service-based interfaces were briefly explained to each participant and a walkthrough of the paper prototypes of both interfaces was done. Finally, they were asked a few questions afterward and their comments were collected. The questions asked are in appendix B.

The following are scanned photos of the paper prototypes used in the participant observation (keep in mind these are not to scale; the paper prototypes were scale diagrams of the application running on a Samsung Galaxy S4).

User-based Interface:

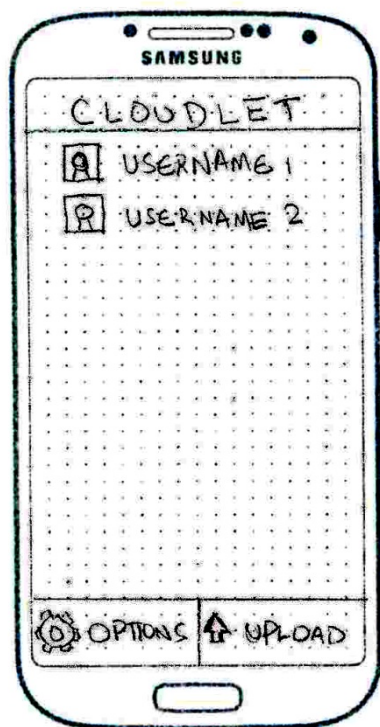


Figure 4.1: Home Screen

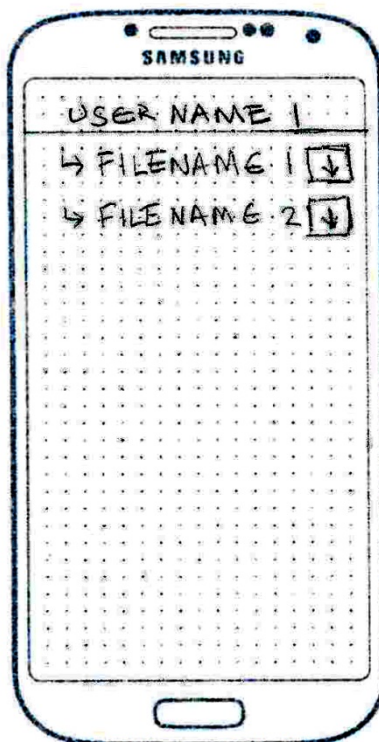


Figure 4.2: User Screen

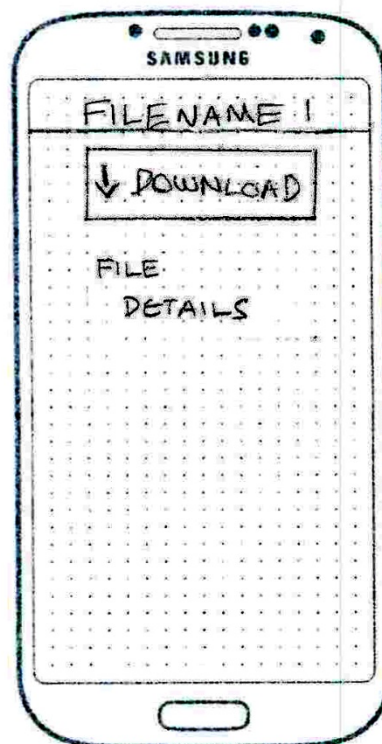


Figure 4.3: File Screen

Service-based Interface:

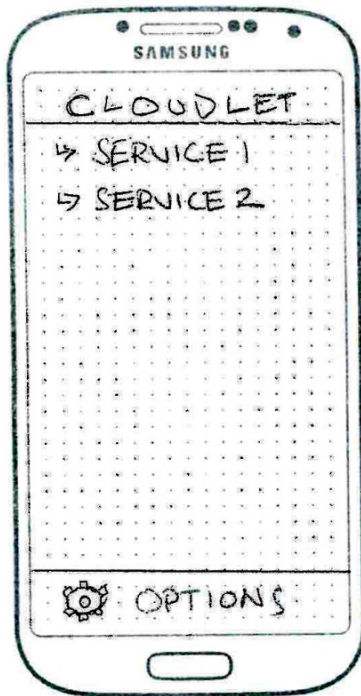


Figure 4.4: Home Screen

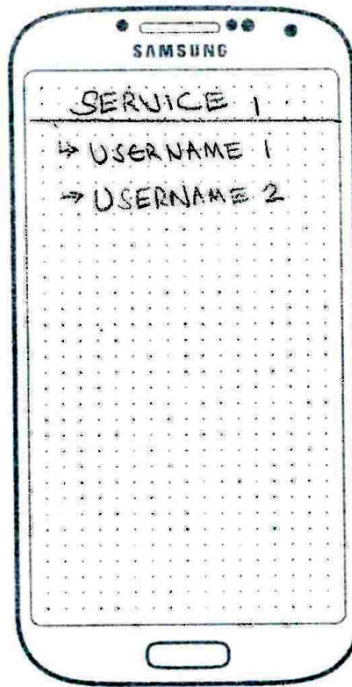


Figure 4.5: Service Screen

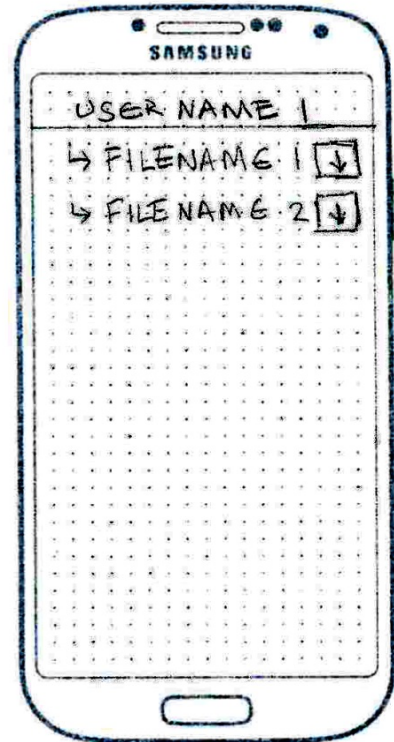


Figure 4.6: User Screen

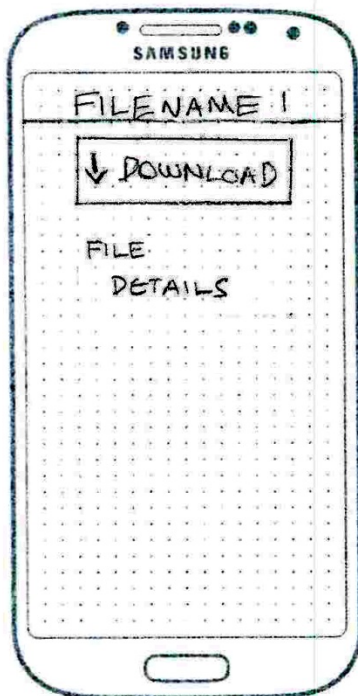


Figure 4.7: File Screen (if selected service is file sharing)

4.2.4. Results

The following is a summary of the results for each question. The pie charts indicate how many participants gave a positive/negative score for each section of the questionnaire. The bar chart indicates the average score each interface got.

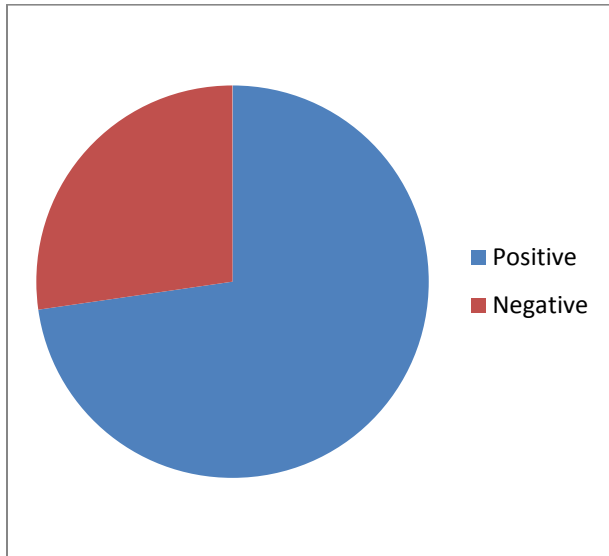


Figure 4.8: Number of Screens on User-based Interface and Service-based Interface

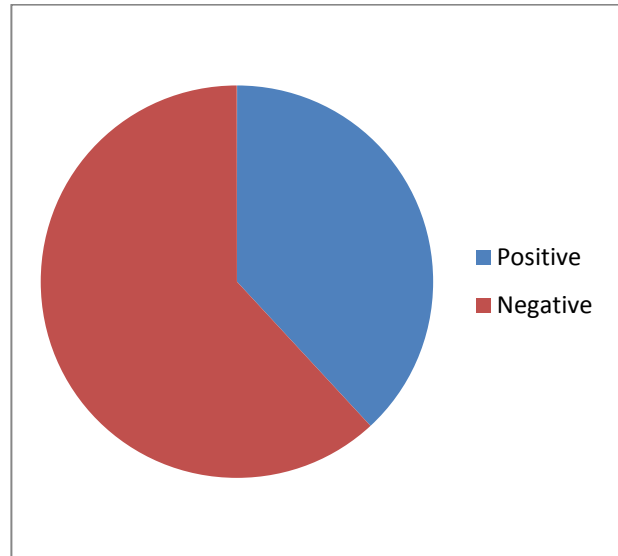


Figure 4.9: Screen real estate scores on both interfaces

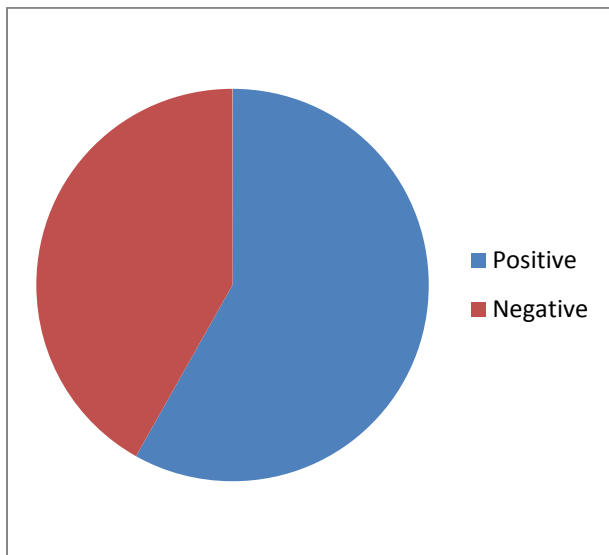


Figure 4.10: Scores for Interface Text on both Interfaces

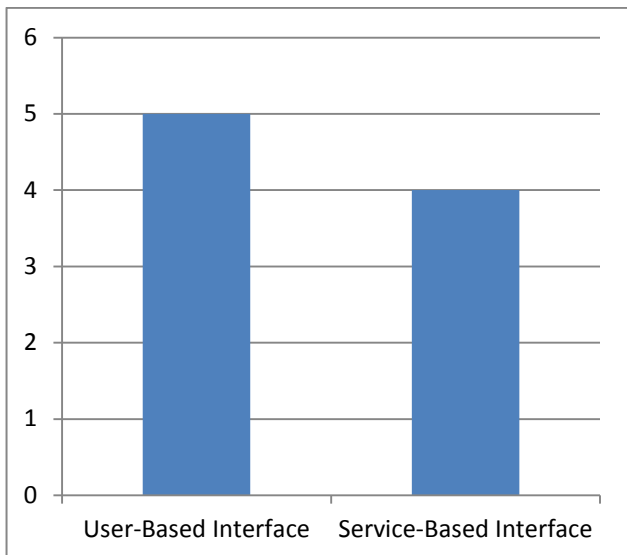


Figure 4.11: Mean Scores for each Interface

4.2.5. Discussion

A few issues were raised by participants during the tests for each section which include the following: As for the screens section of the questionnaire some users (though not many) mentioned they wouldn't use the details screen most of the time. A possible explanation for this is that it would increase the time it takes them to share files and for practical reasons, users want only the file names in most cases without bogging down on other file details (like the latitude and longitude a picture was taken).

In terms of screen real estate, some users pointed out that the download buttons within the user screen could block the text file names of the file name is long. Others found the buttons and icons too small especially if they were to tap them with their thumbs.

The text on the screen generally had positive reviews however, quite a few participants mentioned that they would prefer to discard bullet arrows and use icons instead in a fashion similar to the Whatsapp interface. Some mentioned that the bullet arrows were fine if they were augmented with the inclusion of icons.

As for the general interface type, all users preferred the user based interface because they felt that it was more intuitive and simpler to follow through. This is mostly due to the fact that there are fewer screens therefore users spend less time to do tasks and they don't have to use as much memory to remember where they are while traversing the screens.

4.2.6. Re-Design After Participant Observation: User-Based Interface

After evaluating the results the following problems were identified: there were too many screens, in particular some users said they wouldn't use the file details screen; there is no real need for the bullet arrows for the file names in the user screen; almost all the users noticed that the download button beside the file names in the user screen would block the text if the file names were long; the title text should be different sizes to the rest of the text even if only slightly; larger icons were recommended.

