ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

SCHOOL OF ENGINEERING

DEPARTMENT OF COMPUTING

A MASTER'S THESIS

ON

Cloud-Based Mobile Data Backup and Recovery Technique Using the Ethiopian-Telecommunication GSM Network

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE

OF

MASTER OF SCIENCE IN SOFTWARE ENGINEERING

BY:

Kelele Gebremedhin Berhie

GSR/1315/08

Advisor:

Dr. Mesfin Abebe Haile

May 31, 2018

Ownership Declaration

I, the undersigned, declare that this thesis is my original work has not been presented for a degree or in any other work and that all sources of materials used for the thesis have been respectfully acknowledged.

Declared by:

Name: Kelele Gebremedhin Berhie

Signature: _____

Date: May 31, 2018

This thesis has been submitted for examination with my approval as university advisor.

Confirmed by advisor:

Name: Mesfin Abebe Haile(PhD)

Signature: _____

Date: May 31, 2018

ASTU, May 2018

Approval of Board of Examiners

We, the undersigned, members of the board of examiners of the final open defense by _____ have read and evaluated his/her thesis entitled _____ and examined the candidate. This is, therefore, to certify that the thesis has be accepted in partial fulfilment of the final requirement of the Degree of Masters of Science in software engineering.

Advisor's Name	Signature	Date
Internal Examiner's Name	Signature	Date
External Examiner's Name	Signature	Date
Chairperson's Name	Signature	Date
Department's Head Name	Signature	Date

Acknowledgment

I would first like to thank my thesis advisor **Dr. Mesfin Abebe** for his patience, motivation, interest, immense knowledge in the subject area and the deft ways in which his lovingly challenge and support throughout the whole of this work knowing when to push and to let up. His guidance helped me in all the time of research and writing of this thesis. It was a pleasure working with him. Thank you.

I would also like to thank the participants who were involved in the validation survey for this study. Their passionate participation and input, the validation survey have been successfully conducted. Thank you.

I would also like to acknowledge to **Perago Information Systems plc** owner, **Mr. Ewentu Abera** for kindly providing me with a working environment and internet service. Thank you.

Finally, I must express gratitude to my friend **Miskir Adane** for providing me a Microsoft Azure Storage Account for integration testing of my solution with the web service, Microsoft Azure storage. Thank you.

Table of Contents

List of Figures

List of Tables

Acronyms and Abbreviations

Amazon S3	Amazon Simple Storage Service
API	Application Programming interface
AWS	Amazon Web Service
AWS SDK	Amazon Web Service Software Development Kit
BAAS	Backup as a Service
Blob	Binary Large Object
BR	Backup and Recovery
CDMA	Code Division Multiple Access
DAAS	Database as a Service
Ethio-Telecom	Ethiopian Telecommunication
GSM	Global Systems for Mobile Communications
GT-730	Huawei Mobile Phone, Model 730
HTTP	Hyper Text Transfer Protocol
IAAS	Infrastructure as a Service
IoT	Internet of Things
IT	Information Technology
J2ME	Java 2 Micro Edition
JSR	Java Specification Request
KJVM	Kilobytes Java Virtual Machines
LPMS	Local Partial Mirror Server

MIDP	Mobile Information Device Profile
NIST	National Institute of Science and Technology
PAAS	Platform as a Service
PDA	Personal Digital Assistant
PIM	Personal Information Management
P2P BR	Peer to Peer Backup and Recovery
SAAS	Software as a Service
SDK	Software Development Kit
SIM	Subscriber Identity Module
SMS	Short Message Service
SQS	Simple Queue Service
StAAS	Storage as a Service
Sync	Synchronize
S3CMD	Simple Storage Service Command
TAAS	Test as a Service
TDMA	Time Division Multiple Access
TCP/IP	Transmission Control Protocol/Internet Protocol
UAAS	Utility as a Service
UFPS	User's Feature Phone as Storage
UI	User Interface
USB	Universal Serial Bus

WWW	World Wide Web
2G Network	Second Generation Network

Definition of Key Terms, Concepts, and Variables

App	A short-term defined by Apple for specifying Application.
Bandwidth	Data transfer capacity.
Compression	The process of converting actual characters into other characters, so that more characters will get in the same storage space.
Data Types	A term used for naming this solution. App name for a smartphone, and J2ME. This name is a combination of English, and Amharic to refer my data.
DatayeN	A term used for naming this solution. App name for a smartphone, and J2ME. This name is a combination of English, and Amharic to refer my data.
Decompression	The process of recovering compressed content into actual content.
Decryption	The process of recovering encrypted content into actual content.
De-Duplication	It is a process of reducing duplicated data to utilize computer memory efficiently.
Encryption	The process of converting actual characters into other characters, so that the actual content information will not be understandable by unauthorized users.
Feature Phones	Mobile phones which have less capability, like less processing capability, small memory capacity, a phone that cannot be connected.
GSM Network	A reliable network of cellular devices which enables to communicate with mobile users with each other.
Huffman Coding	It is a lossless compression algorithm based on the frequency of occurrence of the data item. The principle is to use a lower number of bits to encode the data that occurs more frequently.
Internet	International network to enable communication around the globe.
Java SE	Stands for Java standard edition and is normally for developing desktop applications, forms the core/base API

Java ME	Stands for Java micro edition for applications which run on resource-constrained devices (small-scale devices) like cell phones, suitable for very low capacity processors, and memory.
Library	Predefined component, dependency files.
MAS API	Microsoft Azure Storage Application Programming Interface, an endpoint to enable access the Microsoft Azure storage service by clients.
MAS SDK	Microsoft Azure Storage web service SDK, a library which exposes Microsoft Azure Storage web services API's interfaces.
Memory	This term refers to the hardware device of mobile, it measures the capacity of storing data ready for calculation.
Processor	This term refers to the hardware device of mobile, it measures performance, the speed of performing calculations in a specific time.
P2P BR	Peer to peer backup and recovery option for feature phones. This feature is only available in feature phones, which enables clients to store data in peer device as a backup, and can be recovered when needed.
Small Device	Devices which have small capacity in terms of processing performance, storage capacity, and handheld.
Smartphone	A mobile device which has higher and better capability, performance, support Wi-Fi, and memory than the feature phones.
SMS Characters	Characters allowed in short message service at a time.
Thick Cloud	Represents cloud storage server which has extended functions for the purpose of server efficiency.
Thin Cloud	Represents cloud storage server which has limited functions to reduce the load on the server.
Web Services	Client and server applications that communicate over the World Wide Web's (WWW) Hypertext Transfer Protocol (HTTP).

Abstract

Nowadays the growth of IT infrastructures, desktop computers, laptop, tablets, and mobile phones makes technology users dependent on the digital information. Mobile phones, like feature phones, and smartphones main aim is to enable individual users to communicate with one other based on their available contacts addresses, text-messaging and maintain personal information. Feature phones are mobile phones with less capability in all angles, whereas smartphones have the better capability; both enable efficient communication whether we use GSM or CDMA network. As mobile devices are prone to error, easy for hardware damage, or are exposed to theft, the data resides on the devices must be backed up so that it can be recovered any time when it is needed.

The main objective of this study is to provide Backup as a Service solution of a complete backup and recovery service to both feature phones and smartphones using the emerging cloud-based technology and a web service, Microsoft Azure Storage, in which it addresses the problem of permanent data loss and auto-data portability in different mobile phone platforms, and designing a framework for backup and recovery service that supports both feature and smart mobile phone.

This study involves by creating java application for feature phones using J2ME technology, and Android application which enables smartphone mobile users to save their data on the remote server on Microsoft Azure Storage, and recovers it when they need, peer storage and recovery service for feature phones, using P2P BR feature phone app feature and a de-duplication component to prevent unwanted duplicated data during taking backup is also incorporated. Generally, this backup and recovery solution is intended for synchronizing feature phones and smartphones through the cloud storage service.

Keywords: - Software as a Service; De-Duplication; Backup and Recovery; P2P BR; Cloud Storage; Feature Phones; SMS; Smartphones.

Chapter 1 : INTRODUCTION

1.1.Background

Cloud computing is a recent and dominant computing paradigm which provides different services, such as software as a service (SAAS), platform as a service (PAAS), and infrastructure as a service **(**IAAS**)** owned by cloud service providers to clients. Although the above-listed services are a generalized cloud computing service stack, cloud computing constitutes another more service, like database as a service (DAAS), test as a service (TAAS), utility as a service (UAAS), backup as a service (BAAS) and many more services are existing services in the cloud. It has become increasingly popular to talk of "cloud computing" as the next infrastructure for hosting data and deploying software and services [1].

Those above-listed cloud computing services are available everywhere to the customers through the internet, and payment for that service is based on per-use and On-demand self-service. Cloud is also elastic, implies that customers of the cloud can drop or add service(s) what service they want, and when they want; where ever they are.

According to National Institute of Science and Technology (NIST), cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources. NIST also, defines some basic cloud computing characteristics, services delivery model and deployment models as its shown in Figure 1.1.

Backup as a Service (BAAS) is cloud computing services stacks which provides a means of backup and restore service for data found in electronic devices, such as computer desktop, laptop, company data found in computers, smartphones, and tablets using the internet. Examples of cloud computing Backup as a Service provider are Mozy, Carbonite, Symantec's Protection Network, implements integrated solutions that include backup-specific software hosted on both the client and at the data center [1]. Those cloud backup and recovery service providers adopt thick cloud, as they provide application software for compression and cleaning unnecessary data on the cloud server machine, which will lead client vendor lock-in, which is the main challenge of cloud computing service providers.

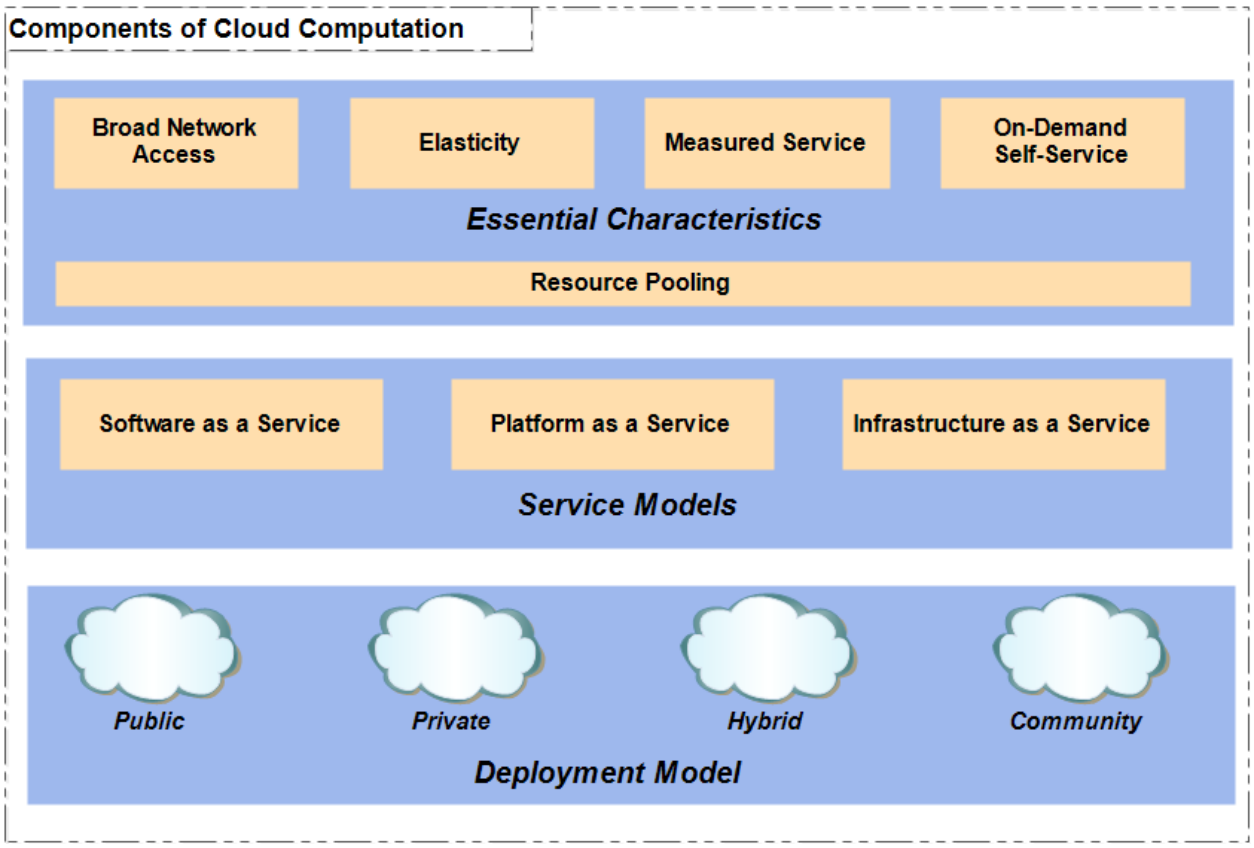

Figure 1-1: Components of Cloud Computation

In this study**,** we are focused on providing Backup as a Service (BAAS) for customers' mobile device data using telecommunication GSM mobile network and the Internet in Ethiopia. These days' mobile devices, smartphones, tablets, laptops are the dominant devices that can store users' most valuable, data. Information is power, which is the base to control and run company business, to control evidence, to make a decision, to strength relationship, and many others. These days, information is stored in electronics (laptops, tablets, smartphones and many more). The longevity of the information(data) is until the hardware is physically safe, memory is not damaged or not lost, or out of the hand from the owner. Therefore backup is a crucial task since hardware faults, software or human errors can lead to the loss of important information. In addition to faults, backups are even more important for devices such as laptops and smartphones, since they are more prone to lose or to theft [2]. But what if the electronics is lost, or if the memory storage is damaged? what happens to the data in the device? will the owner of the mobile device get back his/her data?

1.2.Motivation

In this section, we have discussed why we choose to study on mobile data backup is that we believe data is the most important thing in a world of information technology. We have got many motivations which makes us decide to work on mobile data backup and recovery solution.

1.2.1.Observation

In this section, we discussed the three most important occasions(cases) including challenges and drawbacks of telecommunication companies.

- ***Case 1: A friend of mine who had lost a Nokia mobile: -*** This friend of mine was a beginner at a new business company working on building house furniture. This guy was using the phone I described above for communicating with his customers, infrastructure providers, and other information. After he lost his mobile, he loses everything which was on the mobile phone, such as contacts, call logs, messages including account numbers, input specification for building the infrastructure stored as a to-do list. Then after, this guy, his furniture company productivity was declining because he loses infrastructure providers contact address, he was unable to find his customers and necessary messages and he was like starting over his business activity.
- ***Case 2: I lost my smartphone; Huawei with Model Number GT-7300: -*** Inside my smartphone, I had much information including about 123 contacts, many unknowns but call logs I need to have, many important messages, many selected research ideas, and titles stored as a to-do list. I had a backup on the cloud but the mobile I replaced with was unable to get me my data from the cloud because of its specification including it does not support internet connection and platform difference from the previous one.
- ***Case 3: Cloud adoption challenges of telecommunication: -*** Telecommunication companies has the drawbacks of adopting cloud services for their customers. Customers of telecommunication need advanced technology enhancement from the provider, but if the company fails to add advanced enhancement, customer satisfaction drops step by step and start to think to change the provider. In terms of our country, adopting cloud is seeming long but as they defined in their mission, adopting cloud computing is the institutional mission.