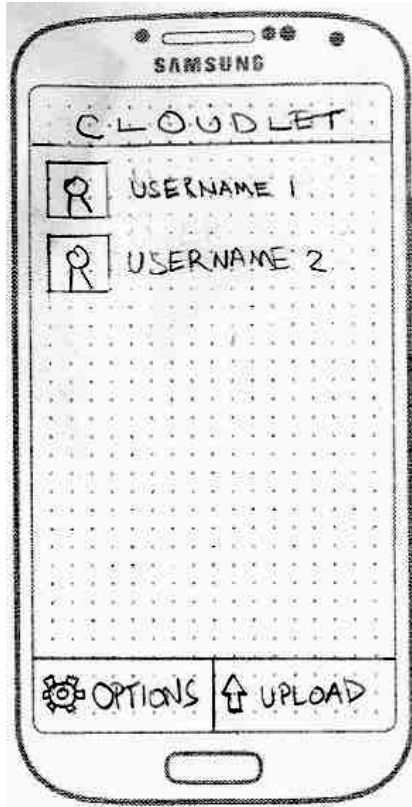


Figure 4.12: Home Screen
Increased icon size; Increased title text size;

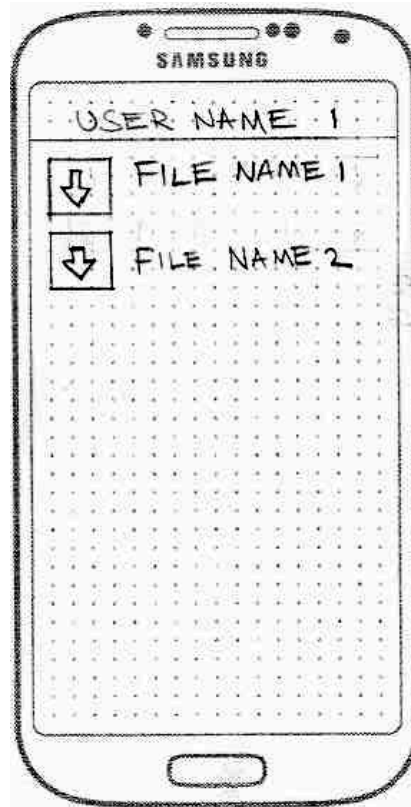


Figure 4.13: User Screen
Increased icon size; Increased title text size; File screen discarded;

4.2.7. Re-Design After Participant Observation: Service-Based Interface

After evaluating the results the following problems were identified (with some problems being the same as the user based interface problems): as with the user-based version, the last screen was not seen as necessary; the arrow bullets were not viewed as aesthetic with every user preferring the use of icons or pictures instead; the use of shortcut buttons (e.g. download buttons) were recommended on the service screen so that it was easier to use like the user-based version; icons were too small, especially considering the download button has to be tapped.

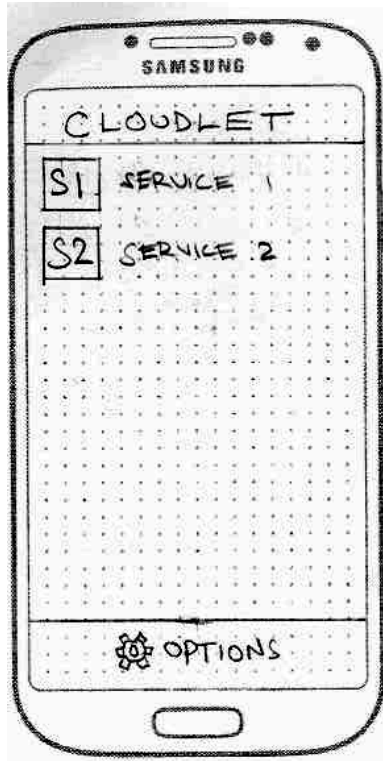


Figure 4.14: Home Screen
Service icons added; Increased icon size; Increased title text size; Arrow bullet points discarded;

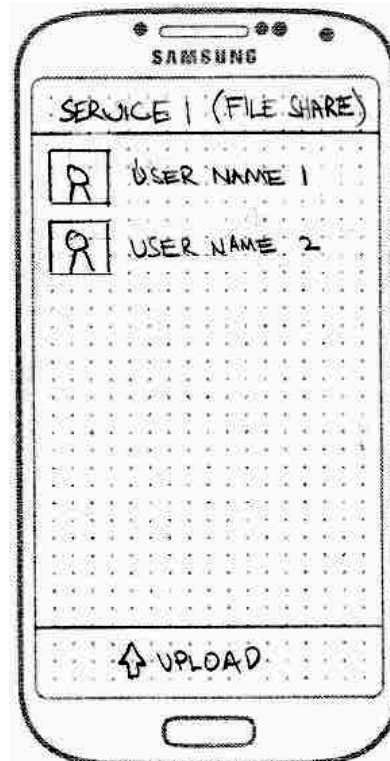


Figure 4.15: Service Screen
User icons added; Increased icon size; Increased title text size; Arrow bullet points discarded;

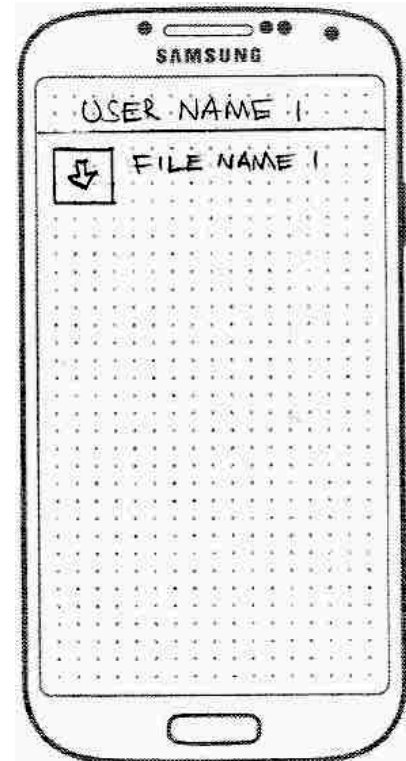


Figure 4.16: User Screen
Download button/icon added to the left side of text; Increased icon size; Increased title text size; File screen discarded; Arrow bullet points discarded;

4.3. Final Paper Prototype Draft

After careful consideration a few screens were added after the participant observation with the previous paper prototypes. The following figures depict the final paper prototype:

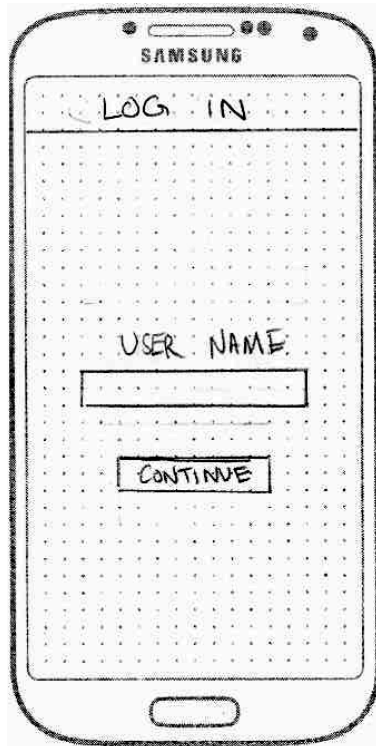


Figure 4.17: Log In Screen

User enters a suitable username that fits the confines which will be identified by other users; Continue button [sends user to home screen]

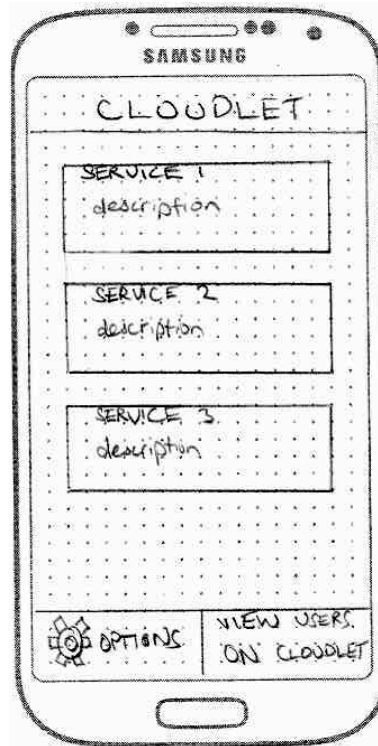


Figure 4.18: Home Screen

User selects service [goes to service screen]; Options button [goes to log in screen]; view users on cloudlet button [goes to available users screen]

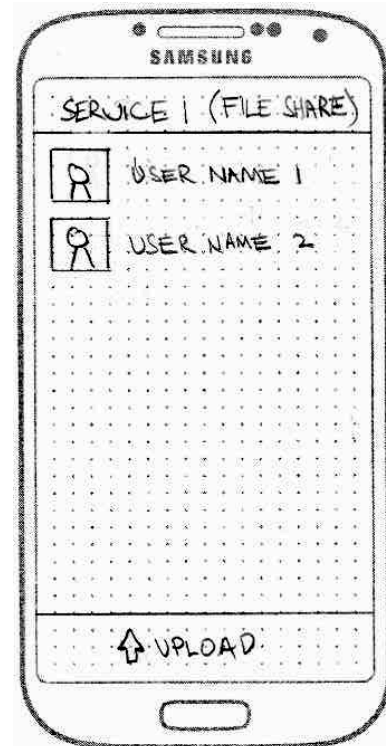


Figure 4.19: Service Screen[file sharing service]

User name [goes to user screen]; upload button[goes to upload screen];

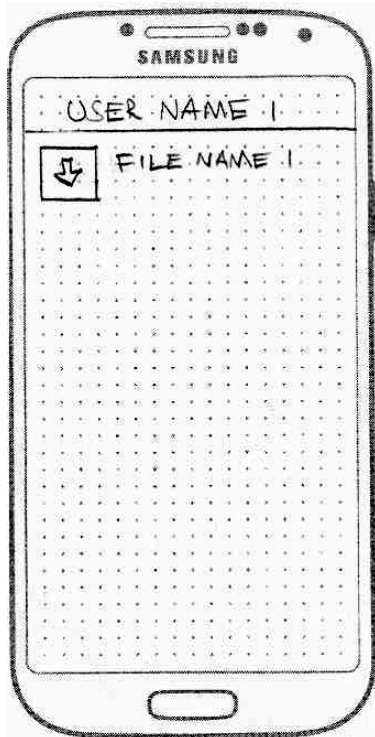


Figure 4.20: User Screen [file sharing service]

Download button [downloads file]

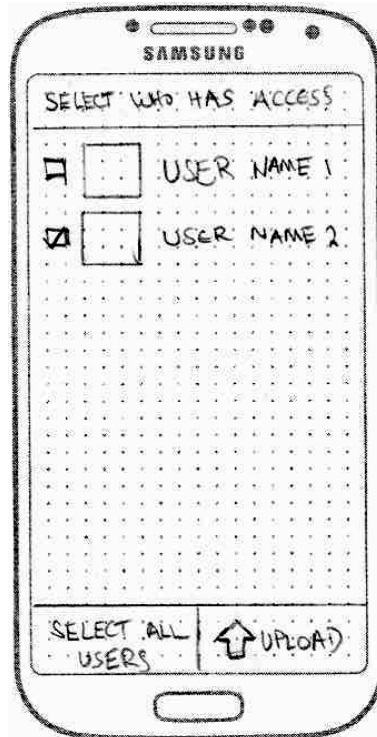


Figure 4.21: Shared Files Screen [same as file access screen]

Check box [selects user]; select all users button [fills all check boxes]; upload button [goes to upload screen]

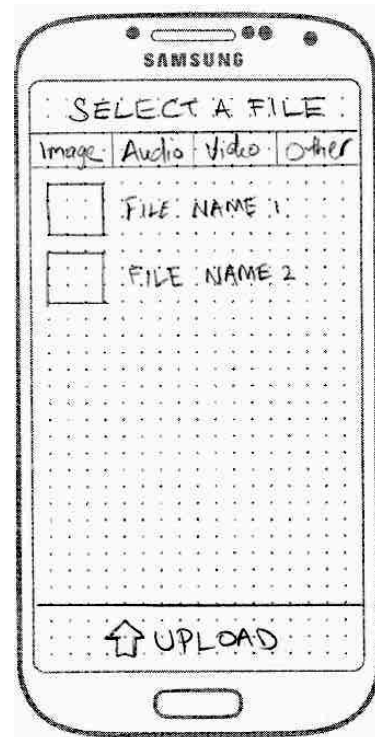


Figure 4.22: Upload Screen

Tapping file name [selects a file]; Upload button [uploads selected file]

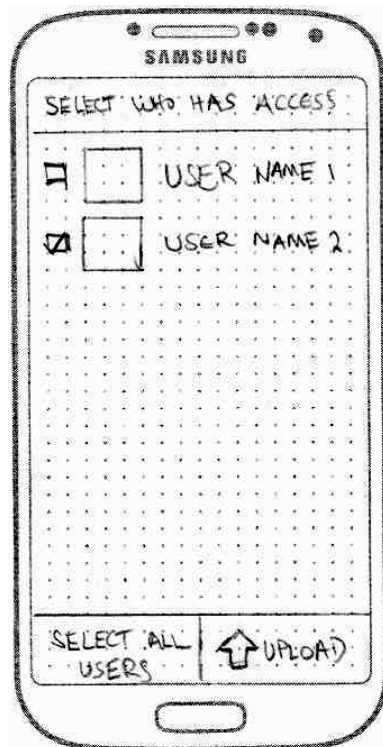


Figure 4.23: File Access Screen

Check box [selects user]; select all users button [fills all check boxes]; upload button [goes to upload screen]