1.2.2.Preliminary Analysis

For the purpose of supporting our observation, we conducted a survey random sampled user, and the collected response is analyzed. We have targeted using purposive sampling around 100 participants by preparing a questioner which consists of 13 questions. The sample participants include BSc students, PG students, and Ethiopia Telecommunication corporation's employees, that work in the customer service agent division.

The following table shows the analyzed result of the collected survey taken from a target of 100 participants. The participants were asked to answer around 13 questions, such as what kind of data they store on their mobile phone, do they use cloud backup and recovery service, data lost impact ratio, data priority, and what do they think on using backup and recovery using both GSM network, and internet and other detail questions.

Table 1-1: Summary of collected Survey Response

Variables	*Participants (%)*
Android phone users	58
iPhone users	3
Windows phone users	3
Feature phone users	34
Backup and recovery service users(smartphones)	26
Users who don't use cloud	52
Users who recover lost data	16
Users' contact storing ratio	100
Users' message storing ratio	90
Users' photo storing ratio	90
Users' file storing ratio	83

Users' video storing ratio	75
Users' call logs storing ratio	67
Contacts lost impact ratio	86
Photo lost impact ratio	68
File lost impact ratio	50
Message lost impact ratio	27
Backup request to Ethio-Telecom	20
Internet backup and recovery service is useful	94
Backup and recovery service using GSM network is useful, and address all mobile phone users	94

Summarizing the survey, the participant's store contacts, messages, pictures, music, files and video on their mobile phone device. Based on the collected data, contact addresses are the most important information found on their devices, will be highly affected if they lose the contact addresses. Pictures and files also the most important information found their mobile next to contact addresses followed by messages. Regarding mobile usage and lost mobile replacing ratio, almost all participants use Android devices, and lost mobile replacing ratio by the Android device is almost 100%.

In the following page, a detail collected data analysis is represented, each question with answers represented in charts.

***Mobile usage device ratio:* -** the below diagram shows that mobile phone devices ratio used by the targeted sample participants of the interviewing process.

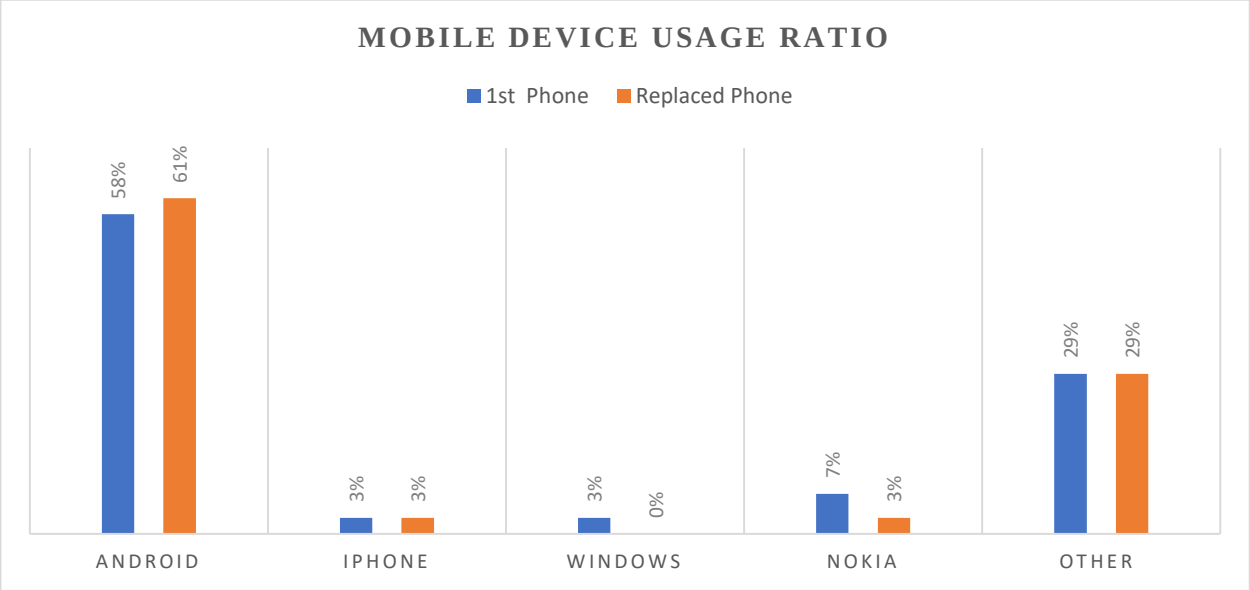

Figure 1-2: Mobile Device Usage Ratio

***Data type storing ratio:* -** the diagram below shows the kind of data stored in mobile devices by the participants.

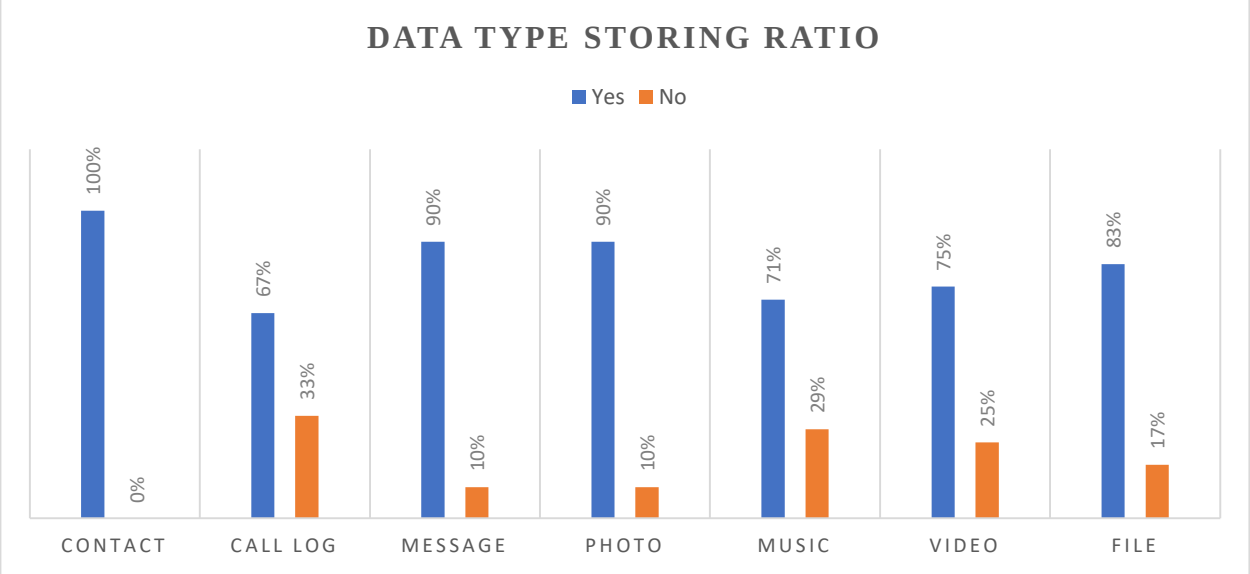

Figure 1-3: Data Type Storing Ratio

Important mobile data rank: **-** the diagram below shows the important data rank found on the participants' mobile devices. The participants were asked and given numbering options, starting from 1 to 7. The number representation is 1 stands for first important, 2 stands for second important, 3 stands for third important, 4 stands for fourth important, 5 stands for the fifth important, 6 stands for sixth important and 7 stands for seventh important.

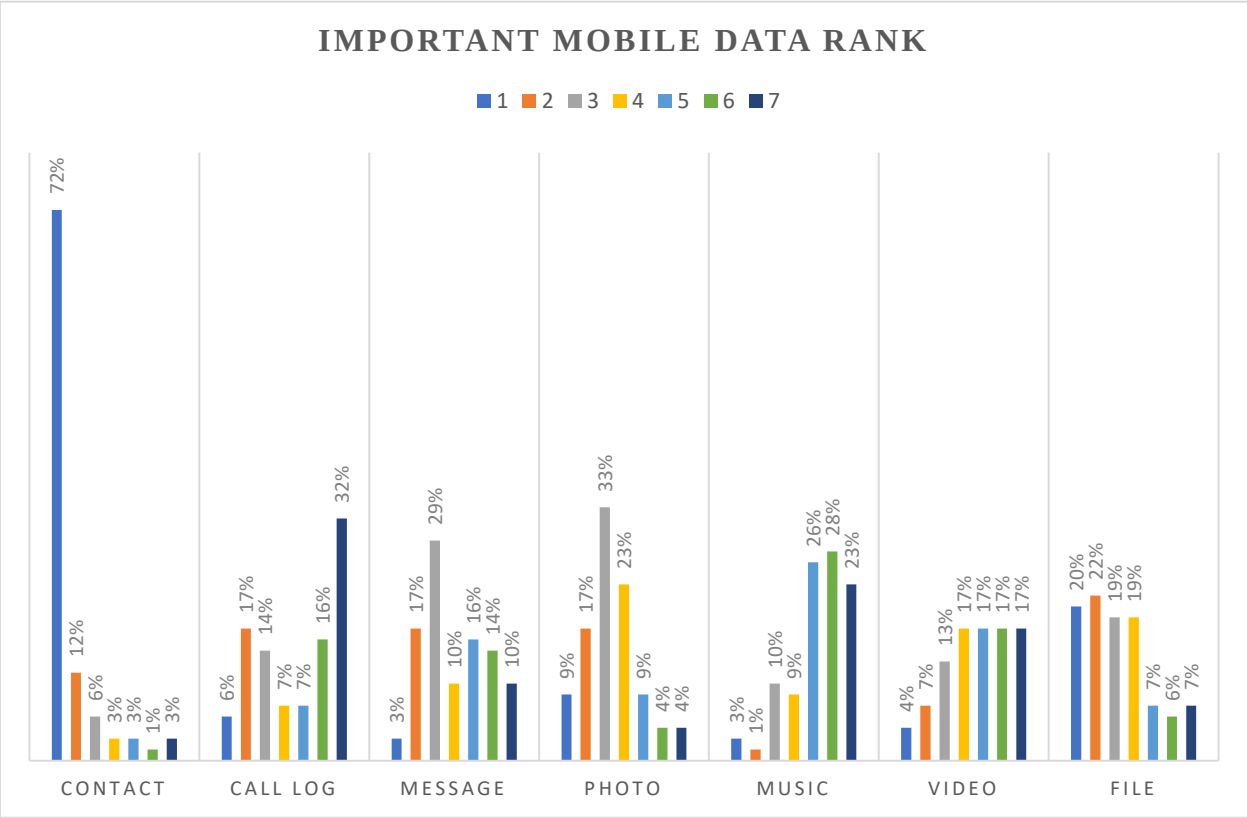

Figure 1-4: Important Mobile Data Rank:

To explain some of the information represented, 72% of the participants describe contacts found in their mobile device is the most and highly important digital information, 12% of the participants said that contacts are second most important, 1% of the participant said contact addresses are less important, 3% of them said contacts are least important data. Files are the second most valuable and important digital information found on their mobile devices. To represent the numbers, 20% of the participants described as the file is the most important data, 22% of the participants said files are their second most important data, 7% of the participant said the file is the least important data. Pictures are the third most important digital information found on their mobile devices, rated by the targeted participants.

Data lost impacts ratio: - the diagram below shows how the participants will get affected if they lost the digital information stored on their mobile devices. The impact measured is using three levels, by measuring high, medium, and low.

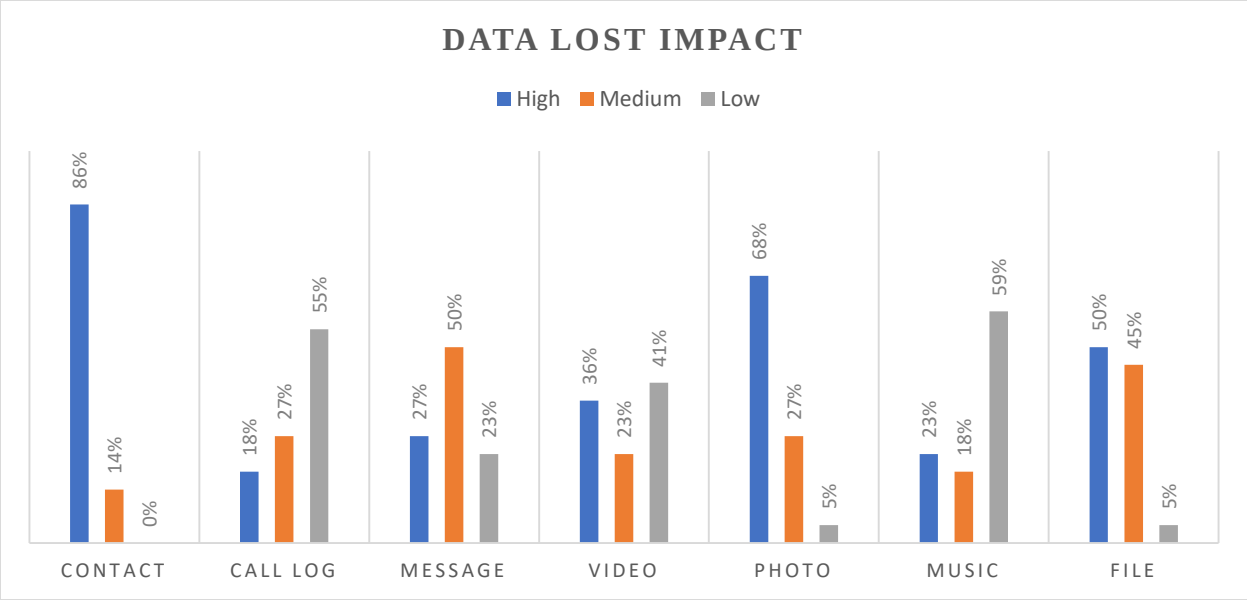

Figure 1-5: Data Lost Impact

Phone replacing & data recovery ratio: - this shows mobile lost, the ratio of backup and recovery service solution users, data recovery ratio and the ratio of data recovery request to Ethio-Telecom.

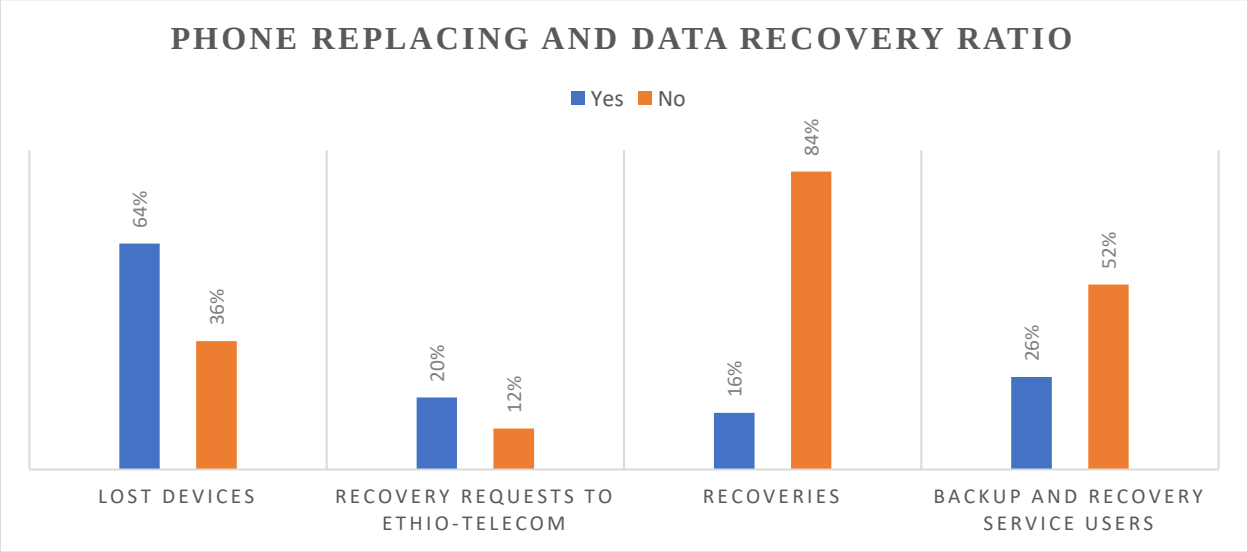

Figure 1-6: Phone Replacing and Data Recovery Ratio

1.3.Statement of the Problem

Backup is the process of copying files, documents, or any other data to a local drive or cloud storage remotely so that they will be restored in case of device failure, device theft or other problems. Cloud storage is an online storage service over the network where the data will be stored in the cloud. Cloud storage will help to offload the active data on-premise to cloud with pay-per-usage model [6].

Backup is the most important process when managing digital information, and it is believed that this service should be provided to mobile customers of Ethio-Telecom so that at the time of the device damage, or lost, the customers can get it back from the cloud storage server. Furthermore, when a mobile device is damaged or lost, the owner loses two things, the hardware and the software (information), but if the user was using backup service, he/she loses only the hardware component while the data is existing on the remote storage, in which the user will restore it, get it back.

These days cloud backup and recovery solutions are only focusing on smartphones while feature phones ignored. In this study, we are considering to implement backup service for Ethio-Telecom mobile network customers for all mobile devices with SIM card both using the internet and Ethio-Telecom GSM network. The critical question has been on how to provide backup service without using internet service but using Ethio-Telecom GSM network(2G) to Ethio-Telecom GSM mobile network users. In this backup and restoring service, our consideration is only on the data of the mobile device contacts, call logs, to-do list, events, and messages. And in addition to this we are considering to implement cloud computing service called BAAS, so that data portability in between different platforms. After this research problem is solved and provided for public usage; hopefully it helps the users to find and restore back lost data. Finally, data portability between different platforms is also supported. For instance: if a user was using a smartphone and using this backup service, and lost his/her phone, and buys one of the feature phones, it is possible for him/her to can recover the data that was backed up on the remote servers and vice versa (like contacts, messages).

1.4.Research Questions

The following basic research questions were used to guide the study, and they are answered in this study properly:

RQ1. What are the possible mechanisms for a feature phone to communicate with a web service, cloud storage, and enable those devices to store backup data in the cloud storage, recover the data back from the cloud storage?

RQ2. How to enable data portability among different mobile phones platform, especially in between feature phones and smartphones through the cloud?

RQ3. What are the possible compression algorithms that can increase the number of characters of that can be sent using GSM SMS service?

1.5.Objectives of the Study

The aim of this research is to provide a backup service for of Ethio-Telecom GSM network, mobile users, to maintain backup for their data on the remote server using both internet and GSM network and recover it back when they need it

1.5.1.General Objective

The primary objective of this study is to create backup and recovery service on cloud storage that can support different mobile phones such as feature phones and smartphones for Ethio-Telecom GSM network customers.

1.5.2.Specific Objectives

- To review the related literature on data backup and recovery, adopting cloud storage.
- To explore cloud computing adoption in Ethiopia.
- To study the best compression standards for backup and recovery.
- To develop a framework for backup and recovery service that supports both feature and smart mobile phone.
- To implement a prototype application and system for the backup service using Microsoft Azure cloud storage platform and web services.
- To evaluate the prototype app and system.

1.6.Scope and Limitation of the Study

1.6.1.Scope

This study focuses on designing and implementing a backup and recovery solution to Ethio-Telecom mobile users, to both feature phones and smartphones, using a web service Microsoft Azure Cloud Storage.

- Backup and restore mobile devices' contact address, call logs, events, to-do list, and messages using local GSM network for feature phones.
- Implement internet-based cloud backup as a service for Android mobile devices to maintain backup the data that users need.
- Synchronizing data using GSM network and Internet.
- Developing java app for those feature phones which support Java platform, and smartphones, especially Android devices.
- Implementing data portability among different mobile platforms.

This research will not address the following points:

- It will only focus on backup and recovery, not on solving the challenges of cloud computing.

1.6.2.Limitation

- Less GSM network bandwidth to transfer more data at a time.
- This solution will not take a backup of photos, music's, videos or files using the GSM network for feature phones.
- Data encryption and decryption will not get covered and will not get implemented in this solution because of the feature phones performance and capability, and it consumes bandwidth because encryption increases message length or that can be transmitted in GSM network.

1.7.The significance of the Study

Cloud computing has brought significant change in the information communication world, including storage utilization. This study contributes to the current information technology by enabling synchronization of the Internet and GSM infrastructure backup and recovery process.

And utilization of GSM network for the purpose of data backup and recovery of mobile users' contacts, call logs, and messages. And this research is significant to all company/people who use it. Like,

- ***Cloud Provider**:* - this company will benefit financially, because it provides and rents IT infrastructures storage devices, for storing users' data.
- ***Users:* -** users will benefit because their data will never be lost because of losing a mobile device, they get it back when they need it using whatever platform feature phone or android) and wherever they are.
- ***Ethio-Telecom:* -** this company will benefit financially, because its' clients will keep in touch as there will not be losing their contact address, and this solution also uses the GSM network while backing up and recover the data.
- ***Academic**:* - this research solution is opening a new insight into GSM network synchronizing with the cloud environment, putting foundation on how to synchronize, so it helps others to explore more on the subject.

1.8.Methodology

In this study, starting from the first phase, we have used the following different sequential steps and methods to design and implement the proposed solution; represented in the diagram below.

- ***Observations and Problem Identification**:* - in almost three years of observation, mobile users were having difficulty in data recovery of their lost device. From our observation, mobile users were changing their devices platform back and forth, sometimes they own a smartphone(android) and some other time owns feature phones either they lost their device or needed to change the device. In either way, user's data remains stacked either inside the lost device or the old phone.
- ***Data Collection and Analysis**:* - for the purpose of supporting our observation, we conducted a survey random sampled user, and the collected response is analyzed. We have targeted using purposive sampling around 100 participants by preparing a questioner which consists of 13 questions. The sample participants include BSc students, PG students, and Ethiopia Telecommunication corporation's employees, that work in the customer service agent division.

- ***Literature Review:*** - in addition to our observation, and to the data collection, we explored different related articles, journals, websites, and books.
- ***Tools, Environment Selection, and Specification:*** - software tools, development environment, compression algorithm, plugin libraries, cloud storage server determined and selected.
- ***Architecture Design:*** - in this step, we have designed the proposed architecture, its components and their interaction among them and the type of data to get a transfer, and the requirement modeling (use case model), and class diagram.
- ***Prototype Implementation:*** - in this step, we have developed the prototype system and application. This includes the deployment of the apps and systems on the target device and platform.
- ***Prototype Testing:*** - we have evaluated the developed app and its functions including its integration with cloud storage server.

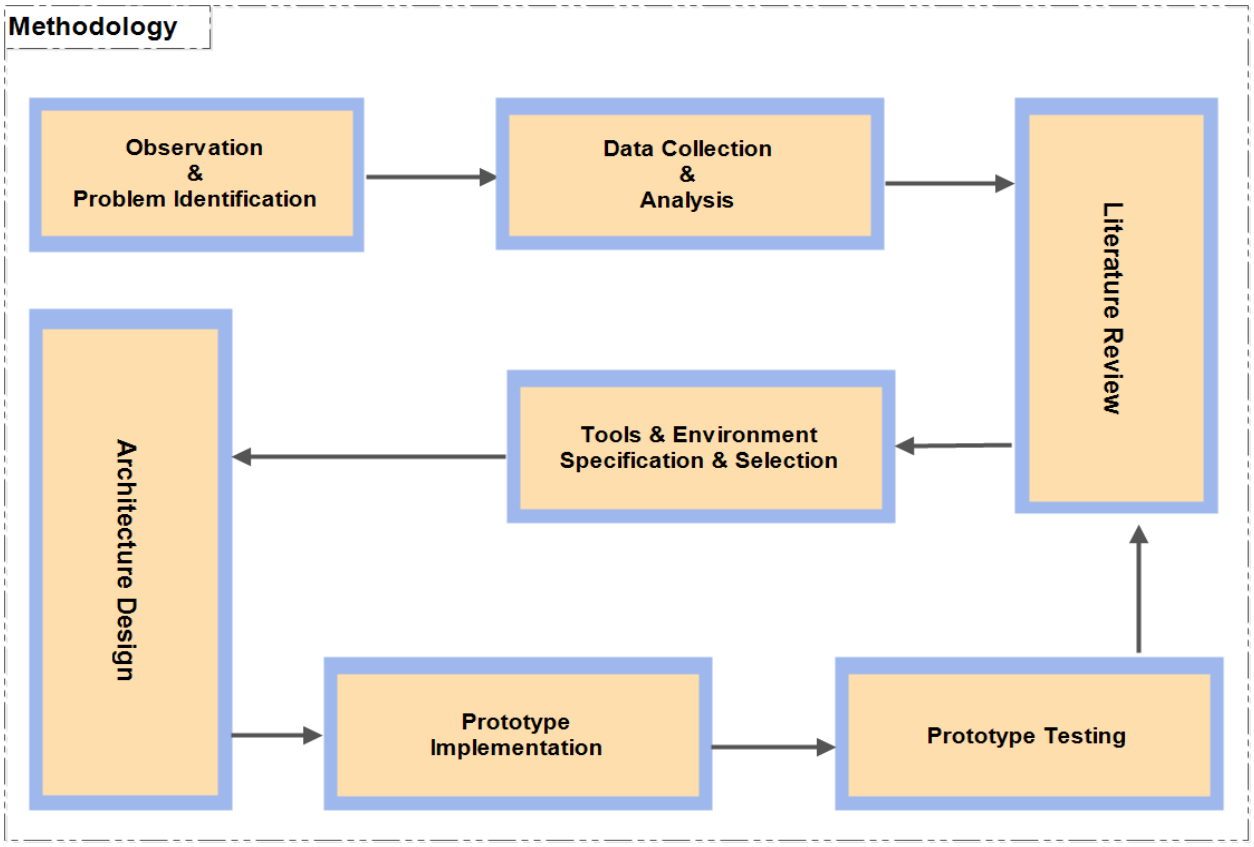

Figure 1-7: Methodology

1.9.Organization of the thesis

The rest of this paper is organized as follows:

- Chapter two, we have described the literature review and related work. In this part, we discussed the concepts cloud, cloud backup, and recovery, and GSM network and its applications, we have discussed three most dominant forces of IAAS providers and analyzed compression algorithms. We also explained and demonstrated the uncovered gaps in this research topic.

- Chapter three, we have discussed the methodology. In this part, we have explained the research design activities used for solving the proposed solution, the data collection techniques, sampling mechanism, and tools used.

- Chapter four, we have designed the proposed architecture of the system and described it in detail. In this part, we discussed system architecture, it's components, their interactions and the protocol they used. We have Also described each component in detail.

- Chapter five, the implementation of the system is described. In this part, we represented system modeling (use case model, class diagram, and deployment model), cloud storage interface, we have explained Microsoft Azure account storage structure and its workflow, used programming languages are explained, used algorithm are represented and explained, and development tools are explained.

- Chapter six, we have discussed this study result and discussion. In this part, we evaluated our work, represented collected data analysis and explained the analysis result, presented and explained this study solution usage scenario, and discussed and interpreted the result of this study.

- Chapter seven, we have discussed conclusions and our future work. In this part, we present our conclusion and set some future work and further insights.

Chapter 2 : LITERATURE REVIEW

2.1. Conceptual Review

2.1.1. GSM Network

Networking is the processes of communicating two or more electronic devices for the sake of sharing documents, files, and other resources. In today's world, there are different classes of networks, but according to this study, the network is classified into two, the first one is the internet, which stands for an international network for communication of any network supporting electronic devices with high bandwidth, and greater data transfer per time across the globe. The second one is a cellular network (GSM, or 2G network), GSM is a digital mobile telephony system that is widely used in the world. GSM uses a variation of time division multiple access (TDMA) and is the most widely used of the three digital wireless telephony technologies (TDMA, GSM, and CDMA). GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot [28]. It operates at either the 900 MHz or 1800 MHz frequency band. GSM network is used for communication of any mobile devices with SIM card within the specified region with low bandwidth, secured and low data transfer rate per time. Selecting one of the above two networks depends on the need of selector, but for this study, we prefer synchronizing and utilizing both networks for data backup and recovery of Ethio-Telecom mobile phone customers.

2.1.2. Data Backup and Recover

Backup and recovery refer to the process of backing up data in case of a data loss and setting up systems that allow data recovery due to data loss. Backing up data requires copying and archiving computer data so that it is accessible in case of data deletion or corruption. Data from an earlier time may only be recovered if it has been backed up. Because data is a powerful tool for enterprises and individuals, it is crucial to protect it. But for the preventing unfortunate and accidental losing data, users should talk safe backup and could recover it when needed. Backup is the process of storing a copy data into remote storage or local storage so that it could be recovered when needed [2].

2.1.3. Cloud Computing and Cloud Storage

Cloud computing [18] is the on-demand delivery of computing power, database storage, applications, and other IT resources through a cloud services platform via the internet with pay-as-you-go pricing. It provides a simple way to access servers, storage, databases and a broad set of application services over the Internet. Cloud computing has six basic advantages and benefits.

a) ***Trade capital expense for variable expense***

Instead of having to invest heavily in data centers and servers before you know how you're going to use them, you can only pay when you consume computing resources, and only pay for how much you consume.

b) ***Benefit from massive economies scale***

Achieve a lower variable cost than you can get on your own. Because usage from hundreds of thousands of customers is aggregated in the cloud, providers such as Amazon Web Services can achieve higher economies of scale which translates into lower pay as you go prices.

c) ***Stop guessing capacity***

Eliminates infrastructure capacity needs. When you make a capacity decision prior to deploying an application, you often either end up sitting on expensive idle resources or dealing with limited capacity. With cloud computing, these problems go away. You can access as much or as little as you need and scale up and down as required with only a few minutes' notices.

d) ***Increase speed and agility***

In a cloud computing environment, new IT resources are only ever a click away, which means you reduce the time it takes to make those resources available to your developers from weeks to just minutes. This results in a dramatic increase in agility for the organization since the cost and time it takes to experiment and develop is significantly lower.

e) ***Stop spending money on running and maintaining data centers***

Focus on projects that differentiate your business, not the infrastructure. Cloud computing lets you focus on your own customers, rather than on the heavy lifting of racking, stacking and powering servers.

f) Go global in minutes

Easily deploy your application in multiple regions around the world with just a few clicks. This means you can provide a lower latency and better experience for your customers simply and at minimal cost.

In [17], cloud computing provides multiple services, such as Software as a Service(SAAS), Storage as a Service(StAAS), Platform as a Service(PAAS), and Infrastructure as a Service. SAAS is probably the most popular form of cloud computing and the easiest to use. SAAS uses the Web to deliver applications that are managed by a third-party vendor and whose interface is accessed on the clients' side. Most SAAS applications can be run directly from a Web browser, without requiring any downloads or installations. SAAS eliminates the need to install and run applications on individual computers. With SaaS, it's easy for enterprises to streamline their maintenance and support, because everything can be managed by vendors: applications, runtime, data, middleware, O/S, virtualization, servers, storage, and networking. Examples of SAAS are Gmail, Google Apps, Microsoft Office 365, Google+, Facebook, Yahoo. Cloud vendors provide different services, such as Microsoft Azure storage, and Amazon S3, a storage over the internet that will get consumed by consumers of the cloud services. They are designed to make web-scale computing easier for developers. Microsoft Azure storage provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, secure, fast, inexpensive infrastructure that Microsoft uses to run its own global network of websites.