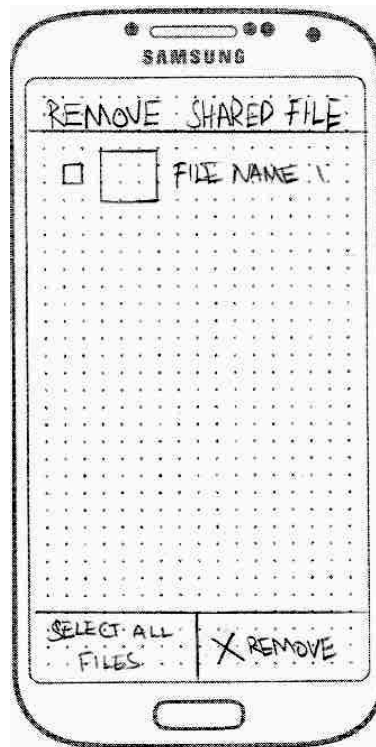


Figure 4.24: Remove Shared Files Screen

Check box [selects file]; select all files button [fills all check boxes] remove button [removes file from cloudlet]

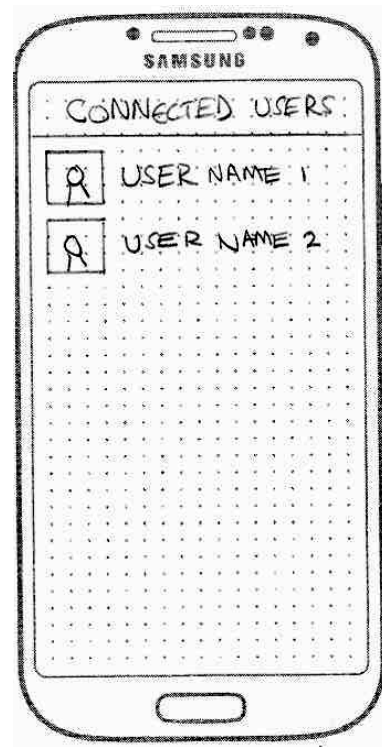


Figure 4.25: Available Users Screen

4.4. First HCI Expert Evaluation

4.4.1. Feedback

An evaluation of the design was done by an HCI expert (University of Cape Town Computer Science PhD student - Pierre Benz) on an implementation of the final paper prototype draft and several issues were raised. Firstly, the conceptualization of the cloudlet may not be easy for the users to comprehend with because the interface doesn't specify or insinuate where the data is stored i.e. users may find it difficult to know whether uploaded data is stored on the cloudlet or if the cloudlet is simply a conduit for the data without any storage occurring within it.

Secondly, to carry on from the first issue, uploaders may not know if shared data is still on the cloudlet after they leave. This issue wasn't raised by the users in the paper prototype tests. This could be because during the tests the users had someone walk them through the interface and use case processes so they didn't have to figure out using the system on their own.

Thirdly, it was recommended by the expert to place back buttons to let users who are unfamiliar with Android OS or iOS easily traverse backwards through the screens. In the initial design it was

assumed that users would be familiar with Android OS and so this implied that users would assume that tapping the back button would return them to the previous screen. This ties in with two Android design principles: *keeping the common application gestures and functions consistent with other Android apps* to reduce the learning curve for new users because they would use muscle memory to figure out the apps functions first; and *making the most frequently used and/or most important actions easiest to do* (traversing back through the screens is undoubtedly one of the most frequently used actions the users will do hence using the back button may be the fastest way to do it because users will already be accustomed to this from other apps). Another reason for not placing back buttons initially was it would take unnecessary screen real estate.

Fourthly, it was advised to view the files publicly and only view the uploaders (owners of the files) in file details, the main reason being that most users would just want to download files only while fewer users may want to know the details of the files that are on the cloudlet. Another reason to do this is that it reduces the number of screens a user has to go through to download a file.

Fifthly, an important issue with the interface is the users have to exit or minimize the app so as to view downloaded files. This goes against one of the Android design principles: *minimize user interruptions* while they are using the app. If a user has to exit the app just to view a shared file or check if a file has been successfully downloaded it would interrupt the user and minimize the immersion of the app. Exiting the app would also increase the amount of time to do use cases. The proposed solution was to add functionality to the app to allow users to view files and even open files within the app.

The final issue raised was new users may find it more difficult to know which screen they are in at any given moment. Unless users were well versed with the app they would find it hard to know which part of the process (use case) they are doing.

4.4.2. Re-Design After Feedback

The app was re-implemented with the following changes to solve the issues that were pointed out.

- The user screen was removed and users doing file sharing only have to view the files on the cloudlet to download. This reduces the number of screens a user has to go through to download a file.
- Buttons were placed at the bottom of the screens to allow users to return to previous screens. The back button functionality was still maintained to allow users accustomed to Android OS to quickly traverse backwards without having to look for the return buttons on the screen.
- File details screen was discarded as well to reduce the number of screens and therefore reduce complexity and minimise the learning curve for the app.
- File screen was edited to display the details (mainly the file uploader) underneath the file name to convey the information without having to go to another screen.
- A set of screens were added to allow users to browse through files on their phones and even open them within the app. This set of screens is also used for uploading files with the difference being when files are selected they aren't opened, rather, they are uploaded.

- Select file screen – allows the user to open their gallery; music; quickoffice (for documents); Android file system (for other files in external storage).
- Gallery screen – displays the albums within the phones gallery.
- Picture screen – displays pictures.
- Music screen – displays songs in the phones memory; allows the user to scroll through music using a simple alphabetical tuner for faster scrolling (if there are a lot of audio files).
- Music player screen – standard media player screen with play, pause, rewind, forward, volume control, timer, audio file metadata and album art.
- Quickoffice screen – browse through documents folder and external storage.
- User screen was replaced with file sharing screen – this allows the user to view the files on the cloudlet; upload files to the cloudlet; view files on the phone; remove files from the cloudlet.
- Screen titles and button names were changed to help the user know where they are within the app. The titles are helpful hints to help users know what to do within the screen.

The following are paper prototypes of the changes made to the implementation.

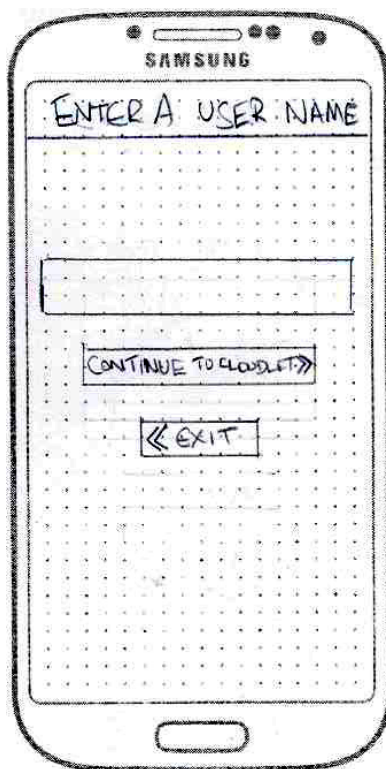


Figure 4.26: Log In Screen

Arrows insinuate direction of action; Title is a hint; Log in button is also a hint; Exit button [quits app]; Continue to cloudlet button [logs in]

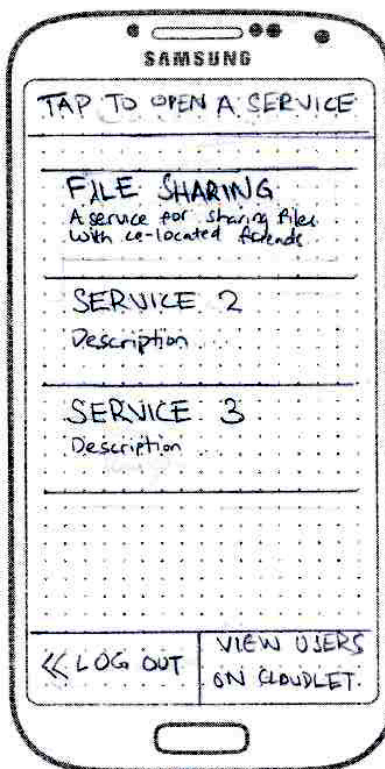


Figure 4.27: Home Screen

Arrows insinuate direction of action; Title is a hint; Log out button is also a hint; Log out [goes to login screen]; view users on cloudlet [goes to view users]

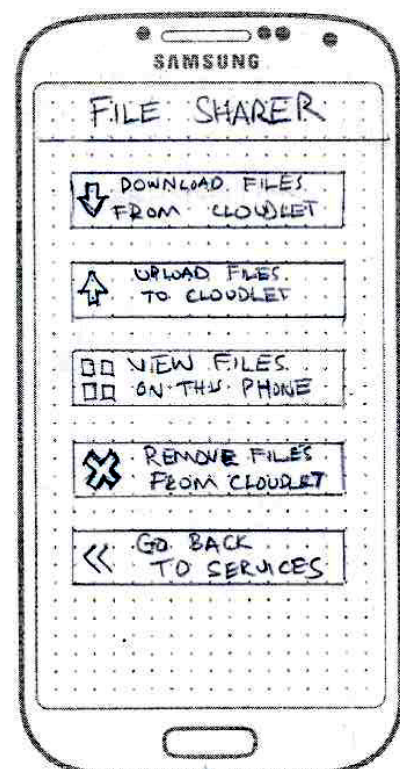


Figure 4.28: Service Screen[file sharing service]

Arrows insinuate direction of action; Back button hints location; Download button [goes to download screen]; Upload button [goes to select file]

screen]

screen]; View files button [goes to select file screen]; Remove button [goes to remove files screen]; Go back button [returns to home screen]

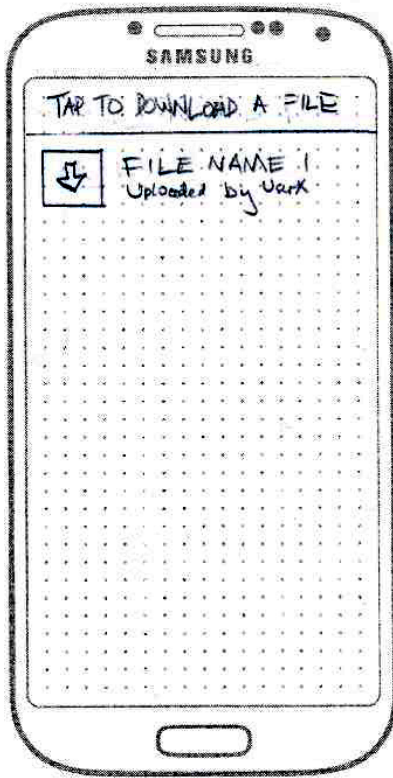


Figure 4.29: Download Files Screen

Title lets users know what to do; Uploader now included; Tapping file [downloads file]

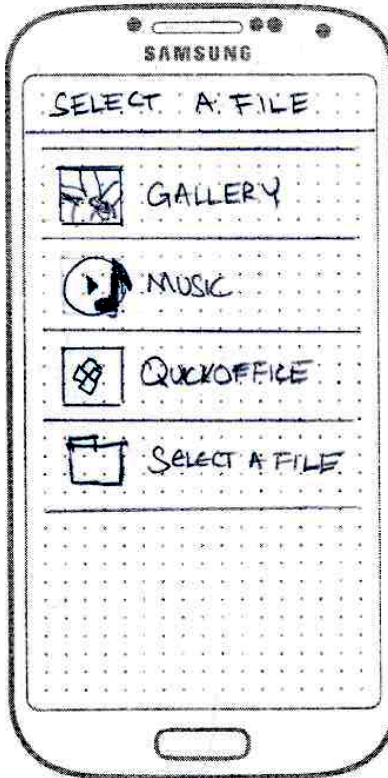


Figure 4.30: Select File Screen

Title lets users know what to do; Each item leads to their respective screens

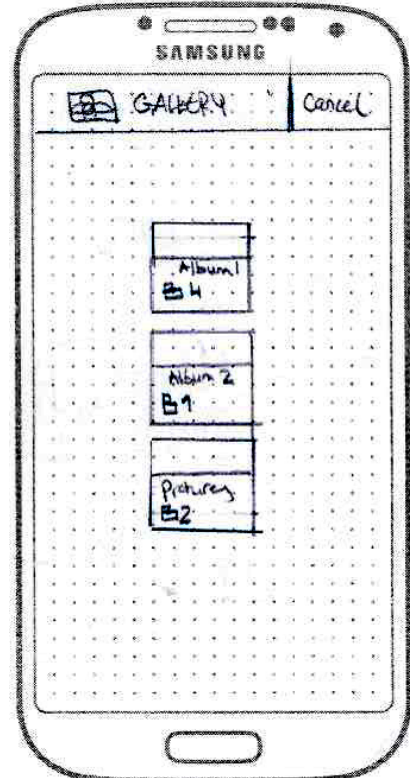


Figure 4.31: Gallery Screen

Users select an album in the phone's gallery

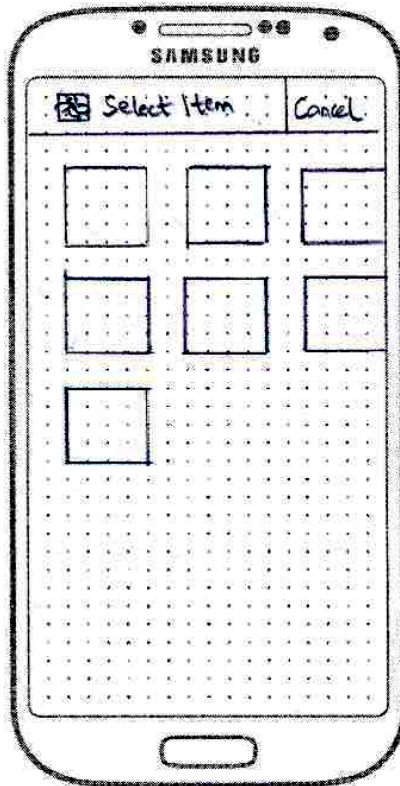


Figure 4.32: Picture Screen
User can select images within an album

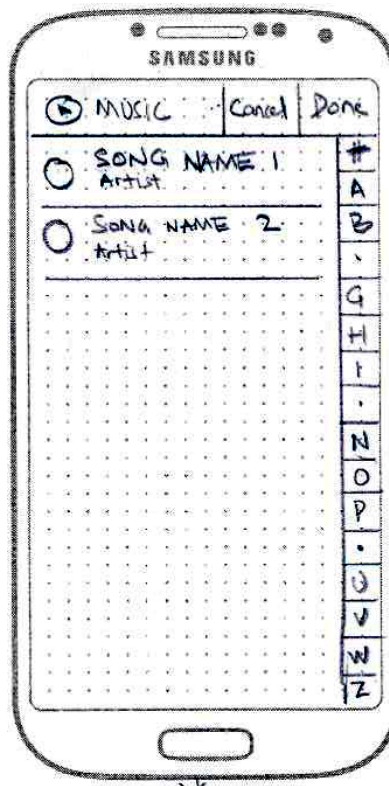


Figure 4.33: Music Screen
User selects audio file with radio button and taps done [goes to music player screen]