In [16], Data storage is one of the today's fundamental services with companies, universities and research centers having the need of storing large amounts of data every day. Cloud storage services are emerging as a strong alternative to local storage, allowing customers to save costs of buying and maintaining expensive hardware. Several solutions are available on the market, the most famous being Amazon S3, Amazon Glacier, Windows Microsoft Azure, and Rack Space Cloud Files.

In [21], mobile data services are most popular services used by customers. This service contains voice and SMS service, supported by the GSM network. This network provides reliable, secured and on time and Real-time communications. Using the GSM, mobile users can send SMS to

another mobile user(s) a maximum of 140 characters of eight bit. Though it has a low bandwidth its popularity shows the benefit of GSM network.

A data backup is a result of copying or archiving files and folders for the purpose of being able to restore them in case of data loss. Data loss can be caused by many things ranging from computer viruses to hardware failures to file corruption to fire, flood, or theft etc. [11]. Data backup and recovery can be performed at two different sites. The first one is local data backup and recovery, a data backup and recovery onto and from offline devices, and the second one is remote data backup and recovery, a data backup and recovery onto and from remote devices using the network. This study focuses on providing Software as a Service (SAAS), specifically Backup as a Service, in a cloud environment for data backup and recovery for both the internet and GSM network user devices using the emergent technology cloud computing storage. In today's world the importance of data or information increased, the need for information backup is highly important so that the information can be recovered when needed from the backup server. The following section, Literature Review shows the importance of data backup and recovery, the study that is done on backup and recovery service, and the best options to provide the required solution.

Software as a Service is an application running on the cloud infrastructure, by subscribing to the application, clients can use the service provided. Once the clients they subscribed, they can access it using different devices, and make user specific-application configuration. but users can't control and manage the underlying infrastructure. Cloud providers are the one who is responsible for installing, configure, maintain and manage the overall cloud infrastructure including the network, server, operation system, storage, virtual machine and the application.

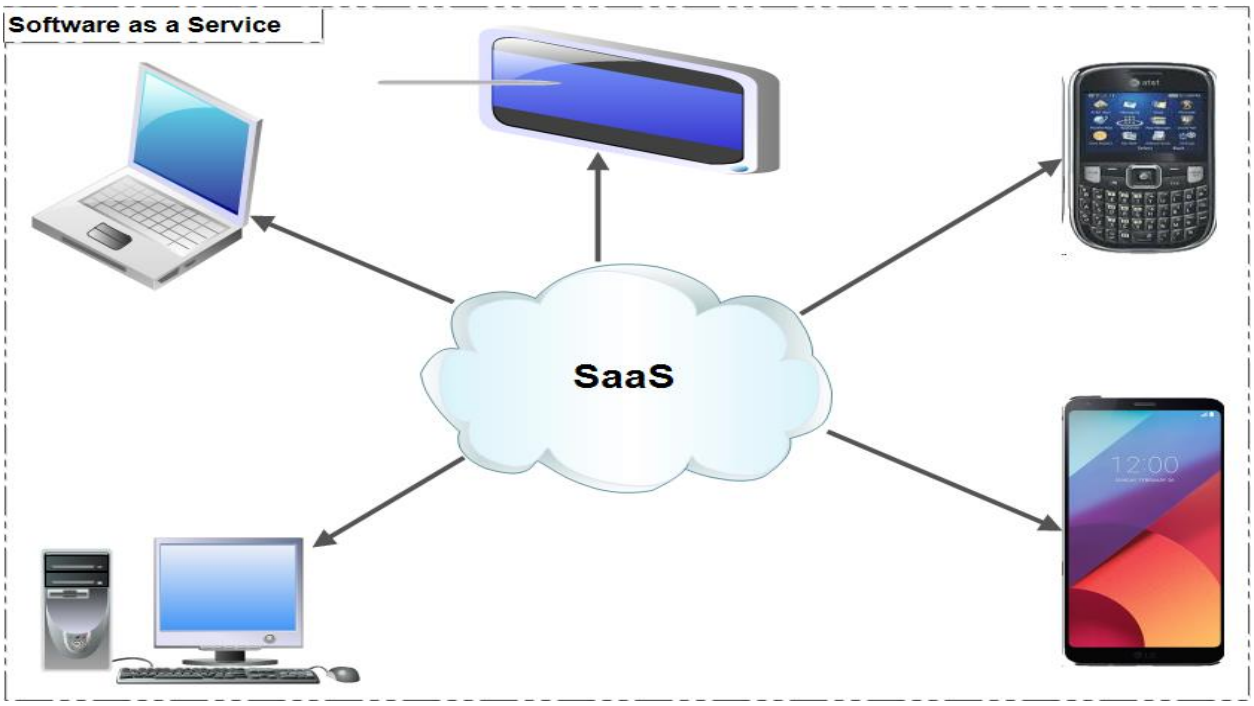

Figure 2-1: Software as a Service Application

compatibility etc. Examples of SaaS providers include Google Apps, Oracle on Demand, SalesForce.com and SQL Azure.

Backup as a Service (BAAS) provides backup and recovery operations from the cloud. The cloud-based BAAS provider maintains necessary backup equipment, applications, process and management in their data center. It is purchased backup and recovery services from an online data backup provider. Instead of performing a backup with a centralized, on-premises IT department, BAAS connects systems to a private, public or hybrid cloud managed by the outside provider. Backup as a service is easier to manage than other offsite services. As shown in the diagram below, Figure 2-2, different clients, platforms, and devices connect to the cloud storage service to store data and to recover. In the following section, we try to see different kinds of literature views that define data backup, recovery and efficient ways of data backup and recovery techniques and mechanism.

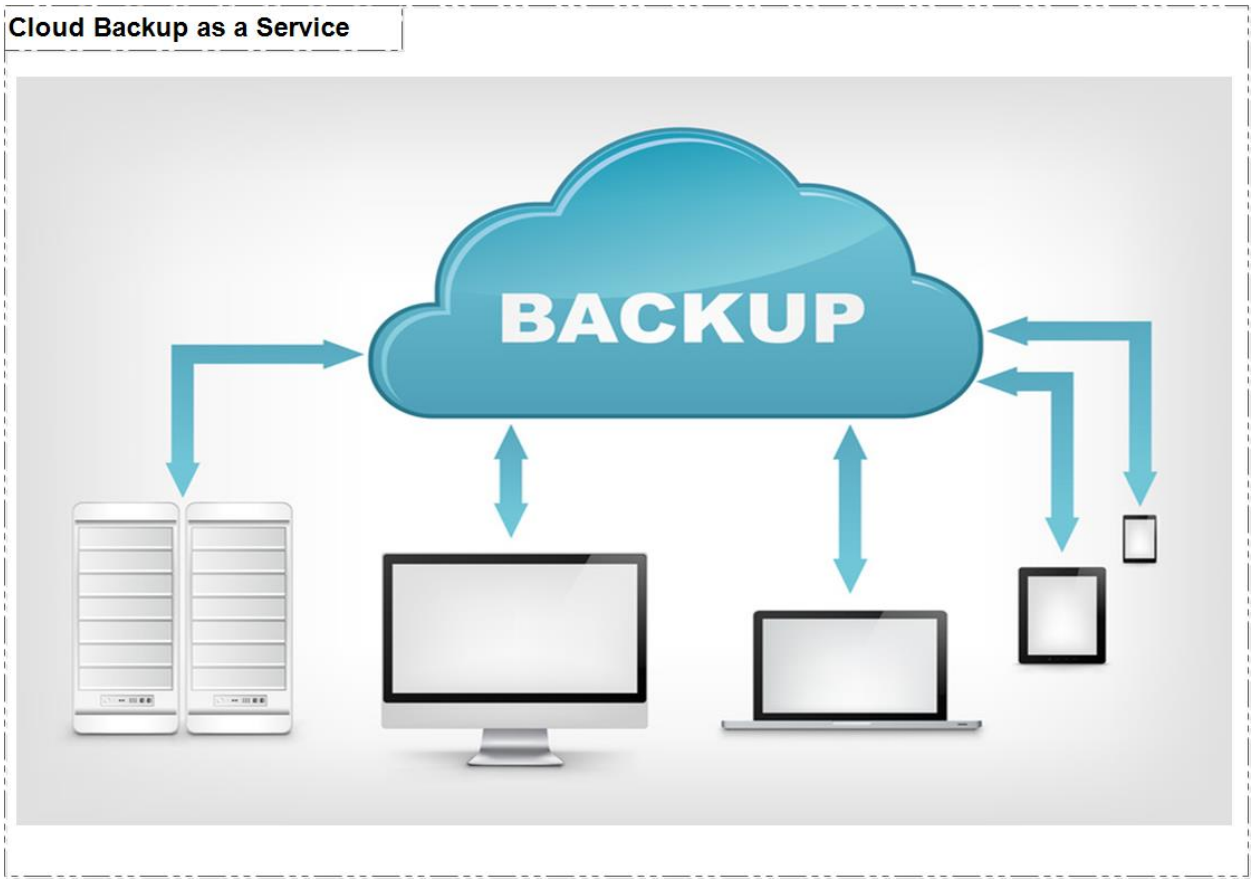

Figure 2-2: Cloud Backup as a Service

For the purpose of storage, many clients use cloud storage provider based on their interest and benefits the cloud provider that can provide. Some of the dominant forces in the cloud storage environment are explained below.

2.1.4. IAAS Providers

In this section, we discussed the most dominant forces in providing cloud environment and web services, like SAAS, PAAS, and IAAS, specifically on cloud storage service.

2.1.4.1. Amazon S3

Amazon Simple Storage Service provides a fully redundant data storage infrastructure for storing and retrieving any amount of data, at any time, from anywhere on the web. This cloud storage is cheap, reliable, availability of 99.95%, secure, scalable, supports encryption in transit and at rest and simple and easy to use. In addition to those, it has a large community, which is useful in case needing the support of communicating and integrating with Amazon s3.

2.1.4.2. Microsoft Azure Storage

Azure storage provides the flexibility to store and retrieve large amounts of unstructured data, such as documents and media files with Azure Blobs; structured NoSQL based data with Azure tables; reliable messages with Azure queues. This cloud storage is expensive in relative to Amazon S3, availability is 99.95%, supports encryption in transit and at rest and supports storage elasticity.

2.1.4.3. Google Cloud Storage

Google storage allows world-wide storing and retrieval of any amount of data and at any time. It provides a simple programming interface. This storage is reliable, availability 99.95%, Scalable, cheap, easy and fast networking infrastructure to perform data operations in a secure and cost-effective manner.

As we have listed above IAAS storage providers, there is no a clean definition which provider leads over the others as each provider have their own strength and weakness, the selection of the platform depends on users. So that we have decided to work with Microsoft Azure storage because the cloud service was available in hand.

2.1.5. Compression Algorithms

2.1.5.1. Huffman Encoding

This compression technique is lossless data compression and provides the opportunity for variable length representation in reading datasets. It generates tree while reading datasets based on the frequencies of each reading. with high-frequency readings at shallower leaf nodes than those occurring less often. Each reading's code is determined by traversing the tree from root to leaf, with each branch node providing one bit to the code. The length of each code is therefore determined by the depth of its leaf [31].

A Survey of Compression and Encryption Techniques for SMS [31] states that there are two types of approaches of Huffman encoding, the first is the traditional encoding technique called static Huffman encoding. This technique depends on the existence of a reading frequency distribution to generate the tree, and the compression performance achieved depends on how well the dynamic data conforms to that frequency distribution. The second is an adaptive Huffman encoding technique that dynamically generates and alters its tree based on the actual

reading frequency distribution as it occurs and changes. This does not require a previously constructed tree or any reading profiling. In the tree, each reading will carry its value along with the frequency of its use. This way, the tree can update itself, keeping the most frequent readings on the shallower levels to minimize their code lengths. This adaptability can potentially provide higher compression performance across all of the static and dynamic variables without additional programming and deployment effort. Therefore, this compression and decompression algorithm is suitable for low-performance devices and devices with low memory.

2.1.5.2. Deflate Compression

A Survey of Compression and Encryption Techniques for SMS [31] refers deflate algorithm finds duplicate strings in the input data; the second occurrence of a string is replaced by a pointer to the previous one. Duplicated strings are found using a hash table. All input strings of length 3 are inserted in the hash table. A hash index is computed for the next 3 bytes. If the hash chain for this index is not empty, all strings in the chain are compared with the current input string, and the longest match is selected. Finally, this Algorithm that has been implemented is not feasible for all formats of the file and takes more time and less Compression ratio.

2.1.5.3. LZW Compression

A Survey of Compression and Encryption Techniques for SMS [31] states that LZW algorithm replaces strings of characters with single codes. This technique doesn't do any data analysis on the incoming readings or text. Instead, it just adds every new string of characters it sees to a table of strings. Compression occurs when a single code is an output instead of a string of characters

2.2. Related Works

Data backup and recovery based on data de-duplication [1], try to analyze and compare data de-duplication with other data storage mechanisms and techniques, proposes for applying data de-duplication technology of data backup and recovery. Data de-duplication ensures data uniqueness, minimizes storage usage, bandwidth utilization, and environment protection by reducing needed for processing more duplicated data.

Cloud backup is a technology that involves sending a copy of your data to an offsite vendor, a cloud provider; that keeps the data secure and accessible in real time, usually for a monthly fee. What's great about the cloud is that it keeps a copy of your files at a safe and secure off-site location so you don't have to worry about losing data if your computer crashes or you accidentally delete a file or photo [12].

A novel approach for unique data backup in cloud storage [20] tries to provide efficient and complete backup and recovery solution in the cloud using data de-duplication. A unique data backup in a cloud storage can be maintained using the data de-duplication technology, which enables a unique copy of data on the cloud storage server. It states that data de-duplication is a data reduction process and technique that will be stored as part of data backup, in which it reduces data center expansion, reduce the network bandwidth, excessive power consumption, and the need of buying expensive data storage devices and server devices.

In [20], IT focuses on data center operating expenses, also known as green computing. Green computing implies that to environmentally sustainable IT by implementing energy efficient applications and reducing the power consumption. Utilizing the storage efficiently and effectively with minimal or no impact on the environment and with less IT management expenses should be the focus while designing backup application. Data de-duplication can be performed in client side or server side on client-server model architecture. Server-side data de-duplication is preferable because it minimizes the burden on clients. According to this paper, adopting and using cloud storage have the benefits of reduced cost; instead of building your own data center and buying storage devices, increase mobility and accessibility, simplified cost and

consumption model, the right option for addressing business changes, which is cloud is elastic resizable whenever is needed, simple integration and secured infrastructure.

Cumulus [2] refers to cloud computing is a getting popular IT infrastructure for hosting data and deploying software service. It tries to explain that backup is an attractive application for outsourcing to the cloud because is relatively simple, the growth of a disk capacity has created an efficiency and cost inflection point, and the cloud offers easy site-storage. Cumulus is a file system backup over the internet in a cloud storage server. It refers that building an efficient backup is based on providing thin cloud with an incremental backup of files from the client. It refers also that some cloud backup service providers develop software which will get installed on the client device and on the server, and this results in bandwidth consumption (compression, cleaning), locking on a single service provider. Cumulus has used Amazon S3 storage for prototype modeling.

The application of text compression to short message service using Huffman table [23] refers that developing an application using J2ME that can enable sending more SMS characters at a time is possible that the standard SMS number of characters, 140 each of 8 bites at a time.

In [23], states that SMS is the way of sending short messages in a quick and relatively cheap price. Using the standard SMS service only allows 140 characters each of 8 bites, and this results in difficulty for a person who wants send more characters than the standard 140 characters 8-bit, 160 7-bit characters or 70 characters for 16-bit Japanese, Mandarin and Korean language. To overcome this difficulty of sending more characters in one message, text compression is needed. Compression is the process of encoding data using a smaller number of bits from the sender side so that the small number of bits represents the same information as the actual message. SMS compression application is made while the sender sends a message, and decompression will get processed at the receiver. The compression and decompression processes are performed for sake of bandwidth utilization, including more message characters at a time and security.

According to [23], text compression using Huffman table in SMS between sender and receiver results in overall compression SMS with average compression ratio 28.7%.

Android Application for SMS Compression [26] explores and compares two different compression-decompression algorithms for adopting one better algorithm in the android application for Android devices to optimize the maximum character capacity of SMS body. According [26], there exists two data compression technique, arithmetic coding, and Huffman coding. Arithmetic coding requires additional memory space and high precision and effective encoder-decoder to calculate and present its code number, while Huffman coding tries replacing frequently occurring symbols are coded using shorter code words. Comparing those techniques Huffman coding is better at compressing and decompressing techniques rather than arithmetic coding which needs high performance, and additional memory space, not preferable for small devices.

Review of Java 2 Micro Edition (J2ME) [24] states wireless technology has changed mobile communication from voice-oriented to extensible, internet-enabled communications with advanced data and software support. Nowadays there is a great demand for multi-functional mobile devices capable of hosting a broad range of applications software for both commercial and personal use. J2ME offers the wireless community a standard solution on different platforms without significant changes to the system. Almost all mobile phones available on the market support the programming language Java for J2ME. Java is a modern object-oriented language and has far better features and higher-level programming constructs than other languages and tools that are used for wireless software development, it allows software to be developed more efficiently. Nowadays, almost all mobile phones available on the market support the programming language Java for Java 2 Micro Edition (J2ME). J2ME, including other optional packages, allows developers to implement platform-independent applications for mobile devices. Java has become the major object-oriented programming language for developers to implement new mobile applications, which benefit from Java's well-known features for design of graphical interfaces.

Interactive use of cloud services [25] explores and evaluates the interactive usage of Amazon cloud services SQS and Amazon S3. It refers that use of cloud services is an interest to science end users, including for storing and accessing shared data sets. It also states that as uploading large data sets to Amazon S3 may suffer system bottleneck, developers or client should select

carefully how to communicate with the data center, Amazon S3. Amazon S3 provides five different approaches for transferring data to the server. The first one is, The Amazon Web Console provides a user-friendly graphical interface, and most users choose this by default. However, it does not seem to support multipart file uploading. The second one is, s3cmd is a popular command line tool, which is useful for shell scripting. The third one is, Cyberduck is an easy-to-use GUI application, written in Python using the jets3t library. The fourth one is, AmazonS3Client.putObject() is a low-level Java API that uses a single connection per transfer. The fifth one is, TransferManager.upload() is a high-level Java API that uploads a file in parts using multiple connections and threads. According [25] comparison between the five interfaces TransferManager achieves the highest throughput followed by Cyberduck.

Amazon S3 [22] is an Amazon web service that enables to store, collect and analyze data regardless of format all on a massive scale through the internet. Amazon S3 is object storage built to store and retrieve any amount of data from anywhere, websites and mobile apps, corporate applications, and data from IoT sensors or devices. It can be utilized for Backup and Recovery, Data Archiving, and Big Data Analytics. Amazon S3 gives any developer access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of websites.

2.2.1. Related Work Summary

Table 2-1: Related Work Summary

Title	***Author***	***Year***	***Description***
Data backup and recovery based on data de-duplication	Guo-Zi Sun, Yu Dong, Dan-Wei Chen, and Jie Wei	2010	Stores unique data using de-duplication and compressing technique to minimize storage needed.
A novel approach for unique data backup in cloud storage	Ch.anilkumar and Dr.R.Kirshnaiah	2013	Removes unwanted redundant and duplicated data to minimize data center expansion, empowering green computing,

			and cost of storage devices
Cumulus: File System Backup to the Cloud	Michael Vrable, Stefan Savage, and Geoffrey M. Voelker	-	A file system backup to Amazon S3 based on the thin cloud to avoid vendor lock-in; by providing simple client interfaces.
The application of text compression to short message service using Huffman table	Ahmad Affandi, Saparudin, and Erwin	2011	A J2ME app for compression of SMS characters so that a user can send more characters than the standard characters.
Android Application for SMS Compression	Kirti Madhukar Battase, Neha Vijay Barve, Parinita Bajirao Kandekar, Lata Vitthal Sanap	2014	Compares two different compressing techniques for SMS data compression of mobile phones in Android devices, and develops an android app using best compression algorithm so that users can send more characters than the standard one using Android devices.
Review of J2ME and J2ME-based Mobile Applications	Anna Isakov and Hao Shi	2008	Explores the J2ME capability, libraries, main features and J2ME app usage in mobile devices.
Interactive use of cloud services: Amazon SQS and S3	Hobin Yoon, Ada Gavrilovska, and Karsten Schwan	2012	States the advantage of cloud storage services, especially AWS, and data uploading options to Amazon S3.

2.2.2. The Research Gaps Considered in this Study

Summarizing the reviewed list of literature, most of these literatures are implemented backup and recovery solution for mobile phones using data encryption, data compression and decompression, and de-duplication technologies. Some of the backup and recovery services are based on a local computer, and others are implemented using the emergent and recent technology cloud storage, Amazon S3. But most of the solutions provided only focuses on smartphones; Android phones, windows phones, and iPhone.

Moreover, most of the smartphone vendors also provide built-in backup and recovery solution. In this study, we tried to cover the uncovered topics in the literature'. For example reviewed literature' solutions don't consider feature phone devices data backup and recovery, and the data movement from one platform to another platform, called data-portability. In addition to the above gap, the literature' only focuses on utilization of the internet, while our solution considers both networks, the Internet and GSM network, and the smartphones and the feature phones. Finally, in this solution, we are adding a new feature called P2P BR using the GSM network of small devices, feature phones. P2P BR is a backup and recovery component designed for feature phones, in which it enables that a specified user of feature phone can store contacts, call logs, events, to-do list and messages on peer's device using only the GSM network so that they can recover it back to the owner device using the GSM network.

Chapter 3 : METHODOLOGY

3.1. Research Design

GSM network is a reliable network solution which provides reliable communication of mobile users with each other. GSM service is used for calling another mobile user in real-time and SMS service, for sending a limited number of characters messages at a time. While cloud storage is the recent and efficient technology to be used for storing users' data on a remote server as shown in the Figure 3-1 below, so that the users can access it whenever they want, and wherever they go.

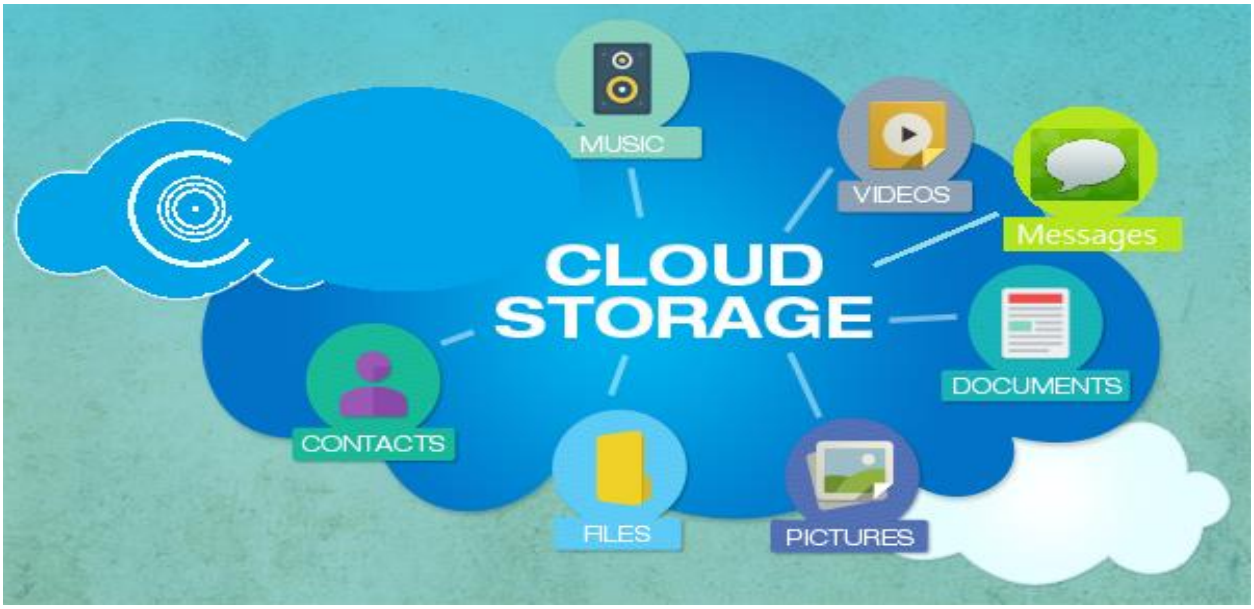

Figure 3-1: Solution's Data to/from Cloud

DatayeN is a backup and recovery solution which focuses on synchronizing data from GSM network and internet using J2ME and Android apps respectively, in cloud storage. GSM network users, especially many feature phone users failed to restore their data when they lost their device, or the device is damaged. Either when a user moved from using feature phone to being smartphone user or vice versa, the user is expected to move the data manually to the current device. By synchronizing, users of feature phones can store their data on the cloud storage so that when they needed to recover it, they can recover it from any device. In addition to this, uses' of feature phone can use P2P BR solution, in which their contacts, to-do list, events list or messages can store with peer users so that they can recover it from the peer. The primary objective of this research is that by utilizing GSM network SMS service to provide backup and recovery solution

for users. Once the solution is provided, its reliability, validity, and acceptance will get measured by users.

3.2. Process Model

In this study, we have used the following solution development process modeling tasks, representing in Figure 3-2.

Observation and Initialization: **-** starting, preparing specification to solve the observed problems and the proposed solution.

Data Collection: **-** preparing interview questions and focus group and analyzing survey result.

Requirement Gathering and Literature Review: **-** reviewing related articles, journals, and books, and collections other solution related requirements.

Design: **-** specifying deployment structure, designing system architecture, requirement modeling, class diagram, and UI.

Prototype: **-** developing solution prototype which includes the main feature of the proposed solution.

Cloud Storage Integration: **-** integrating the developed prototype with the cloud storage, Microsoft Azure Storage, and testing the integration.

Testing: **-** complete functional testing of the porotype and verifying that the provided solution is working what is intended to. While testing the solution repetitive process was used to fix bugs, additional requirement, and adding features starting back from requirement gathering, red-designing, improving the prototype, testing the integration and finally functional testing.

Production: **-** is the final process, deploying the app on clients' devices.

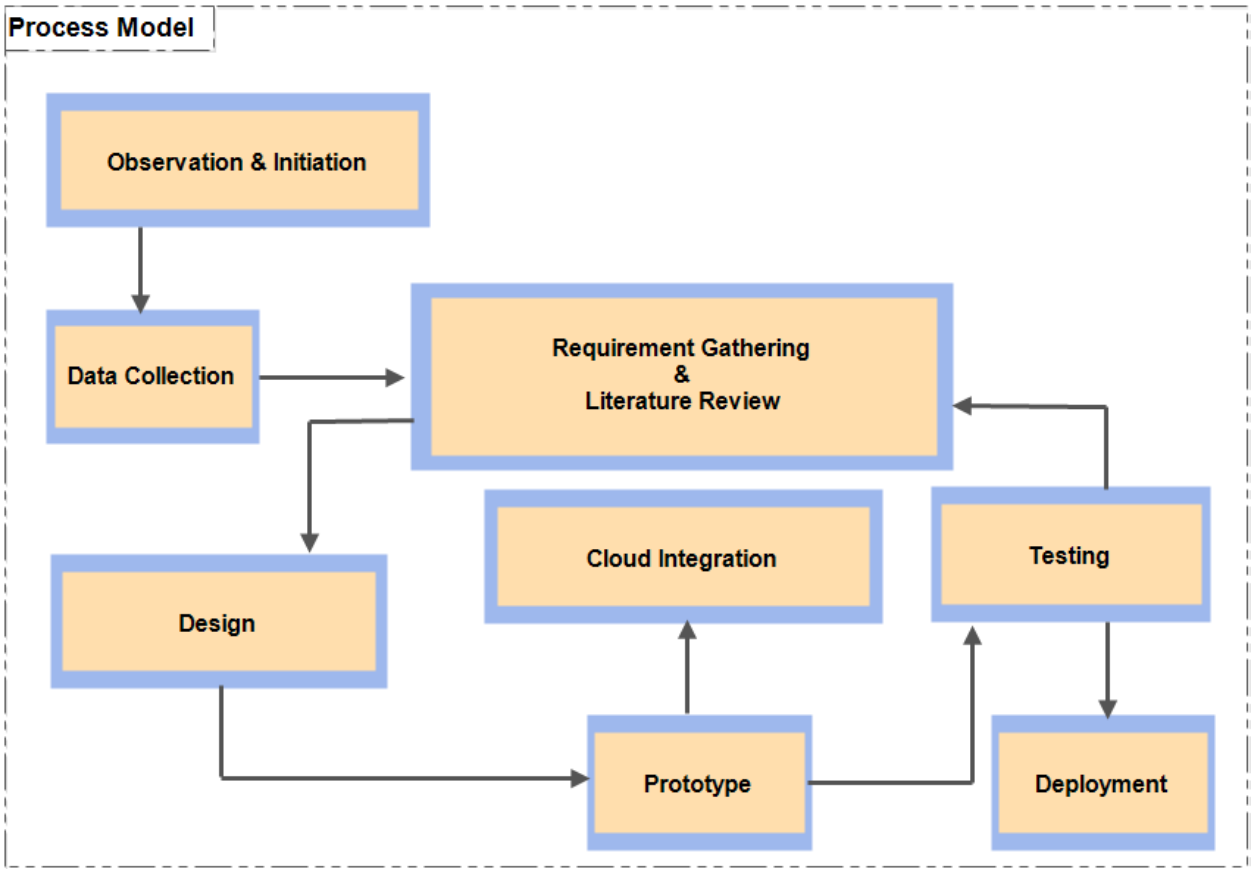

Figure 3-2: Process Model

3.3. Data Source

To explore the need for backup and recovery solution in both mobile technology, challenges, and opportunities, we have used many data source. Those data sources are a literature review of different journals, books, and scientific articles and websites. In the data collection technique, we have used questionnaire and focus group discussion.

3.4. Data Collection Techniques

3.4.1. Questionnaire

The questionnaire is the most dominant and practiced data collection technique because it makes to addresses more people and gets more opinion. We have prepared that a hundred copy of the written questionnaire, thirteen questions in each copy.

3.4.2. Focus Group

In this technique, we created five focus groups, which are Ethio-Telecom customer service agents, in which each of them has five members, groups for discussion and exploring the need of data backup and recovery on both GSM network and internet and synchronizing users' data on any device.