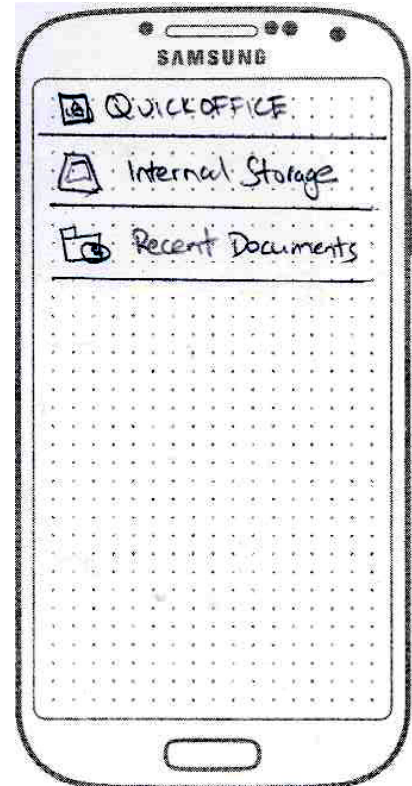


Figure 4.34: QuickOffice Screen
User moves through the phone's directories to find document files [these will be opened automatically by Android's default document apps]

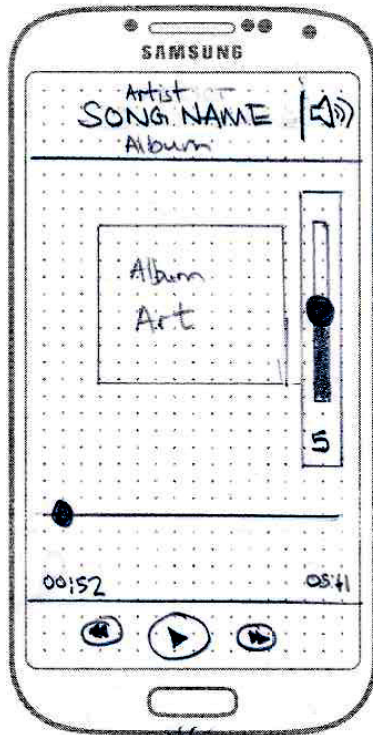


Figure 4.35: Music Player
Screen

Standard Android music player

4.5. Second HCI Expert Evaluation

The final implementation was evaluated by two HCI experts namely, Pierre Benz and Thomas Reitmaier (both UCT Computer Science PhD students) and the feedback was generally positive however some issues were raised.

Firstly, one problem with the application is that people couldn't open files on the cloudlet without downloading it. A major reason for this is that the implementation was trying to keep to the conference table conceptualisation whereby people have to take an object if they want to interact with it. The solution proposed was to potentially include a feature that allows people to open files on the cloudlet without downloading it. To follow on from this issue the experts gave an example of unintended copyright infringement and distribution of illegal media because users can't view files they have to get a copy of it first.

Secondly, they postulated allowing the users to manipulate lifetimes on all the shared media because the implemented system gives all files a lifetime of 1 hour before they are automatically deleted from the cloudlet or if the cloudlet is turned off/reset.

Thirdly, the experts discussed potentially allowing users to have private file sharing streams i.e. whereby a user shares files with only selected users and only these users can interact with the files (as with the initial designs). However, the experts did state that it could ruin the conceptualization of the

public conference table metaphor we were trying to portray to the user i.e. objects on a physical table can be interacted with by anyone near that table.

Fourthly, it was suggested that the use of confirmation dialogs would improve the usability of the interface i.e. asking users if they are sure they want to carry out certain tasks.

Finally, the experts suggested implementing passwords and encryption for security reasons. Although they did agree with some of the reasoning behind not implementing passwords i.e. since the cloudlet is public and sharing is co-located it is assumed that users will have some trust in each other as they will already know each other. Also a password automatically assumes an administrator or owner of the cloudlet. The researchers also considered implementing encryption earlier during the project but decided to opt out of encrypting files because it would increase the time it would take to share files (because files would have to be encrypted and decrypted before opening them) and as with the passwords issue, it is assumed users on the system already know each other (and have some level of trust) because they are co-located (the radius of the Wi-Fi signal is only 10 metres).

Due to time constraints these implementation suggestions couldn't be implemented. The experts did however suggest placing these implementations as potential future work for the project.

4.6. Usability Testing

4.6.1. Purpose

The purpose of the final experiment is to deliver a qualitative assessment of the final implementation of the Cloudlet Android application. The evaluation gauges whether users who represent a target user base would be interested in doing co-located file sharing using the Cloudlet app. The functions, logic, screen sequencing and interface are also tested to see if they are easily comprehended; whether the common public 'sandbox' for the data is effective in carrying out co-located file sharing; and whether the interface and its metaphors allows for users to easily grasp the properties and functions of the cloudlet. Testing took place in the UCT Computer Science Honours Laboratory.

4.6.2. Ethics

In order to perform the participant observation ethical clearance had to be obtained from the UCT faculty of science research committee. However, the participant observation did use UCT students as participants and another clearance was required.

4.6.3. Test Sample

There were 13 participants in the usability tests. All the participants were students aged between 18 and 25 with 7 being male and 6 being female. All participants had previous experience using mobile phones with varying experience using Android OS.

4.6.4. Test Equipment

The following were used in the usability tests:

- Three Samsung Galaxy pockets running Android OS with the Cloudlet app installed.

- A Raspberry Pi running the cloudlet backend with MQTT as a protocol.
- A windows phone running a stopwatch app to carry out timing.
- A user evaluation form to record observations and comments from users
- A consent form, list of tasks and a modified QUIS 7 questionnaire.

4.6.5. Testing Procedure

Participants were given a brief explanation of the study and descriptions of the types of tasks performed with their evaluation processes. For control of the testing environment, participants were tested individually. Participants were handed a consent (see appendix D) form and testing only began after they signed it and gave it back to the researcher. The task descriptions were handed to them and participants had to complete three tasks on the cloudlet namely, uploading, downloading and removing of files from the cloudlet (see appendix E). When they completed each task they had to go through a short checklist to confirm which steps within each task they were able to complete. After all the tasks were completed, participants were handed a questionnaire to fill in.

4.6.6. Evaluation Procedure

During the session, the evaluator would record the length of time it took participants to complete each task and note any time participants asked for help. The evaluation technique used during test sessions was direct observation because the researcher was seated next to each participant while taking notes of the user interactions with the system and was available to answer any queries or solve any issues users had. However, no help was provided unless the participant asked for it. Any help that was given was recorded by the researcher. At the end of the session, participants were handed a modified Questionnaire for User Interface Satisfaction (QUIS 7) (see appendix F) with sections that didn't fall into the scope of the project removed. These sections were: technical manuals and online help; online tutorials; and teleconferencing.

5. Results From Usability Tests

5.1. Overview

This section goes through the results from the usability tests of the Cloudlet app. First the timing of each task is discussed; then the help that users required is discussed; the user comments from the QUIS test are then shown; and finally the QUIS results are discussed.

5.2. Timing

The following are graphs of the times it took participants to complete each task and the average time for each task. Each bar is a participant's time with the last bar being the average time for all participants to complete that particular task.