3.5. Sampling Techniques

For the purpose of data collection, the participant sampling is done on university students: - BSc students, Postgraduate students, and Ethio-Telecom employees, usually called customer services agents. The primary objective of this survey is to explore the need for data backup and recovery using GSM network, and synchronizing with data backup and recovery using internet to cloud backup, so that users of feature phones can store their data on the remote cloud storage server, when needed they can recover it either using feature phone (GSM network), or using smartphone (internet). For the sake of representing the different groups of users, the sampling technique was purposive or judgmental and the numbers include 55% participants were BSc students, 20% participants were postgraduate students, and 25% of them Ethio-Telecom customer agent service employees which are also used for the purpose of group discussion.

3.6. Tools Used

- ***Microsoft Office Word 2016:*** - we used this tool for the presentation and formatting of questionnaires and focus group discussion questions.
- ***Microsoft Project Management 2016:*** - we used this tool for presenting and formatting project and project's tasks breakdown with the timeframe.
- ***Microsoft Office Excel 2016:*** - we used this tool for maintaining, keeping and analyzing survey response, and for presenting the survey responses into the graphical presentation, bar charts.
- ***Grammarly:*** - A software plugin for Microsoft office word to fix contextual spelling, grammar, punctuation and sentence structure document problems.
- ***Edraw Max 6.8:*** - A software tool which was used for representing figures, diagrams, flowcharts and different modeling.

Chapter 4 : PROPOSED ARCHITECTURE OF THE SYSTEM

4.1. System Architecture

In this section, we presented the system architecture of the proposed solution, which contains four large components, and subcomponents found in each component as shown in Figure 4-1.

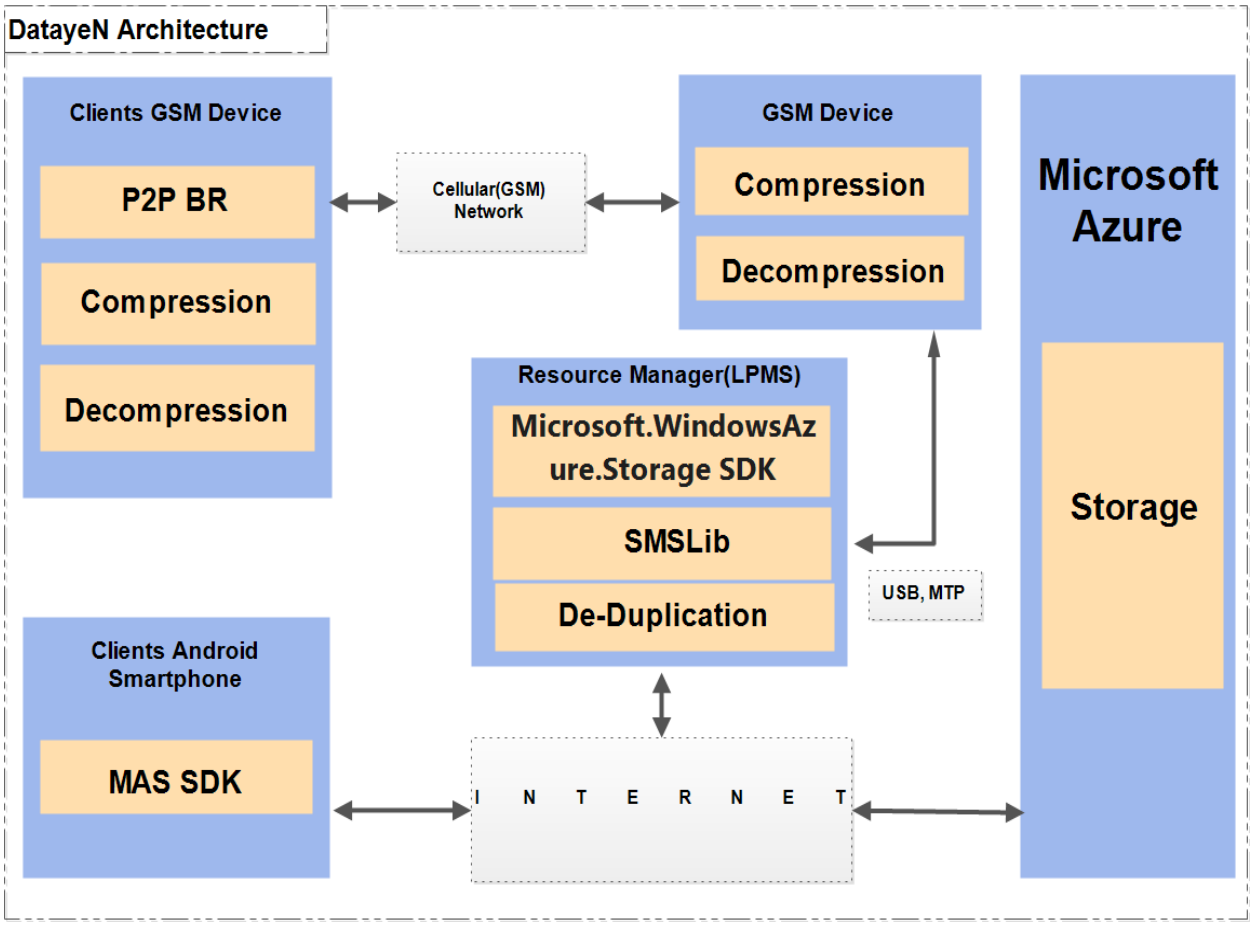

Figure 4-1: System Architecture

The system architecture components and subcomponents are described below one by one.

4.1.1. Microsoft Azure Web Service

This component is a web service provided by Microsoft, as shown in the Figure 4-2 below, in which client can communicate through the web to get different Microsoft Azure services. Microsoft Azure provides different web services; Microsoft Azure Storage is one of the services provided for storage service. Microsoft Azure Storage is the best reference for cloud storage

service next Amazon s3, it provides virtually elastic and unlimited storage space, and accessible through the internet using a set of simple interfaces, put, get, delete and list.

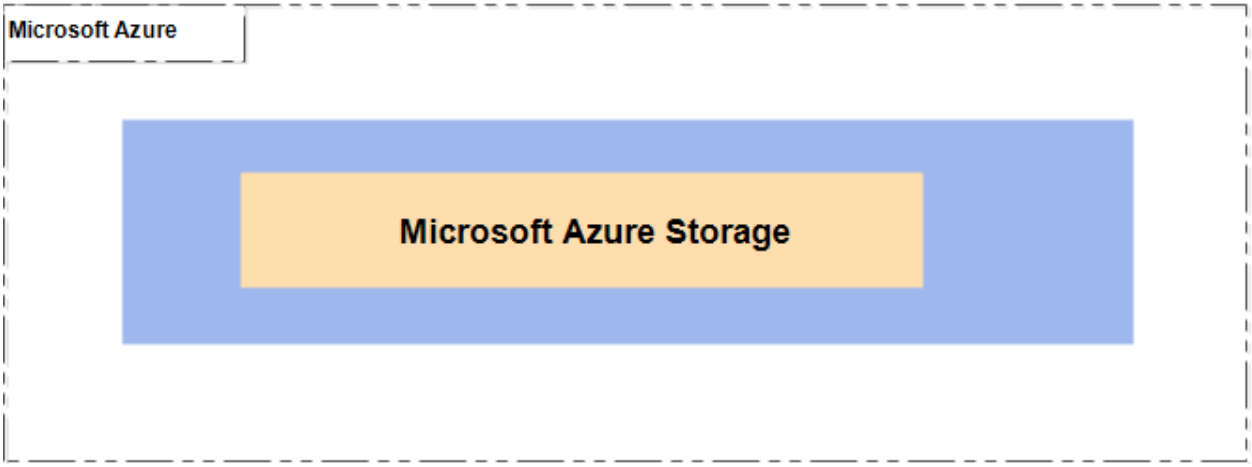

Figure 4-2: Microsoft Azure Storage Component

Microsoft Azure Storage is a storage service, in which users throughout the world can use for the purpose of data storage, by allowing users to create containers for object storage. To use Microsoft Azure Storage, a user must have Microsoft Azure account to create a container and to gain access to using the objects found in the data storage. The following Figure 4-3, shows the communication between Microsoft Azure Storage and clients.

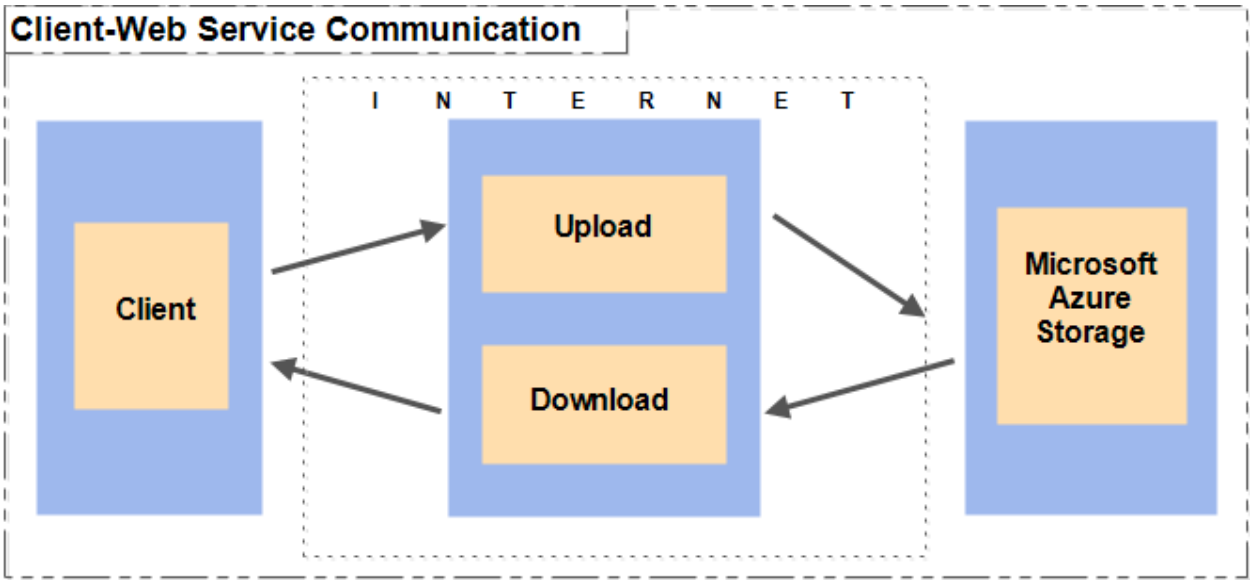

Figure 4-3: Client-Microsoft Azure Storage Communication

Microsoft Azure Storage maintains users' data inside containers, those containers maintain folders which are going to store objects, files, and other documents. The container, which has its own unique name and has a structure as shown, Figure 4-4, below. In the diagram represented below, Microsoft Azure Storage stores file only inside the container. A container is the largest component of Microsoft Azure Storage, which must have a unique name for all-around storage service. Inside the container, users

can create multiple folders of a different category. Then after inside the folders, files can be stored using the uploading interface provided by Microsoft Azure Storage. Microsoft Azure Storage provides different interfaces to end-user clients, such as reading data, deleting data, listing data. For the integration of this research solution with Microsoft Azure Storage, Microsoft Azure Storage web service provides an Android SDK client called Microsoft Azure Storage android SDK. The SDK provides direct access to many Microsoft Azure Storage service API's.

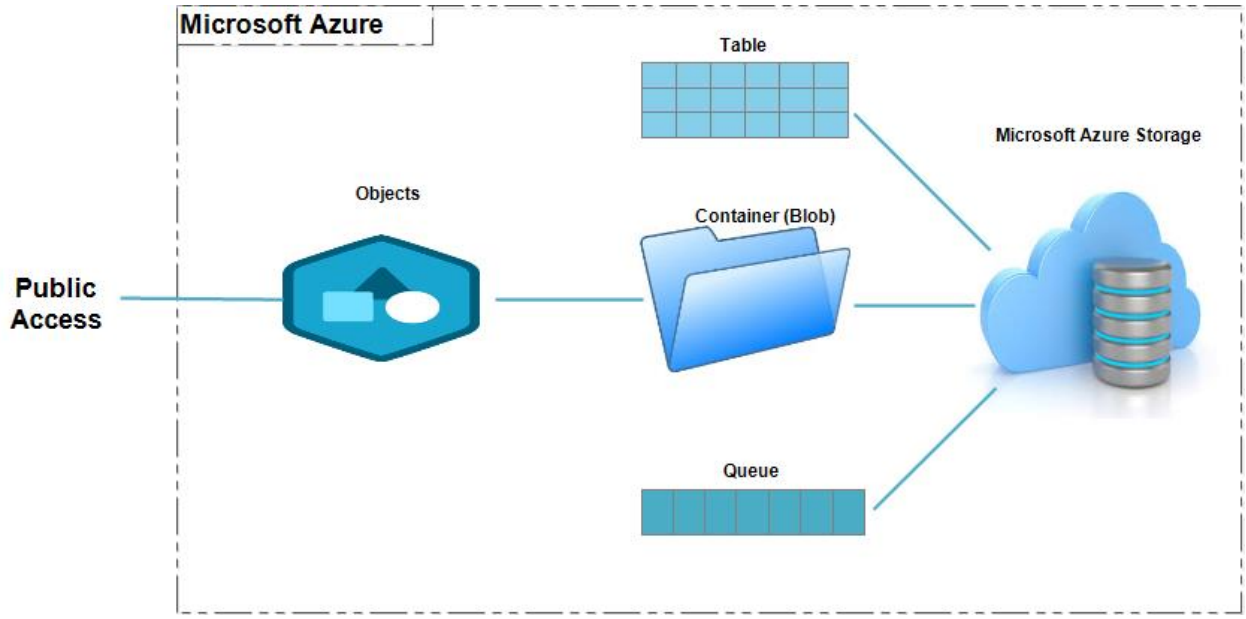

Figure 4-4: Microsoft Azure Storage Structure and Client Interaction

4.1.2. Feature Phone App

This component is a J2ME application, an application developed for feature phones which enable users to send their contact addresses, messages, call logs, and events to the remote cloud storage server using the cellular network as shown in the diagram, Figure 4-5, below.

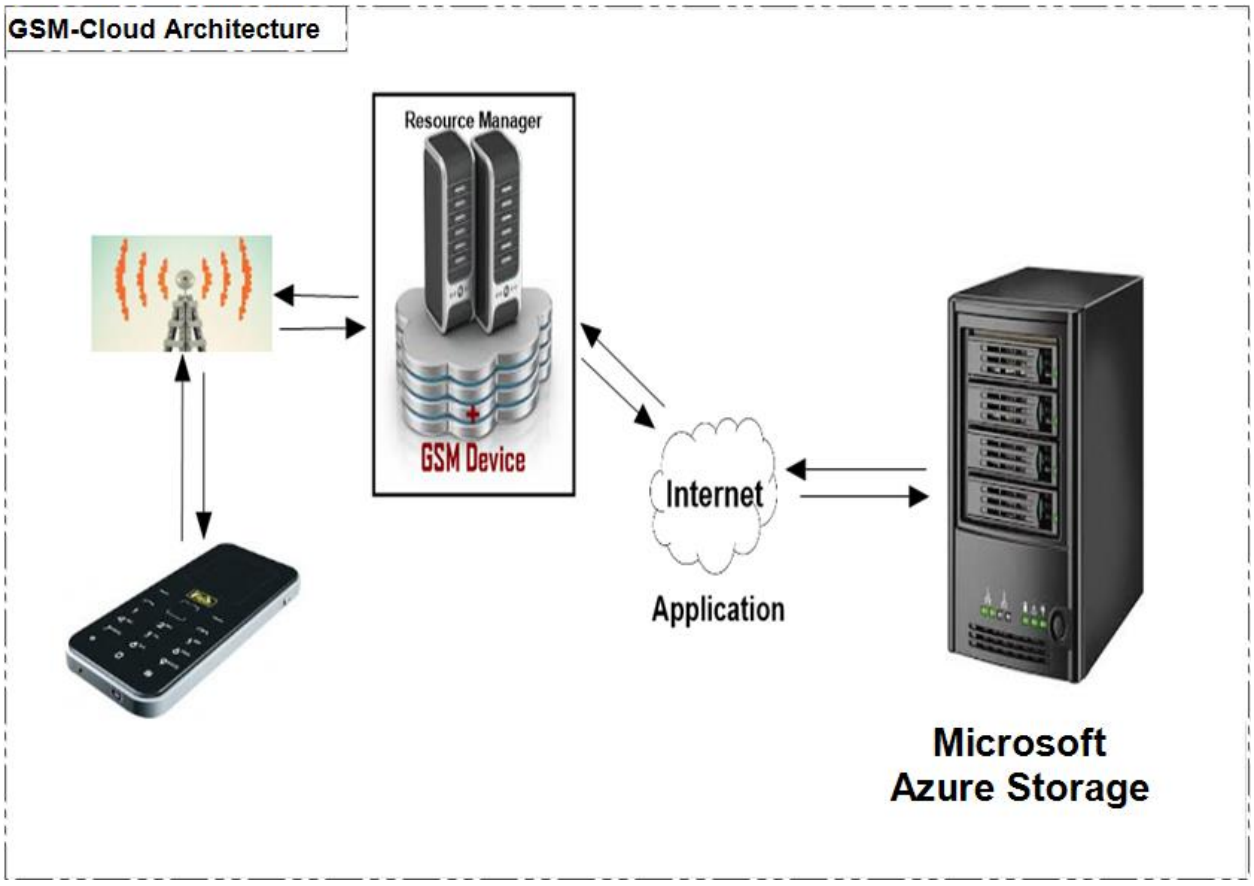

Figure 4-5: GSM-Cloud Architecture

In the diagram, a J2ME app installed feature phone sends data using the cellular network to the resource manager which contains connected GSM device that can extract contents sent using the cellular network, and then the resource manager pushes the data to Microsoft Azure Storage after performing data-duplication.

The J2ME app contains Figure 4-6, P2P BR, compression and decompression components. The compression component used for representing multiple characters per a single SMS and the decompression component is used for decompressing the compressed components. P2P BR component is used for enabling mobile users to store backup data with peers, so they will recover it by sending the request. The compression and decompression algorithm used in this J2ME application is called Huffman table, because it is efficient, as the mobile devices have limited computing resources. like processor and memory. Using this cellular network, the content types that can be transferred to/from the cloud server are defined below.

Table 4-1: Mobile Based Data Transfer Classification

Mobile Phone Device	***Mobile data types that are going to get uploaded, downloaded from and into mobile phone device***								
	To-do List	Events	Contact	Call log	Message	Photo	Music	Video	File
Feature Phone	Yes	Yes	Yes	Yes	Yes	No	No	No	No
Smart Phone	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

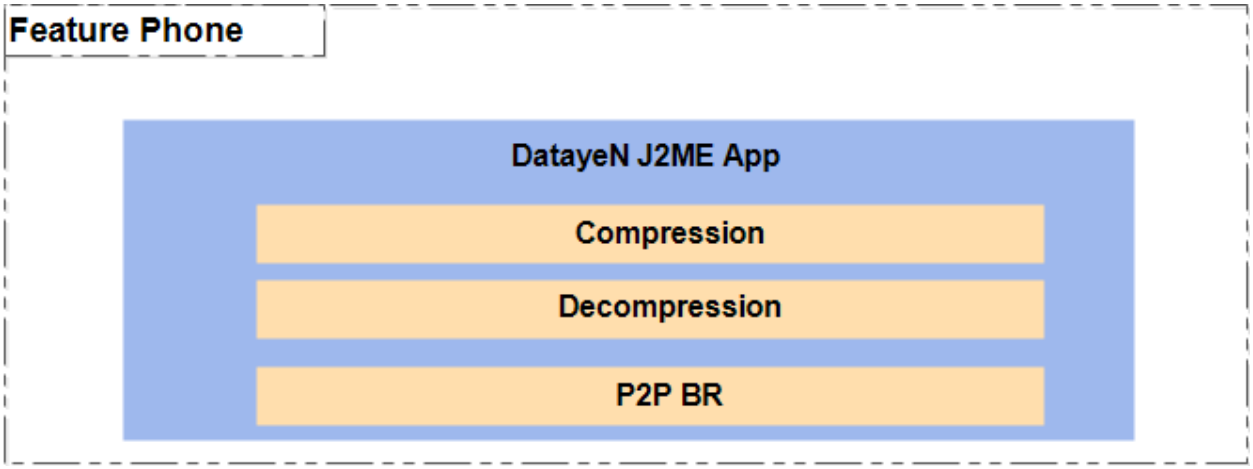

Figure 4-6: J2ME Components

P2P BR is a J2ME app component, which is used for the purpose using peers' feature phone as a storage device, or as a secondary storage for storing backup and recovering when need from the peer's GSM device. There no another way to clients to use this, rather than selecting or entering the client's number. After selecting the peer storage, the system works on normal procedure, like compressing the records to be sent to the peer's storage device. At the end of the storage or the peer's device, the data as backup stays compressed but when the data owner recalls the peer's device to recover the backup data; the peer's device sends back the compressed data to the owner.

4.1.3. Smartphone App

This component represents, as shown in the diagram below, Figure 4-7, for a smartphone application, android application. The application is developed using the Android SDK using android studio IDE. The android application is get compiled and built. The library will get integrated, with Microsoft Azure Storage SDK that exposes Microsoft Azure Storage SDK API's. The data type that can be transferred to/from a device and network is represented in Table 4-1. The architecture of communicating applications with the cloud storage service described in the diagram below, Figure 4-8.

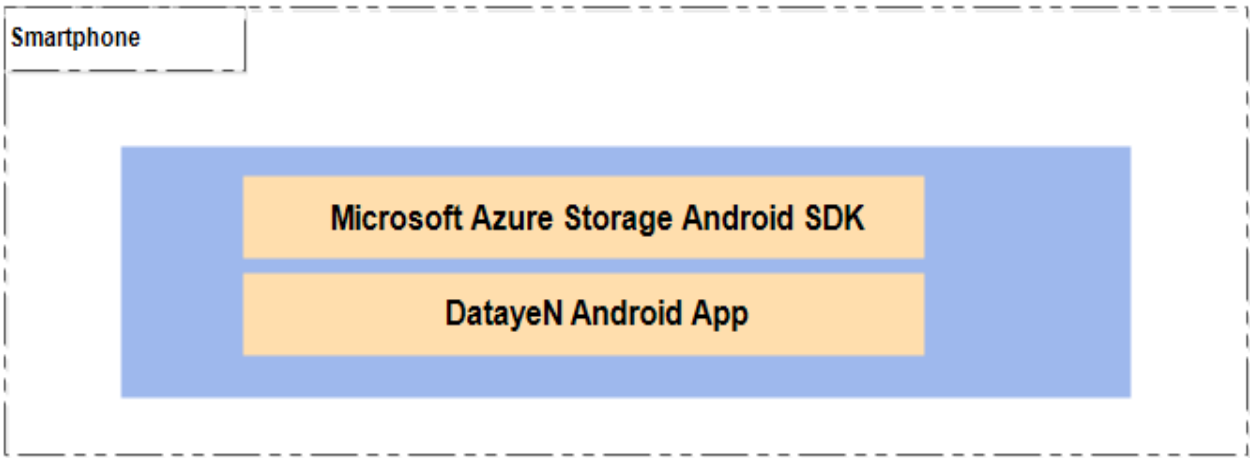

Figure 4-7: Smartphone Application

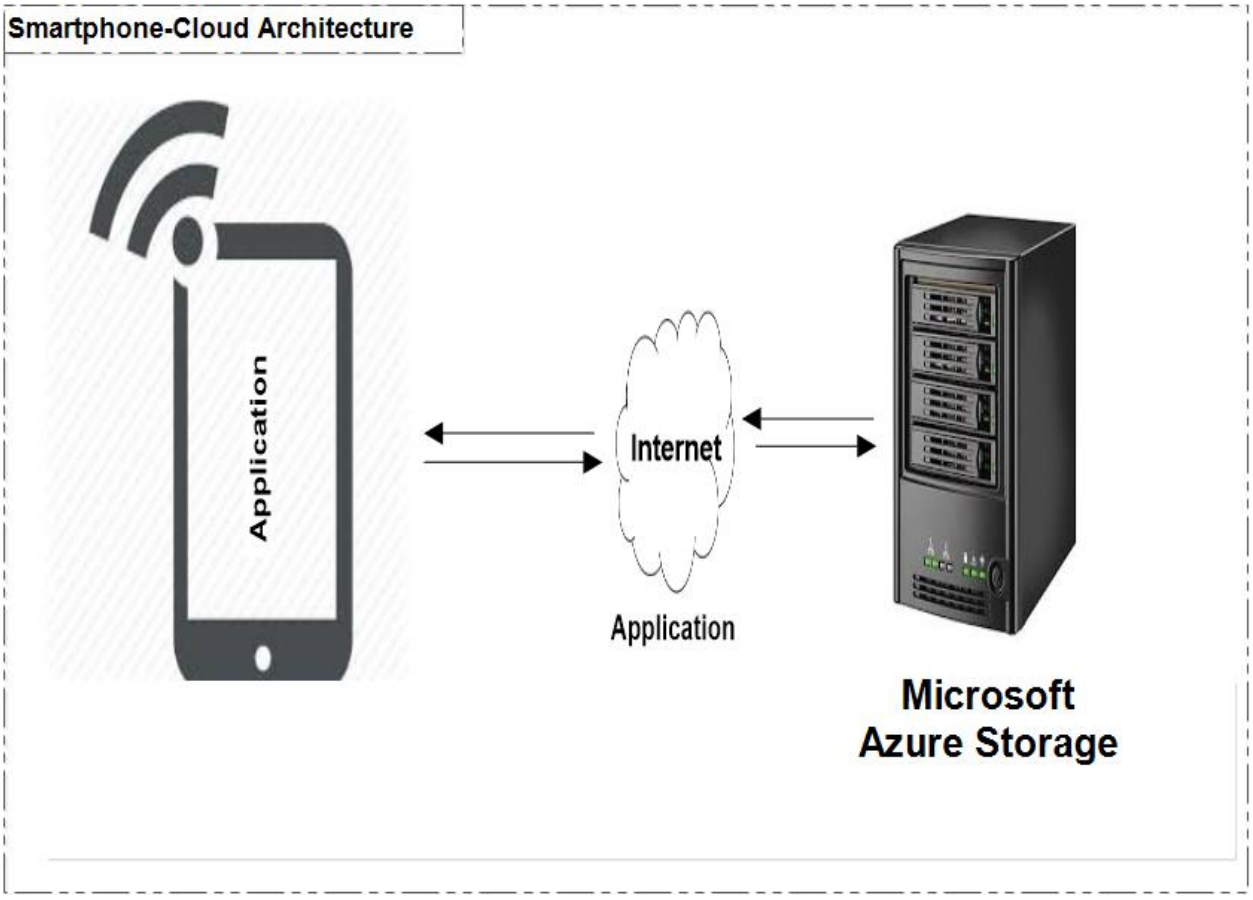

Figure 4-8: Smartphone-Cloud Architecture

4.1.4. Resource Manager

This component is a bridge between Microsoft Azure Storage and users' feature phone J2ME application, for sending and downloading data, performing data de-duplication, and maintaining local mirror server for recent access as shown below, Figure 4-9. This component also extracts requests coming from GSM network.

- ***DatayeN.Net Project:*** **-** this is the resource manager .Net project of the backup and recovery solution designed for feature phones to connect with Microsoft Azure storage using the internet, TCP/IP. This component responsibility is to push or synchronize files extracted from *GSM device* using the *SMSLib*. While synchronizing; it performs removing duplicated records using the *De-Duplication* process.
- ***De*-Duplication: -** is a process of avoiding redundant records existing two or more times in a file but maintains record number of existence frequency.
- ***SMSLib:*** - a .Net SMS library used for extracting SMSs found in mobile devices which is connected directly to the resource manager, the LPMS. This library is going to be installed on **.**Net project.
- ***GSM Device:*** - is a device installed with DatayeN.jar app which accepts GSM request from feature phones. This device is connected directly to LPMS server using the USB cable. DatayeN.jar contains compression and decompression algorithm.
- ***Compression & Decompression:*** **-** this algorithm is designed using Huffman coding, increase the number of characters to be sent, and for recovering the actual content.

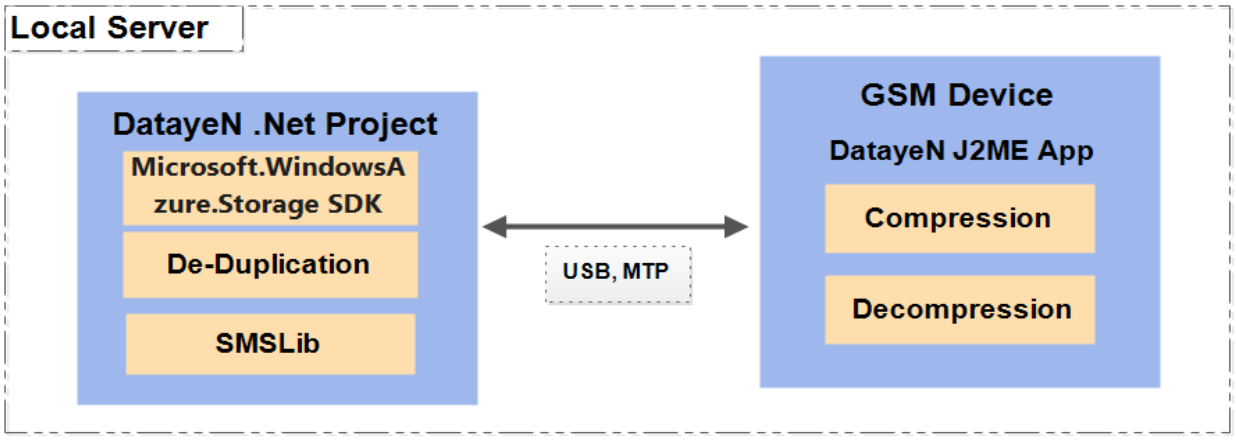

Figure 4-9: Resource Manager Component

Chapter 5 : IMPLEMENTATION OF THE SYSTEM

5.1. Requirements Modeling

In this study, the mobile user is the only external actor, while the system performs many activities behind. The mobile user is only expected to synchronize user's data with the remote cloud storage server, and another extended service is P2P BR, storing a copy of data with their peers. The system actor performs compressing, decompressing, and performing data de-duplication. We use the following use case diagram, as shown in the Figure 5-1 below, to represent the functionalities and the interaction with actors or users.