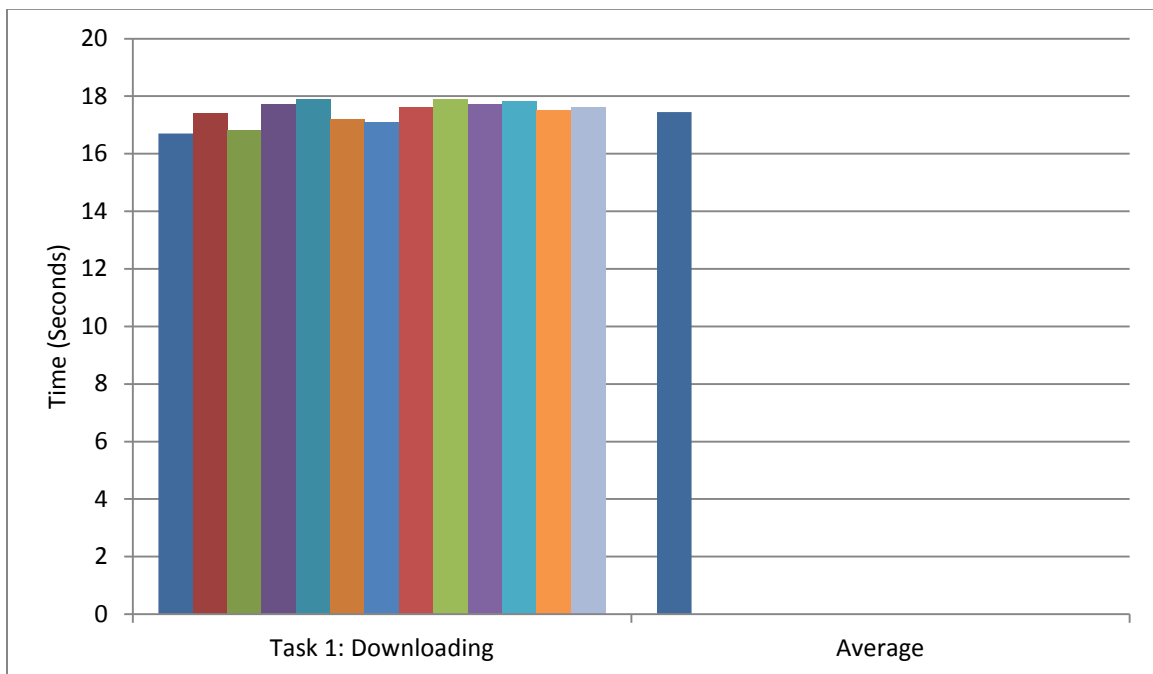


Figure 5.1: Time (in seconds) to complete task 1

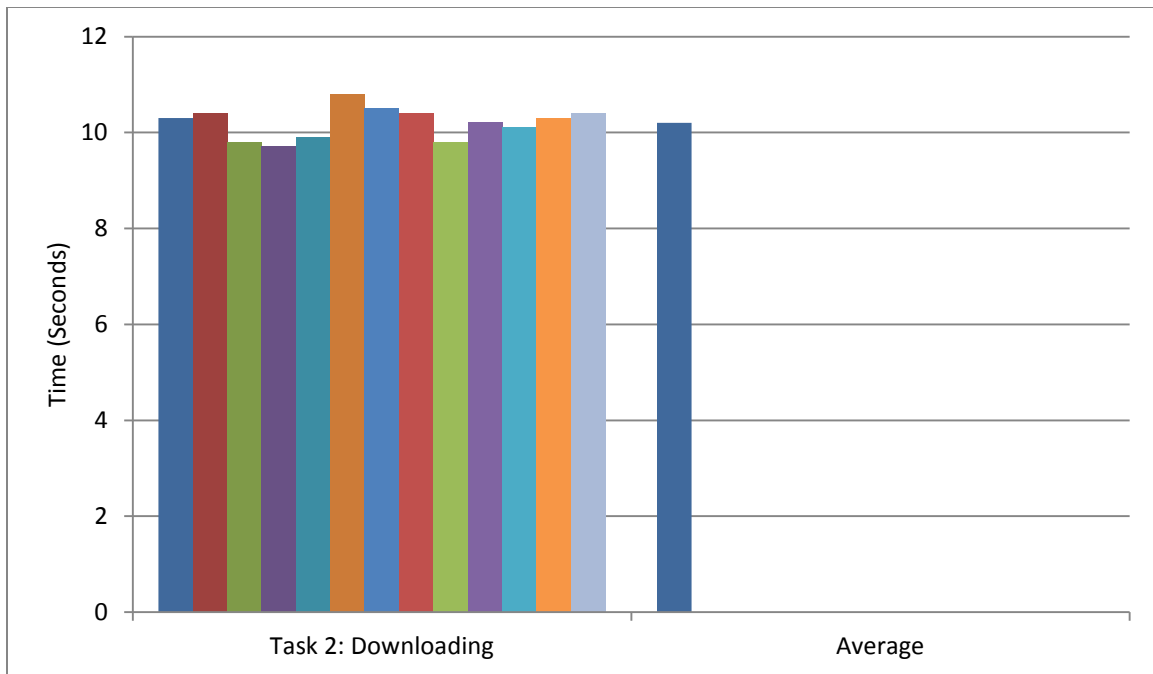


Figure 5.2: Time (in seconds) to complete task 2

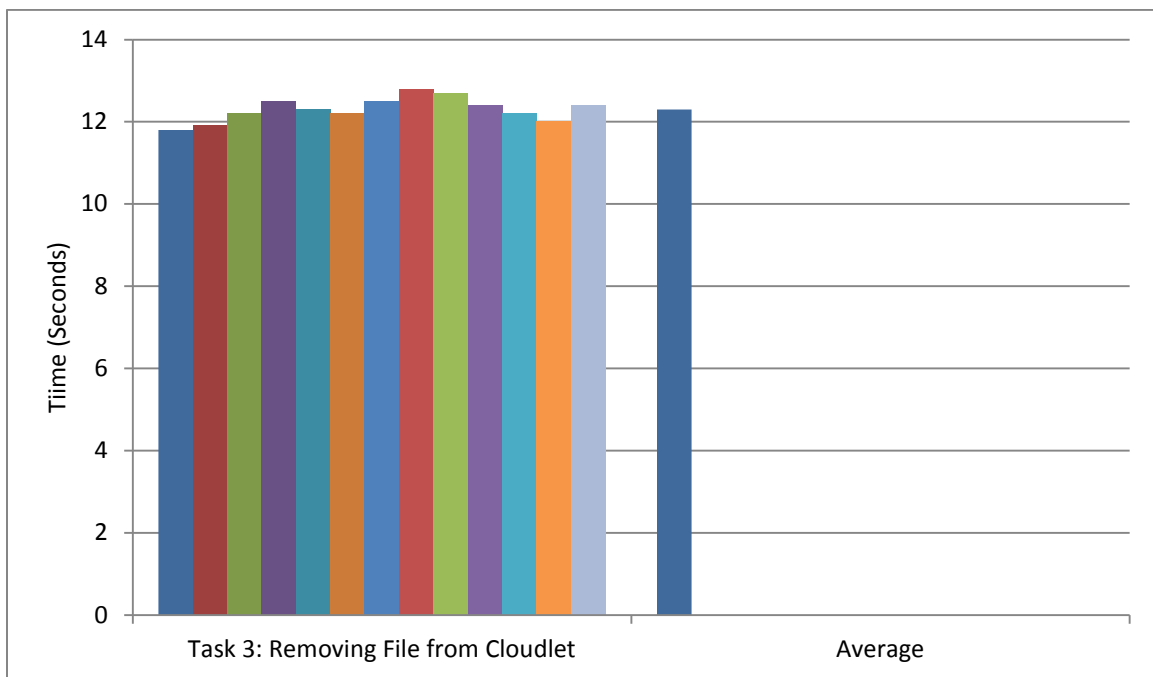


Figure 5.3: Time (in seconds) to complete task 3

5.3. Required Help

Most users didn't require help with only 2 participants stopping in the service screen because they weren't sure what to do as it wasn't clear according to them that they could tap the service descriptions (See appendix C).

5.4. User Comments

There were two main issues some users raised in the comments section of QUIS which are: firstly, 3 participants mentioned that the interface seemed dull and archaic. This may be due to the colour scheme that was implemented which only had 3 colours (namely white, purple and grey). Secondly, use of dialogues and progress bars was recommended by 2 participants to help users know that actions are being carried out in the background. For example, 1 participant mentioned using a progress bar for uploading before showing a toast that a file had successfully uploaded.

Positive comments included: the interface was easy to learn and didn't require any specialised knowledge to use; and there are a few number of steps a user has to go through to perform the tasks making it easy to use even for first time users.

5.5. Results From Questionnaire

There were 6 features of the interface that were tested. These include overall user reactions; screen; terminology and system information; learning; system capabilities; multimedia. The following are the average scores per section each participant gave. Each bar is a participant.

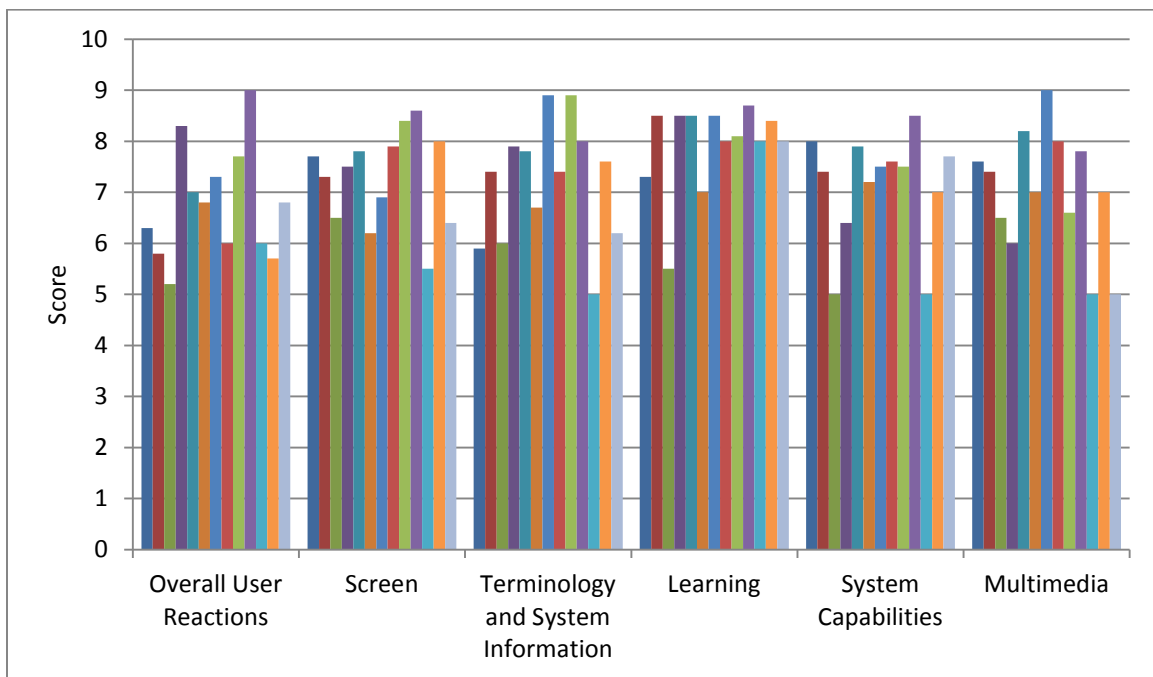


Figure 5.4: Questionnaire Results

Here are the average user scores for each section.

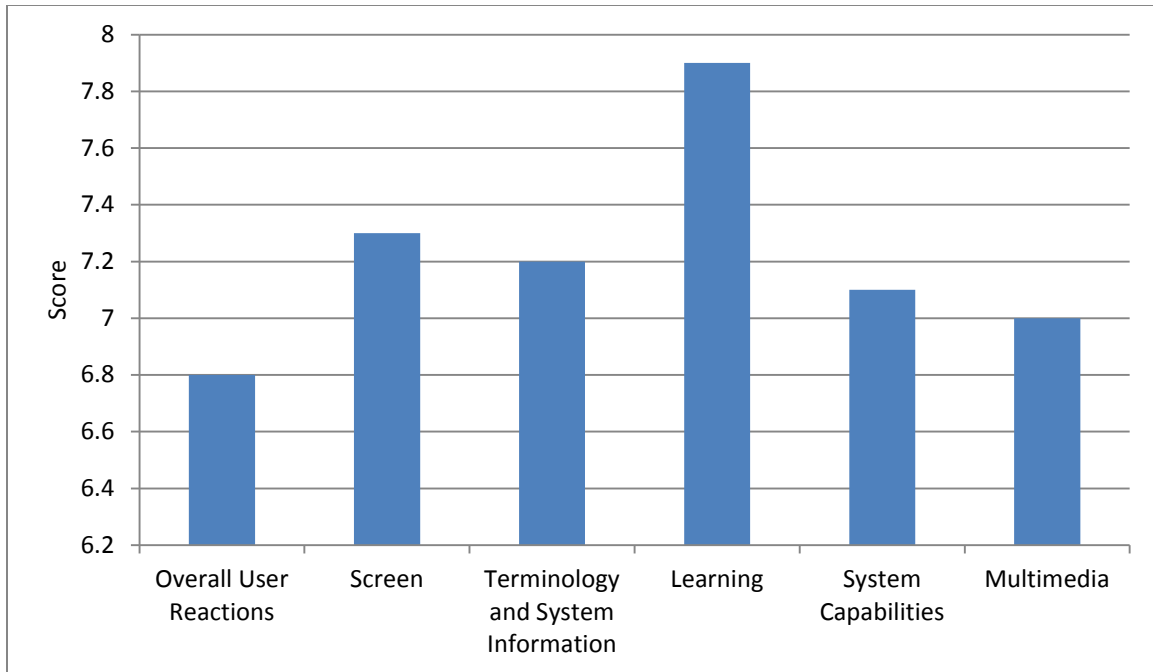


Figure 5.5: Questionnaire Results Averages

5.6. Discussion

All users were able to complete all the tasks. Most users were able to complete the tasks fairly quickly, with the first task taking slightly longer to complete compared to the other two for all the users. Users were 41.7% faster and 28.6% faster to complete task 2 and 3 respectively when compared to the times it took them to complete task 1. This can be attributed to the novelty of the interface when they first use the app to perform the first task therefore users spend slightly longer time to carry out the first task. Task 1 and 3 seem to follow similar timing patterns among the users with users who took less time to complete task 1 also getting the lowest times to complete task 3. This could be attributed to the fact that some users had more experience using Android and the interface may have been more effective and easy to use for them compared to others.

In the QUIS results there was little variance in the scores which means that most participants gave similar scores for each section. The learning scores were consistently higher than the other sections and users felt that they could learn the system easily. As evidence for this the average times decreased with the next two tasks even though the third task has more steps to do (and should theoretically take longer) compared to the first two. The screen also got relatively higher scores with users who commented positively saying that the screens were easy to navigate and the limited number of options made it easy to find the correct option. The main complaint about the screens was the buttons not being intuitive and some users suggesting the use of pictures or icons instead. All in all, the general opinion gathered from looking at the results and analysing the observations was that most users found the Cloudlet application easy to use and perform tasks with the given system.

The participants were UCT students aged between 18 and 25 and they all had previous experience with Android OS. This may not be an accurate representation of all the target users for the system. Although, they may have liked the interface other users of different ages and/or having different levels of experience with Android OS and mobile phones in general may not have produced similar results. Secondly, the app assumed users understood and read English; this isn't an accurate representation of potential users of the system. Finally, the size of the test sample may be too small to be highly accurate and a larger sample size could improve the accuracy and reliability of results.

6. Conclusion

6.1. Evaluation

This investigation set out to evaluate the feasibility of a Cloudlet running on the Raspberry Pi to allow co-located file sharing between users with smartphones. An experimental design process was used which allowed for incremental improvements on the system and its interface after each phase. Here is a general overview of the design process that occurred:

- a) Design of initial paper prototypes (P1);
- b) Heuristic evaluation of those paper prototypes (using P1);
- c) Re-design of the paper prototypes (P2);
- d) Carrying out participant observations with the paper prototypes (using P2);
- e) Re-design of the paper prototypes to get final paper prototypes (P3);
- f) Implementing the system (I1 using P3);
- g) Carrying out the first HCI expert evaluation on the implementation (using I1);
- h) Re-design of the implementation (I2);
- i) Carrying out a second HCI expert evaluation (using I2)(It should be noted at this point a third implementation (I3) couldn't be developed due to time constraints).
- j) Carrying out usability testing (using I2)

The key research questions included:

1. Determining if a common sandbox (public digital table/shelve) for ephemeral data (i.e. data that's not permanently on the cloudlet) could be implemented and used in a practical setting. This was a success but it could be investigated further because although it was implemented on the Android OS and tested on Samsung Galaxy Pockets, to fully test the practicality of a common sandbox, the application would have to be implemented on other mobile devices running on different software (e.g. having a cloudlet with Apple, Android and BlackBerry devices all sharing files). However, implementing it on different mobile operating systems was out of the projects scope.
2. Determining if there is an effective interface for conveying the cloudlet and the ephemeral nature of the file shared. This was fairly successful because most users understood the functions of the cloudlet immediately after completing a few short tasks on the cloudlet in one session. However, the final HCI expert evaluation pointed out ways in which the interface could be improved slightly to convey the conceptualisation better.

The Cloudlet app was met with fair enthusiasm with the user groups and there was genuine interest in using the Cloudlet beyond the scope of the project. The success can be attributed to carrying out evaluations and getting feedback along the design process which led to revamping the application a few times.

An interesting lesson learned during this project was the effectiveness of using different ways to evaluate the system. This allowed for flaws to be pointed out which were missed by other methods on

previous phases. For example, users didn't notice that the conceptualisation of the cloudlet (namely, the conference table) may have been difficult to get in the first participant observation however this issue was brought to the light right after in the first expert evaluation.

6.2. Future Work

Due to the very narrow scope of this project, the full potential of the cloudlet may not have been entirely investigated. The expert evaluations gave credible additional features to the system that could be implemented and tested i.e. encrypting files; interacting with files on the cloudlet without downloading them; having the users choose lifetimes of the files they share instead of them being automatically handled by the system. As mentioned in the results chapter of the thesis, a larger and more demographically diverse group of test participants would have to be used to get more reliable results. Other future work could include: first, implementing the cloudlet app on other devices that don't run Android OS for example, iOS and BlackBerry OS. This would grant access to cloudlets to more devices. Secondly, implementing more services other than file sharing. These services could include the offloading of computation to the cloudlet. Thirdly, implementing the system across multiple cloudlets and allowing data to flow through them seamlessly. This could be useful for wider geographical areas.

7. Bibliography

- [1] Kraut et al, "Internet paradox revisited," *Journal of Social Issues*, pp. 49-74, 2002.
- [2] Chris Clay. (2014, January) ZDNet | Technology News, Analysis, Comments and Product Reviews for IT Professionals. [Online]. <http://www.zdnet.com/raspberry-pi-11-reasons-why-its-the-perfect-small-server-7000025206/>
- [3] Apple Inc. (2013, January) Mac Developer Library. [Online]. <https://developer.apple.com/library/mac/documentation/userexperience/conceptual/applehiguidelines/HIPrinciples/HIPrinciples.html>
- [4] Peter Mell and Timothy Grance, "The NIST Definition of Cloud Computing," Gaithersburg, 2011.
- [5] WhatsApp Inc. (2014, May) WhatsApp : Home. [Online]. <http://www.whatsapp.com/>
- [6] Nishtha Kanal. (2014, January) Tech2. [Online]. <http://tech.firstpost.com/news-analysis/move-over-sms-whatsapp-now-processes-50-billion-messages-a-daywhatsapp-processes-50-billion-messages-day-may-overtaken-sms-216627.html>
- [7] Eric Knorr and Galen Gruman. (2011, January) InfoWorld. [Online]. <http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031>
- [8] Mache Creeger. (2009, June) ACM Queue. [Online]. <http://queue.acm.org/detail.cfm?id=1554608>
- [9] Chris Hoffman. (2013, July) How-To Geek. [Online]. <http://www.howtogeek.com/167344/how-to-easily-transfer-files-between-nearby-smartphones/>
- [10] Chris Hoffman. (2013, February) How-To Geek. [Online]. <http://www.howtogeek.com/138307/how-to-use-android-beam-to-wirelessly-transfer-content-between-android-devices/>
- [11] Codrut Neagu. (2014, January) 7Tutorials. [Online]. <http://www.7tutorials.com/where-find-files-received-bluetooth-your-windows-phone>
- [12] Will Kelly. (2013, January) TechRepublic. [Online]. <http://www.techrepublic.com/blog/smartphones/send-files-between-android-and-ios-with-fast-file-transfer/>
- [13] Idilio Drago et al., "Inside dropbox: understanding personal cloud storage services," in *IMC '12 Proceedings of the 2012 ACM conference on Internet measurement conference*, New York, NY, USA, 2012, pp. 481-494.

- [14] Ookla. (2014, August) Net Index. [Online]. <http://www.netindex.com/download/allcountries/>
- [15] Daniel Nations. (2014, January) About.com. [Online]. http://ipad.about.com/od/iPad_Guide/ss/What-Is-Airdrop-How-Does-It-Work.htm
- [16] Raspberry Pi Foundation. (2014, May) Raspberry Pi. [Online]. <http://www.raspberrypi.org/>
- [17] Mahadev Satyanarayanan, Victor Bahl, Ramón Cáceres, and Nigel Davies, "The Case for VM-Based Cloudlets in Mobile Computing," *Pervasive Computing, IEEE CS*, vol. 8, no. 4, pp. 2-11, 2009.
- [18] J Carbonell et al., "Context-Based Machine Translation," in *7th International Conference of the Association for Machine Translation in the Americas*, Boston, 2006, pp. 19-28.
- [19] A Adler and M.E. Schuckers, "Comparing Human and Automatic Face Recognition Performance," *Institute of Electrical and Electronics Engineers Trans. Systems, Man, and Cybernetics*, vol. 37, no. 5, pp. 1248-1255, 2007.
- [20] H.A. Lagar-Cavilla et al., "Interactive Resource-Intensive Applications Made Easy," in *Proc. Middleware 2007: ACM/IFIP/Usenix 8th Int'l Middlewae Conf.*, 2007, pp. 143-163.
- [21] Shadi Ibrahim et al., "CLOUDLET: Towards MapReduce Implementation on Virtual Machines," in *HPDC*, Munich, Germany, 2009, pp. 1-2.
- [22] Zhi Yang et al., "AmazingStore: Available, Low-cost Online Storage Service," *IPTPS*, p. 2, 2010.
- [23] Emmanouil Koukoumidis, Dimitrios Lymberopoulos, Karin Strauss, Jie Liu, and Doug Burger, "Pocket cloudlets," in *ASPLOS XVI Proceedings of the sixteenth international conference on Architectural support for programming languages and operating systems*, New York, 2011, pp. 171-184.
- [24] Thomas Reitmaier, Pierre Benz, and Gary Marsden, "Designing and Theorizing Co-Located Interactions," in *CHI 2013*, Paris, France, 2013, pp. 1-10.
- [25] J. Clawson, A. Voids, N. Patel, and et al., "Mobiphos: A Collocated-Synchronous Mobile Photo Sharing Application," *MobileHCI*, pp. 187-195, 2008.
- [26] Dane Galpin. (2012, September) Dane Galpin - Industrial Design. [Online]. <http://danegalpin.blogspot.com/2012/09/week-7.html>
- [27] Eyal Dim and Tsvi Kuflik. (2013, March) German Research Center for Artificial Intelligence. [Online]. http://www.dfki.de/LAMDa/2013/accepted/13_DimKuflik.pdf
- [28] Erik Berglund, Anders Larsson, and Micael Sjölund, "Smartphone Views: Building Multi-device Distributed User Interfaces," in *Mobile Human-Computer Interaction - MobileHCI 2004*. Glasgow:

- Springer Berlin Heidelberg, 2010, pp. 507-511.
- [29] Anna Esposito, Nick Campbell, Carl Vogel, Amir Hussain, and Anton Nijholt, *Development of Multimodal Interfaces: Active Listening and Synchrony*. Dublin: Springer Berlin Heidelberg, 2009.
 - [30] Paul Marshall, Yvonne Rogers, and Nadia Pantidi, "Using F-formations to analyse spatial patterns of interaction in physical environments," in *CSCW '11 Proceedings of the ACM 2011 conference on Computer supported cooperative work*, New York, 2011, pp. 445-454.
 - [31] Edward T Hall, *The Hidden Dimension*. New York: Anchor Books, 1966.
 - [32] Saul Greenberg, Nicolai Marquardt, Till Ballendat, Rob Diaz-Marino, and Miaosen Wang, "Proxemic interactions: the new ubicomp?," *interactions*, pp. 42-50, 2011.
 - [33] Wikipedia. (2014, October) Proxemics - Wikipedia, the free encyclopedia. [Online]. <http://en.wikipedia.org/wiki/Proxemics>
 - [34] Reitmaier et al, "Designing and Theorizing Co-Located Interactions," in *CHI 2013*, Paris, France, 2013, pp. 1-10.
 - [35] Saul Greenberg, Kasper Honbaek, Aaron Quigley, Harald Reiterer, and Roman Rädle, "Proxemics in Human-Computer Interaction (Dagstuhl Seminar 13452)," *Dagstuhl Reports*, pp. 1-, 2014.
 - [36] Donald William, *Aristotle Poetics*.: Lucas, 1968.
 - [37] Lesley Brown, *The New Shorter Oxford English Dictionary*. Oxford: Oxford University Press, 1993.
 - [38] Wikipedia. (2014, August) Wikipedia. [Online]. http://en.wikipedia.org/wiki/Interface_metaphor
 - [39] Pippin Barr, "User-Interface Metaphors in Theory and Practice," Wellington, 2003.
 - [40] Aaron Marcus, "Managing metaphors for advanced user interfaces," in *workshop on Advanced visual interfaces*, Bari, 1994, pp. 12-18.
 - [41] George Kingsley Zipf, *Human behavior and the principle of least effort*. Oxford: Addison-Wesley Press, 1949.
 - [42] Wikipedia. (2014, August) Wikipedia. [Online]. http://en.wikipedia.org/wiki/Pareto_principle
 - [43] Sujay Jayakar, "Pareto Optimal User Interface Design," Colorado Springs, 2011.
 - [44] Anders Toxboe. (2009, June) UI-Patterns.com. [Online]. <http://ui-patterns.com/blog/The-8020-rule--the-Pareto-principle>

- [45] Donald A Norman, "Affordance, conventions, and design," *interactions*, pp. 38-43, 1999.
- [46] mattink.com. (2013, November) inks & things. [Online].
<http://mattink.com/post/66003190793/affordance-shell>
- [47] Google Inc. (2014, August) Android Developers. [Online].
<http://developer.android.com/design/get-started/principles.html>
- [48] J Preece, Y Rogers, and H Sharp, *Interaction Design: Beyond Human-Computer Interaction*. New York: Wiley, 2002.
- [49] Peter Vukovic. (2014, August) 99designs. [Online]. <http://99designs.com/designer-blog/2014/01/15/7-unbreakable-laws-of-user-interface-design/>
- [50] Ian Pegg. (2014, February) EggCup Web Design. [Online]. <http://eggcupwebdesign.com/usability-the-importance-of-user-feedback-in-user-interface-design/>
- [51] Kyle Sollenberger. (2012, August) Treehouse. [Online]. <http://blog.teamtreehouse.com/10-user-interface-design-fundamentals>
- [52] Joshua Bloch, "How to design a good API and why it matters," in *OOPSLA '06 Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications* , New York, 2006, pp. 506-507.
- [53] Wilbert O Galitz, *The essential guide to user interface design: an introduction to GUI design principles and techniques.*: John Wiley & Sons, 2007.
- [54] Harold Thimbleby, "Character level ambiguity: consequences for user interface design," *International Journal of Man-Machine Studies*, pp. 211-225, 1982.
- [55] Stefano Levialdi. (2014, August) HCI: Human Computer Interaction. [Online].
<http://wwwusers.di.uniroma1.it/~ium/slid4.htm>
- [56] Noam Tractinsky, A S Katz, and Dror Ikar, "What is beautiful is usable," *Interacting with computers*, pp. 127-145, 2000.
- [57] William E Hick, "On the rate of gain of information," *Quarterly Journal of Experimental Psychology*, pp. 11-26, 1952.
- [58] Xiaojun Bi, Yang Li, and Shumin Zhai, "FFitts law: modeling finger touch with fitts' law," in *CHI '13 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* , New York, 2013, pp. 1363-1372.

- [59] William Lidwell, Kritina Holden, and Jill Butler, *Universal principles of design, revised and updated: 125 ways to enhance usability, influence perception, increase appeal, make better design decisions, and teach through design*. Beverly, Massachusetts: Rockport Publishers, 2010.
- [60] Jakob Nielsen. (1995, January) Nielsen Norman Group - Evidence-Based User Experience Research, Training, and Consulting. [Online]. <http://www.nngroup.com/articles/summary-of-usability-inspection-methods/>
- [61] Wikipedia. (2014, March) Wikipedia. [Online]. http://en.wikipedia.org/wiki/Cognitive_walkthrough
- [62] Wikipedia. (2014, August) Wikipedia. [Online]. en.wikipedia.org/wiki/Heuristic_evaluation
- [63] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *CHI'90 Conf.*, Seattle, WA, 1990, pp. 249-256.
- [64] Jill Gerhardt-Powals, "Cognitive engineering principles for enhancing human - computer performance," *International Journal of Human-Computer Interaction*, pp. 189-211, 1996.
- [65] Jeff Sauro. (2011, August) Measuring Usability. [Online]. <http://www.measuringusability.com/blog/he-cw.php>
- [66] Natasha Mack, Cynthia Woodsong, Kathleen M MacQueen, Greg Guest, and Emily Namey, *Qualitative research methods: a data collectors field guide*. Durham, NC USA: Family Health International, 2005.
- [67] Barbara B Kawulich, "Participant observation as a data collection method," *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research*, 2005.
- [68] U.S. Department of Health & Human Services. (2014, August) usability.gov. [Online]. <http://www.usability.gov/how-to-and-tools/methods/usability-testing.html>
- [69] Serco Ltd. (2000, December) Cost-effective User Centred Design - Satisfaction questionnaires. [Online]. <http://www.usabilitynet.org/trump/methods/satisfaction.htm>
- [70] mqtt.org. (2014, September) mqtt.org. [Online]. <http://mqtt.org/>
- [71] The Eclipse Foundation. (2014, September) Eclipse Luna. [Online]. <http://www.eclipse.org/paho/>
- [72] Brighten Studios Inc. (2013, April) Brighten. [Online]. <http://brightenstudios.com/blog/2013/4/9/hicks-law-applied-to-application-design>

8. Appendices

Appendix A: Figures

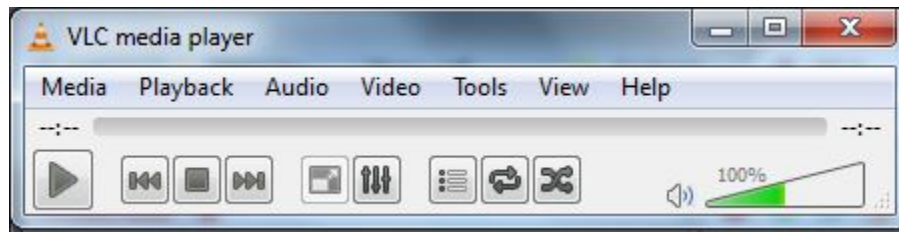


Figure 8.1: VLC media player 2.0.3



Figure 8.2: Samsung Galaxy's S-Planner

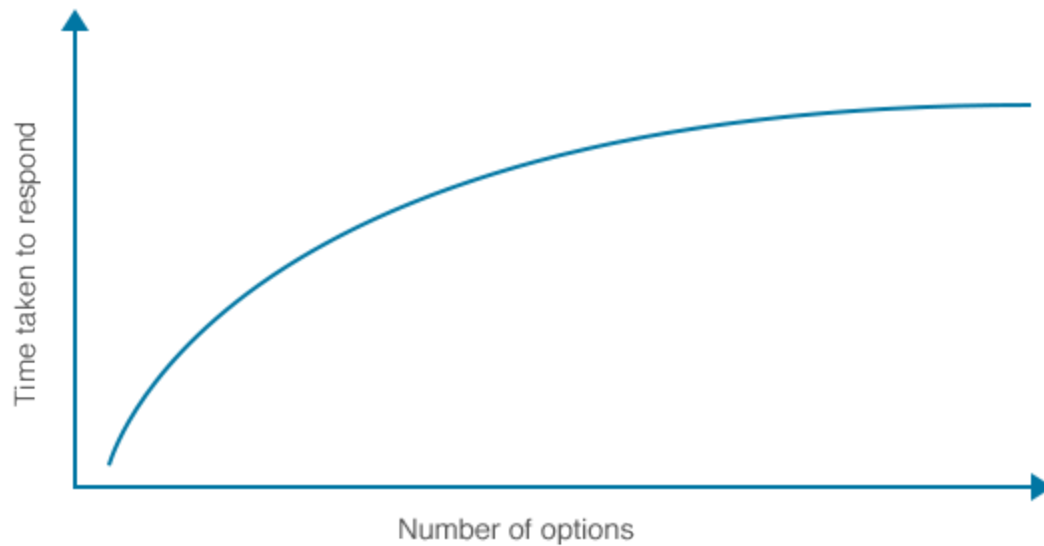


Figure 8.3: Hickaw Function Graphically Displayed [72]

Harder	Easier	Easiest
<input type="radio"/> Polar Bear <input type="radio"/> Brown Bear <input type="radio"/> Black Bear <input type="radio"/> Atlas Bear <input type="radio"/> Bergman's Bear <input type="radio"/> Blue Bear <input type="radio"/> Gobi Bear <input type="radio"/> Kodiak Bear <input type="radio"/> Spectacled Bear <input type="radio"/> Sun Bear <input type="radio"/> Grizzly Bear	<input type="radio"/> Polar Bear <input type="radio"/> Brown Bear <input type="radio"/> Black Bear <input type="radio"/> Atlas Bear <input type="radio"/> Bergman's Bear	<input type="radio"/> Polar Bear <input type="radio"/> Brown Bear <input type="radio"/> Black Bear <input type="radio"/> Other

Figure 8.4: Simple example of Hick's Law

Appendix B: Participant Observation Questions

Screens

1. Do you think the number of screens is too many or too few? If there are too many say which ones you would prefer gone. If there are too few say what you would add.

Screen Real Estate

2. Do you think the position of the buttons and icons could be improved? If so, how?
3. Do you like the size of the buttons and icons? If not, say whether they should be increased or decreased.
4. Would you prefer a different screen location for the titles? If so, where?

Text

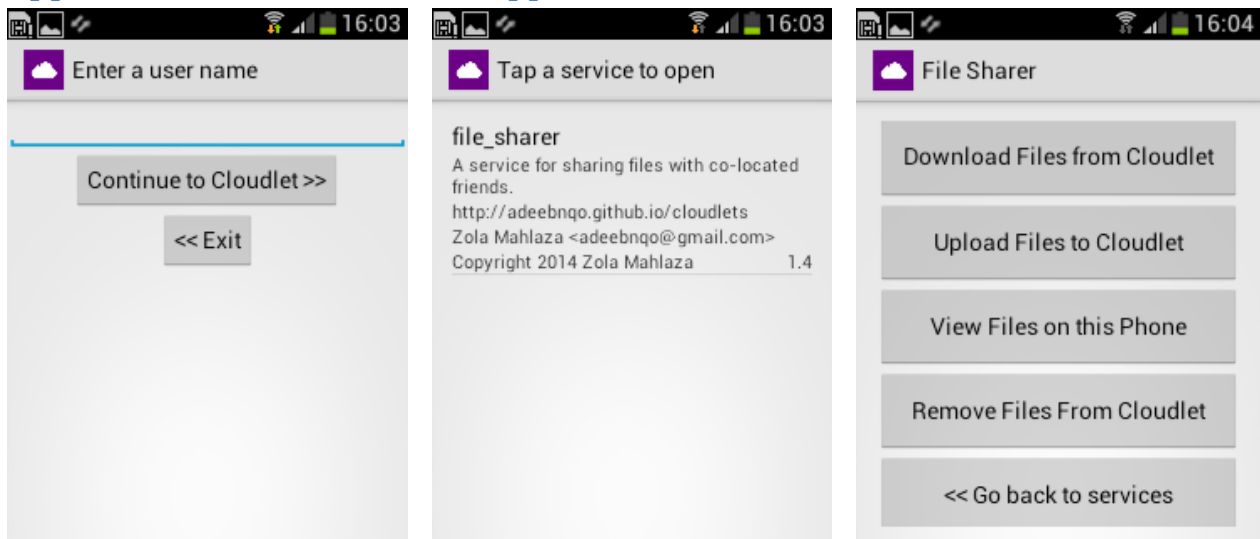
5. Do you like the text as capital letters or do you prefer it to be small letters.
6. Would you prefer the text to be bold/italicised (slanted) etc.?
7. Would you prefer different bullet points or discard them totally?
8. Do you like the icons or should they be coupled with text to make them easier to understand?

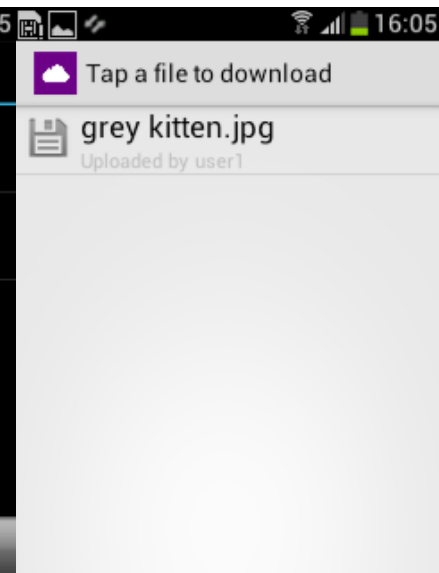
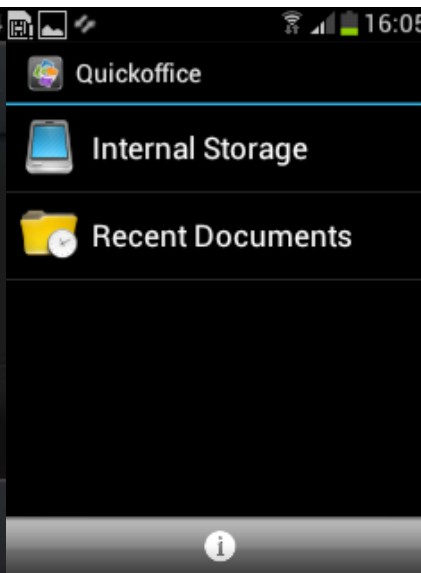
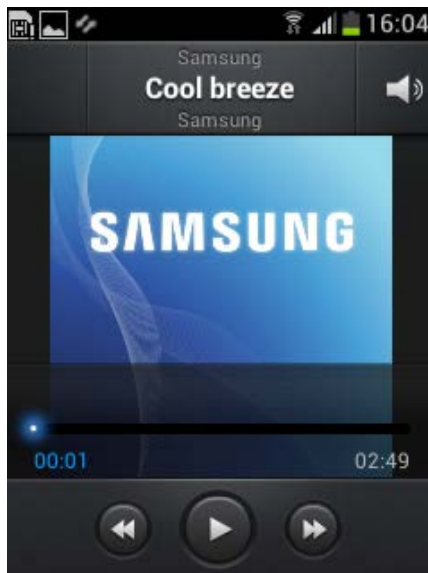
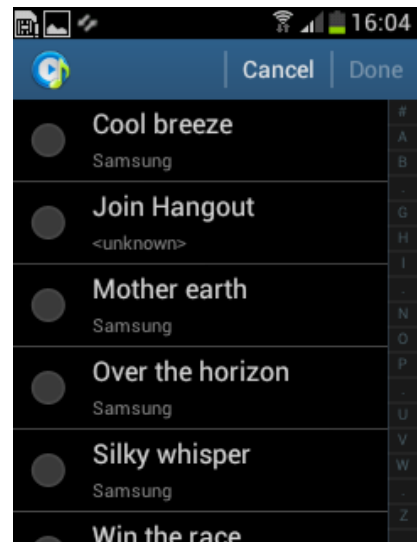
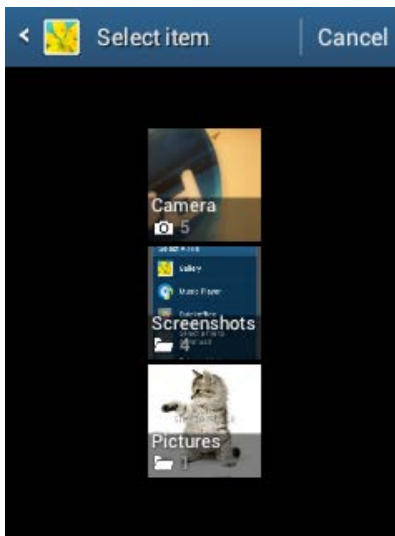
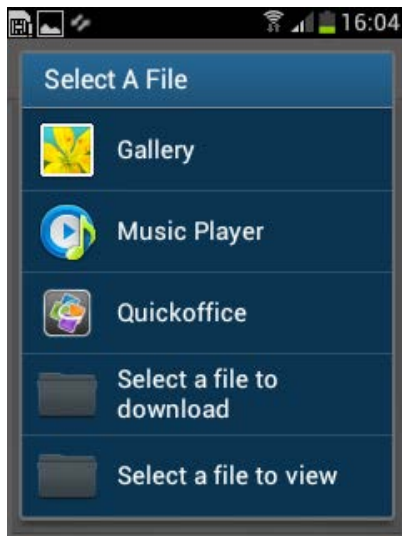
Interface Type

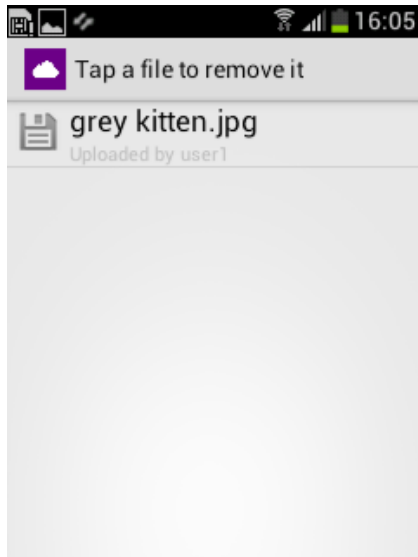
9. Rate the user-based interface from 1-5. 1 Being terrible and 5 being excellent.
10. Rate the service-based interface from 1-5. 1 Being terrible and 5 being excellent.

Which do you prefer the user-based or the service-based interface? Why?

Appendix C: Cloudlet Android Application Screens







Appendix D: Consent Form

User Testing Consent Form

To assess the quality of the system, the researchers require users to interact and carry out a few actions that would be expected to be commonly done on the cloudlet's file sharing services. Users will be asked to carry out three different tasks namely, *uploading, downloading and deleting* of shared files from the cloudlet. This will allow the researchers to test the effectiveness of the interface and the common sandbox of the cloudlet.

Note: your identity will remain anonymous throughout the research process and will not be disclosed in the report of the tests.

I understand the purpose of the tests and give my permission for any results obtained to be used in the report for the Cloudlet Honours project.

Participant's Signature:.....

Date:...../10/2014

Appendix E: Usability Test Tasks

Cloudlet Android App User Testing

Please complete the following tasks before filling in the questionnaire at the end. Please fill in the task completion evaluations after each task.

Actions in **bold** require interaction with the researcher.

Actions in *italics* require text input to the app.

Task 1: Sharing a file (upload)

1. **The researcher will give you a phone** with the Cloudlet app opened.
2. Log in with any 5 letter user name e.g. *user1*
3. Open the file sharing service
4. Try to download a file:
 - a. There should be no files on the cloudlet
5. Upload a picture of a grey kitten (it is located in the phone's gallery)
6. Exit the app.

Task 1 completion evaluation:

Please circle the correct answer.

I was able to log in	YES	NO
I was able to open the file sharing service	YES	NO
I was able to check whether there were files on the cloudlet	YES	NO
I was able to upload the picture	YES	NO
I was able to exit the app	YES	NO

Task 2: Downloading a file

1. **The researcher will give you another phone** with the Cloudlet app opened.
2. Log in with any 5 letter user name NOT the same as the previous task e.g. *user2*
3. Open the file sharing service
4. Check to see if there are any pictures in the gallery
5. Download the picture of a grey kitten uploaded by your first user name
6. Check to see if the file is in the gallery
7. Exit the app.

Task 2 completion evaluation:

Please circle the correct answer.

I was able to log in	YES	NO
I was able to open the file sharing service	YES	NO
I was able to check whether there were files on the cloudlet	YES	NO
I was able to download the picture	YES	NO
I was able to open the gallery and find the picture	YES	NO
I was able to exit the app	YES	NO

Task 3: Removing a file from cloudlet

1. **The researcher will give you another phone** with the Cloudlet app opened.
2. Log in with a username that is DIFFERENT FROM THE LAST 2 TASKS e.g. *user3*.

3. Open the file sharing service.
4. Try to remove the picture that was uploaded from task 1 from the cloudlet.
 - a. This shouldn't work because this user didn't upload it.
5. Exit the app.
6. Open the Cloudlet app.
7. Log in with the USERNAME YOU USED IN TASK 1 e.g. *user1*.
8. Open the file sharing service.
9. Remove the file from the cloudlet.
10. Try to download the removed file.
 - a. There should be no files on the cloudlet.
11. Exit the app.

Task 3 completion evaluation:

Please circle the correct answer.

I was able to log in with the user name from task 1	YES	NO
I was able to open the file sharing service	YES	NO
I was able to remove the file from the cloudlet	YES	NO
I was able to exit the app	YES	NO

Appendix F: Usability Test Questionnaire

Age: _____

Gender: _____ male

_____ female

PART 1: System Experience

1.1 How long have you worked on this system?

- | | |
|--|---|
| <input type="checkbox"/> less than 1 hour | <input type="checkbox"/> 6 months to less than 1 year |
| <input type="checkbox"/> 1 hour to less than 1 day | <input type="checkbox"/> 1 year to less than 2 years |
| <input type="checkbox"/> 1 day to less than 1 week | <input type="checkbox"/> 2 years to less than 3 years |
| <input type="checkbox"/> 1 week to less than 1 month | <input type="checkbox"/> 3 years or more |
| <input type="checkbox"/> 1 month to less than 6 months | |

PART 2: Past Experience

2.1 How many operating systems have you worked with?

- | | |
|-------------------------------|--------------------------------------|
| <input type="checkbox"/> none | <input type="checkbox"/> 3-4 |
| <input type="checkbox"/> 1 | <input type="checkbox"/> 5-6 |
| <input type="checkbox"/> 2 | <input type="checkbox"/> more than 6 |

2.2 Of the following devices, software, and systems, check those that you have personally used and are familiar with:

- | | |
|---|--|
| <input type="checkbox"/> computer terminal | <input type="checkbox"/> word processor |
| <input type="checkbox"/> personal computer | <input type="checkbox"/> graphics software |
| <input type="checkbox"/> lap top computer | <input type="checkbox"/> spreadsheet software |
| <input type="checkbox"/> color monitor | <input type="checkbox"/> database software |
| <input type="checkbox"/> touch screen | <input type="checkbox"/> computer games |
| <input type="checkbox"/> floppy drive | <input type="checkbox"/> voice recognition |
| <input type="checkbox"/> CD-ROM drive | <input type="checkbox"/> video editing systems |
| <input type="checkbox"/> keyboard | <input type="checkbox"/> CAD computer aided design |
| <input type="checkbox"/> mouse | <input type="checkbox"/> rapid prototyping systems |
| <input type="checkbox"/> track ball | <input type="checkbox"/> e-mail |
| <input type="checkbox"/> joy stick | <input type="checkbox"/> internet |
| <input type="checkbox"/> pen based computing | |
| <input type="checkbox"/> graphics tablet | |
| <input type="checkbox"/> head mounted display | |
| <input type="checkbox"/> modems | |
| <input type="checkbox"/> scanners | |

PART 3: Overall User Reactions

Please circle the numbers which most appropriately reflect your impressions about using this computer system.

Not Applicable = NA.

3.1	Overall reactions to the system:	terrible	wonderful	
		1 2 3 4 5 6 7 8 9		NA
3.2		frustrating	satisfying	
		1 2 3 4 5 6 7 8 9		NA
3.3		dull	stimulating	
		1 2 3 4 5 6 7 8 9		NA
3.4		difficult	easy	
		1 2 3 4 5 6 7 8 9		NA
3.5		inadequate power	adequate power	
		1 2 3 4 5 6 7 8 9		NA
3.6		rigid	flexible	
		1 2 3 4 5 6 7 8 9		NA

PART 4: Screen

4.1	Characters on the computer screen	hard to read	easy to read	
		1 2 3 4 5 6 7 8 9		NA
4.1.1	Image of characters	fuzzy	sharp	
		1 2 3 4 5 6 7 8 9		NA
4.1.2	Character shapes (fonts)	barely legible	very legible	
		1 2 3 4 5 6 7 8 9		NA

4.2	Highlighting on the screen	unhelpful	helpful	
		1 2 3 4 5 6 7 8 9		NA
4.2.1	Use of reverse video	unhelpful	helpful	
		1 2 3 4 5 6 7 8 9		NA
4.2.2	Use of blinking	unhelpful	helpful	
		1 2 3 4 5 6 7 8 9		NA
4.2.3	Use of bolding	unhelpful	helpful	
		1 2 3 4 5 6 7 8 9		NA
4.3	Screen layouts were helpful	never	always	
		1 2 3 4 5 6 7 8 9		NA
4.3.1	Amount of information that can be displayed on screen	inadequate	adequate	
		1 2 3 4 5 6 7 8 9		NA
4.3.2	Arrangement of information on screen	illogical	logical	
		1 2 3 4 5 6 7 8 9		NA
4.4	Sequence of screens	confusing	clear	
		1 2 3 4 5 6 7 8 9		NA
4.4.1	Next screen in a sequence	unpredictable	predictable	
		1 2 3 4 5 6 7 8 9		NA
4.4.2	Going back to the previous screen	impossible	easy	
		1 2 3 4 5 6 7 8 9		NA
4.4.3	Progression of work related tasks	confusing	clearly marked	
		1 2 3 4 5 6 7 8 9		NA

Please write your comments about the screens here:

PART 5: Terminology and System Information

5.1	Use of terminology throughout system	inconsistent	consistent	
		1 2 3 4 5 6 7 8 9		NA
5.1.2	Work related terminology	inconsistent	consistent	
		1 2 3 4 5 6 7 8 9		NA
5.1.3	Computer terminology	inconsistent	consistent	
		1 2 3 4 5 6 7 8 9		NA
5.2	Terminology relates well to the work you are doing?	always	never	
		1 2 3 4 5 6 7 8 9		NA
5.2.1	Computer terminology is used	too frequently	appropriately	
		1 2 3 4 5 6 7 8 9		NA
5.2.2	Terminology on the screen	ambiguous	precise	
		1 2 3 4 5 6 7 8 9		NA
5.3	Messages which appear on screen	inconsistent	consistent	
		1 2 3 4 5 6 7 8 9		NA
5.3.1	Position of instructions on the screen	inconsistent	Consistent	
		1 2 3 4 5 6 7 8 9		NA

5.4	Messages which appear on screen	confusing	clear	1 2 3 4 5 6 7 8 9	NA
5.4.1	Instructions for commands or functions	confusing	clear	1 2 3 4 5 6 7 8 9	NA
5.4.2	Instructions for correcting errors	confusing	clear	1 2 3 4 5 6 7 8 9	NA
5.5	Computer keeps you informed about what it is doing	never	always	1 2 3 4 5 6 7 8 9	NA
5.5.1	Animated cursors keep you informed	never	always	1 2 3 4 5 6 7 8 9	NA
5.5.2	Performing an operation leads to a predictable result	never	always	1 2 3 4 5 6 7 8 9	NA
5.5.3	Controlling amount of feedback	impossible	easy	1 2 3 4 5 6 7 8 9	NA
5.5.4	Length of delay between operation	unacceptable	acceptable	1 2 3 4 5 6 7 8 9	NA
5.6	Error messages	unhelpful	helpful	1 2 3 4 5 6 7 8 9	NA
5.6.1	Error messages clarify the problem	never	always		

		1	2	3	4	5	6	7	8	9	NA
5.6.2	Phrasing of error messages	unpleasant		pleasant							
		1	2	3	4	5	6	7	8	9	NA

Please write your comments about terminology and system information here:

PART 6: Learning

6.1	Learning to operate the system	difficult	easy								NA
		1	2	3	4	5	6	7	8	9	
6.1.1	Getting started	difficult	easy								NA
		1	2	3	4	5	6	7	8	9	
6.1.2	Learning advanced features	difficult	easy								NA
		1	2	3	4	5	6	7	8	9	
6.1.3	Time to learn to use the system	slow	fast								NA
		1	2	3	4	5	6	7	8	9	
6.2	Exploration of features by trial and error	discouraging	encouraging								NA
		1	2	3	4	5	6	7	8	9	
6.2.1	Exploration of features	risky	safe								NA
		1	2	3	4	5	6	7	8	9	
6.2.2	Discovering new features	difficult	easy								

		1 2 3 4 5 6 7 8 9	NA
6.3	Remembering names and use of commands	difficult easy	
		1 2 3 4 5 6 7 8 9	NA
6.3.1	Remembering specific rules about entering commands	difficult easy	
		1 2 3 4 5 6 7 8 9	NA
6.4	Tasks can be performed in a straight-forward manner	never always	
		1 2 3 4 5 6 7 8 9	NA
6.4.1	Number of steps per task	too many just right	
		1 2 3 4 5 6 7 8 9	NA
6.4.2	Steps to complete a task follow a logical sequence	never always	
		1 2 3 4 5 6 7 8 9	NA
6.4.3	Feedback on the completion of steps	unclear clear	
		1 2 3 4 5 6 7 8 9	NA

Please write your comments about learning here:

PART 7: System Capabilities

7.1	System speed	too slow fast enough	
-----	--------------	----------------------	--

		1 2 3 4 5 6 7 8 9	NA
7.1.1	Response time for most operations	too slow fast enough	
		1 2 3 4 5 6 7 8 9	NA
7.1.2	Rate information is displayed	too slow fast enough	
		1 2 3 4 5 6 7 8 9	NA
7.2	The system is reliable	never always	
		1 2 3 4 5 6 7 8 9	NA
7.2.1	Operations are	undependable dependable	
		1 2 3 4 5 6 7 8 9	NA
7.2.2	System failures occur	frequently seldom	
		1 2 3 4 5 6 7 8 9	NA
7.2.3	System warns you about potential problems	never always	
		1 2 3 4 5 6 7 8 9	NA
7.3	System tends to be	noisy quiet	
		1 2 3 4 5 6 7 8 9	NA
7.3.1	Mechanical devices such as fans, disks, and printers	noisy quiet	
		1 2 3 4 5 6 7 8 9	NA
7.3.2	Computer generated sounds are	annoying pleasant	
		1 2 3 4 5 6 7 8 9	NA
7.4	Correcting your mistakes	difficult easy	
		1 2 3 4 5 6 7 8 9	NA
7.4.1	Correcting typos	complex simple	

		1	2	3	4	5	6	7	8	9	NA
7.4.2	Ability to undo operations	inadequate							adequate		
		1	2	3	4	5	6	7	8	9	NA
7.5	Ease of operation depends on your	never							always		
	level of experience	1	2	3	4	5	6	7	8	9	NA
7.5.1	You can accomplish tasks knowing										
	only a few commands	with difficulty							easily		
		1	2	3	4	5	6	7	8	9	NA
7.5.2	You can use features/shortcuts	with difficulty							easily		
		1	2	3	4	5	6	7	8	9	NA

Please write your comments about system capabilities here:

PART 8: Multimedia

10.1	Quality of still pictures/photographs	bad							good		
		1	2	3	4	5	6	7	8	9	NA
10.1.1	Pictures/Photos	fuzzy							clear		
		1	2	3	4	5	6	7	8	9	NA
10.1.2	Picture/Photo brightness	dim							bright		
		1	2	3	4	5	6	7	8	9	NA
10.2	Colors used are	unnatural							natural		

		1	2	3	4	5	6	7	8	9	NA
10.4.1	Amount of colors available	inadequate		adequate							NA
		1	2	3	4	5	6	7	8	9	NA

Please write your comments about multimedia here:

Appendix G: Paper Prototype Participant Observations