5.1.1. Use Case Diagram

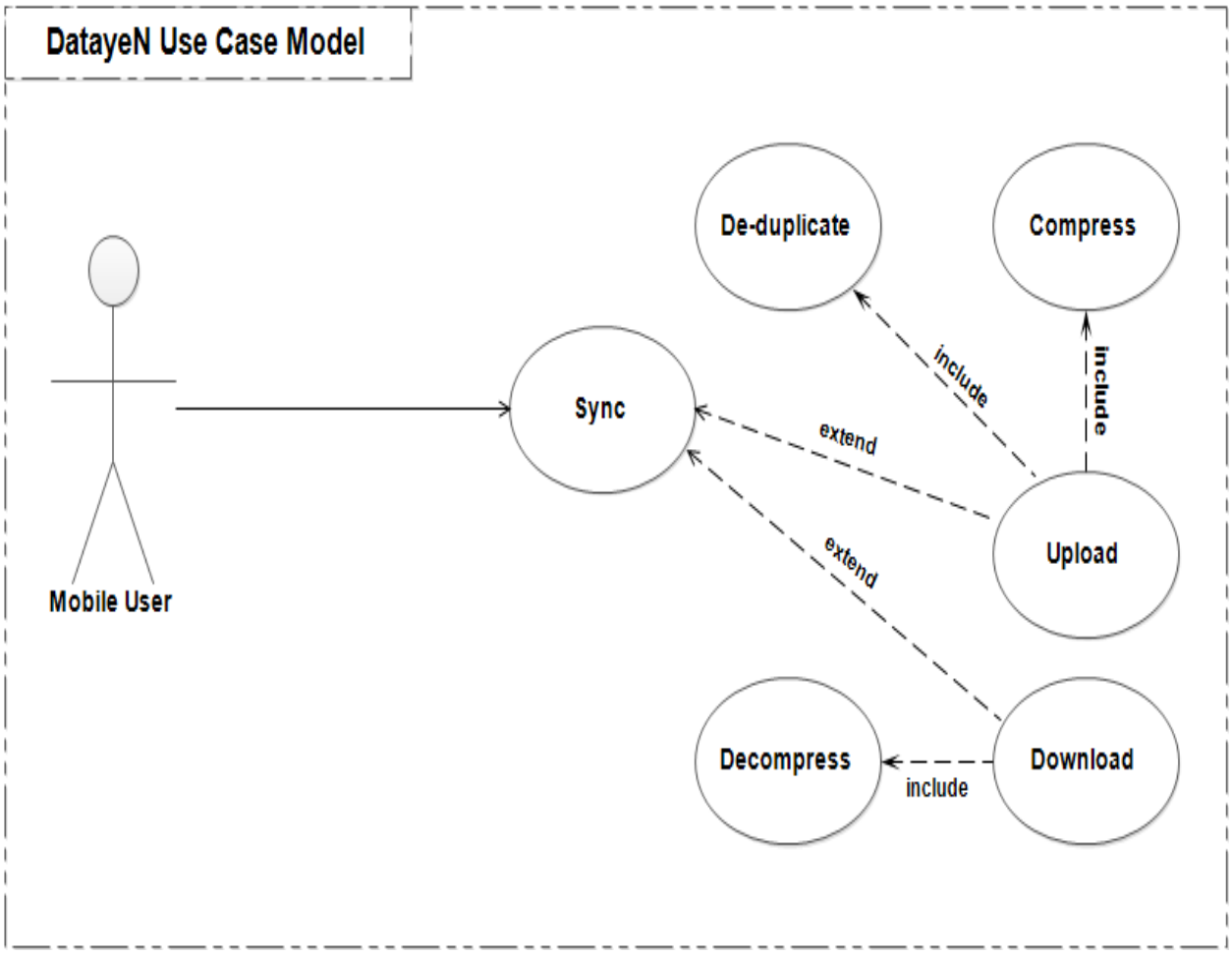

Figure 5-1: DatayeN Use Case Model for Feature Phone

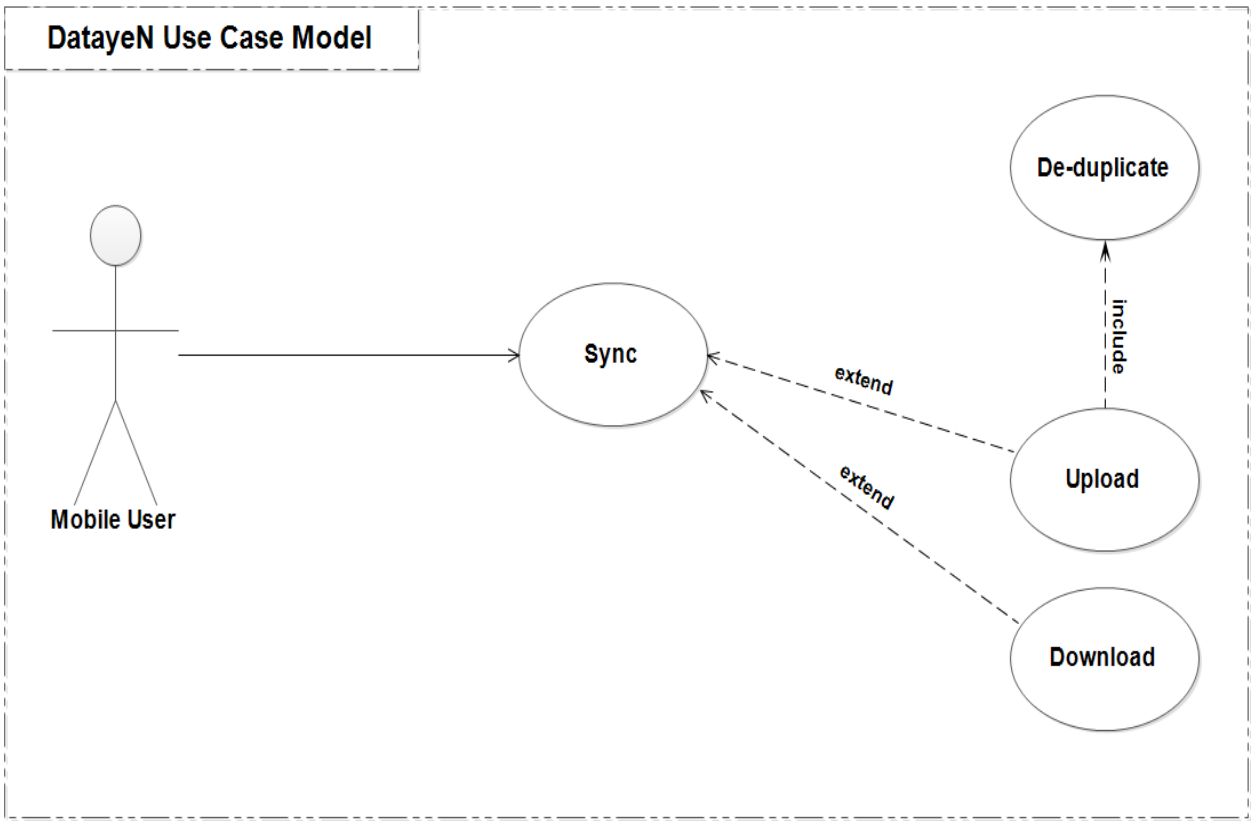

Figure 5-2: DatayeN Use Case Model for Smartphone

5.1.2. Use Case Description

5.1.2.1. Sync

Table 5-1: Sync Use Case Description

Name	Sync
Description	This system feature performs mobile users to interact with the cloud storage server and synchronize their mobile data with the server, implies it download the server data and merges with local data, and pushes it back into the storage server
Actor	Mobile User
Precondition	1. Internet should be available. 2. The mobile should be subscribed.
Basic Flow	1. Select the data type to synchronize. 2. Click the **Sync** button found on the on the bottom bar. 3. Then the system downloads the server version file and merges

	with local data. 4. Finally, synchronized and latest version of the information will be available on the storage server and local device.
Alternative Flow	1. Enter the phone number for a subscription. 2. Select the data type to synchronize. 3. Click the **Sync** button found on the on the bottom bar. 4. Then the system downloads the server version file and merges with local data. 5. Finally, synchronized and latest version of the information will be available on the storage server and local device.
Exception Flow	1. No internet connection. 2. Synchronization failed.
Post-condition	Synchronized and latest version of information on both local device, and storage server.

5.1.2.2. Compress

Table 5-2: Compress Use Case Description

Name	**Compress**
Description	This system feature has not a direct interaction with users, but it is referred when the user tries to synchronize the local data with the server. Its main functionality is that to minimize the number of characters to be sent to the server using the GSM network.
Actor	System
Precondition	Actual data must be available.
Basic Flow	1. The user selects to Sync specific data. 2. Compression accepts data blocks. 3. The system represents the actual data into a minimum number of bits with the same meaning.
Exception Flow	1. No internet connection. 2. Synchronization and compression failed.

Post-condition	A minimized version of the actual data.

5.1.2.3. Decompress

Table 5-3: Decompress Use Case Description

Name	Decompress
Description	This system feature has not a direct interaction with users, but it is referred when the user tries to synchronize the local data with the server. Its main functionality is that to minimize the number of characters to be sent to the server using the GSM network.
Actor	System
Precondition	Compressed data must be available
Basic Flow	1. The user selects to Sync specific data. 2. decompression accepts data blocks. 3. The system represents the compressed data into actual data representation.
Exception Flow	1. No internet connection. 2. Synchronization and compression failed.
Post-condition	Compressed data converted into actual human-readable representation.

5.1.2.4. *De*-duplicate

Table 5-4: De-duplicate Use Case Description

Name	De-duplicate
Description	This system feature is a de-duplication process for removing duplicated data blocks in a file before the data is uploaded to the remote cloud storage server.
Actor	System
Precondition	1. Internet should be available.

	2. The mobile should be subscribed.
Basic Flow	3. The user selects to Sync specific data. 4. The system requests to download a file to the remote cloud storage server. 5. downloads the specific data. 6. Compares the downloaded data block by block with local data blocks. 7. Removes found the duplicated data and writes unique block to the file. 8. Maintains a clean file with non-redundant data blocks.
Exception Flow	1. No internet connection. 2. Downloading or uploading failed.
Post-condition	Clean file with non-redundant data blocks.

5.1.2.5. Upload

Table 5-5: Upload Use Case Description

Name	**Upload**
Description	This system feature used for storing content or file on the remote cloud storage server.
Actor	System
Precondition	1. Internet should be available. 2. The mobile should be subscribed.
Basic Flow	1. The user selects to Sync specific data. 2. The system requests to upload a file to the cloud storage server. 3. uploads the specific data.
Exception Flow	3. No internet connection. 4. Uploading failed.
Post-condition	Client data is uploaded to the cloud storage server and synchronized.

5.1.2.6. Download

Table 5-6: Download Use Case Description

Name	Download
Description	This system feature used to bring a file from the cloud storage server to the local device for the purpose of data synchronization.
Actor	System
Precondition	3. Internet should be available. 4. The mobile should be subscribed.
Basic Flow	1. The user selects to Sync specific data. 2. The system requests to download a file from the cloud storage server. 3. Downloads the specific data.
Exception Flow	1. No internet connection. 2. Downloading failed.
Post-condition	Downloaded file on the client device.

5.2. Design

In this section, we present the solution logical design approach for making backup and recovery using both GSM network and the internet to a thin cloud storage server, Microsoft Azure Storage. Here, we have focused on designing the class diagrams of the two applications, the deployment of this solution, server storage interface, which are presented below.

5.2.1. J2ME Class Diagram

The following diagram, as shown in the Figure 5-2 below represents that a class diagram for DatayeN J2ME application. The class diagram shows the logical implementation structure of the J2ME application used to build the application. The diagram below is represented by one main java class called DatayeN with its respective inner classes, namely Message, Phonebook, Event, and TodoList.

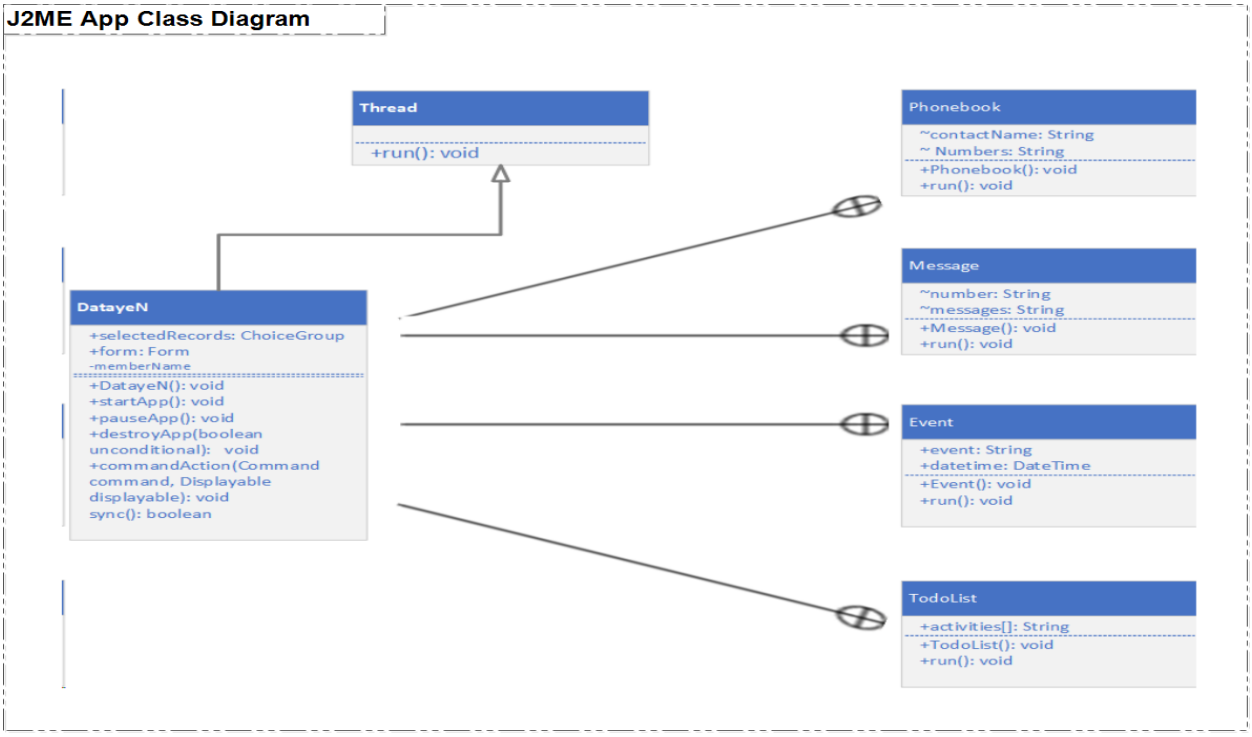

Figure 5-3: J2ME App Class Diagram

5.2.2. Android Class Diagram

The following diagram, as shown in the Figure 5-3 below represents that a class diagram for DatayeN android application. The class diagram shows the logical implementation structure of the Android application used to build the application.

The diagram below represents six main java/android classes with their respective inner classes, namely Message, Video, Contact, Music, Photo, and Document. Each Android class inherits shared features of the application from class BaseActivity, and each class refers Microsoft Azure Storage SDK classes (library files) to communicate with the Microsoft Azure Storage web service.

Figure 5-4: Android App Class Diagram

5.2.3. Deployment Diagram

The following diagram, as shown in the Figure 5-4 below represents that the deployment model of the solution, in which each component is installed in a specific and defined platform. According to the diagram, five different devices are needed for the installation and deployment of the system.

The first one is clients' feature phone which is going to be installed a java jar application called DatayeN.jar. This app is used for sending and storing device's contacts and message either to the peers' or to the Azure storage through LPMS and enables to recover user's contacts and messages from either storage device.

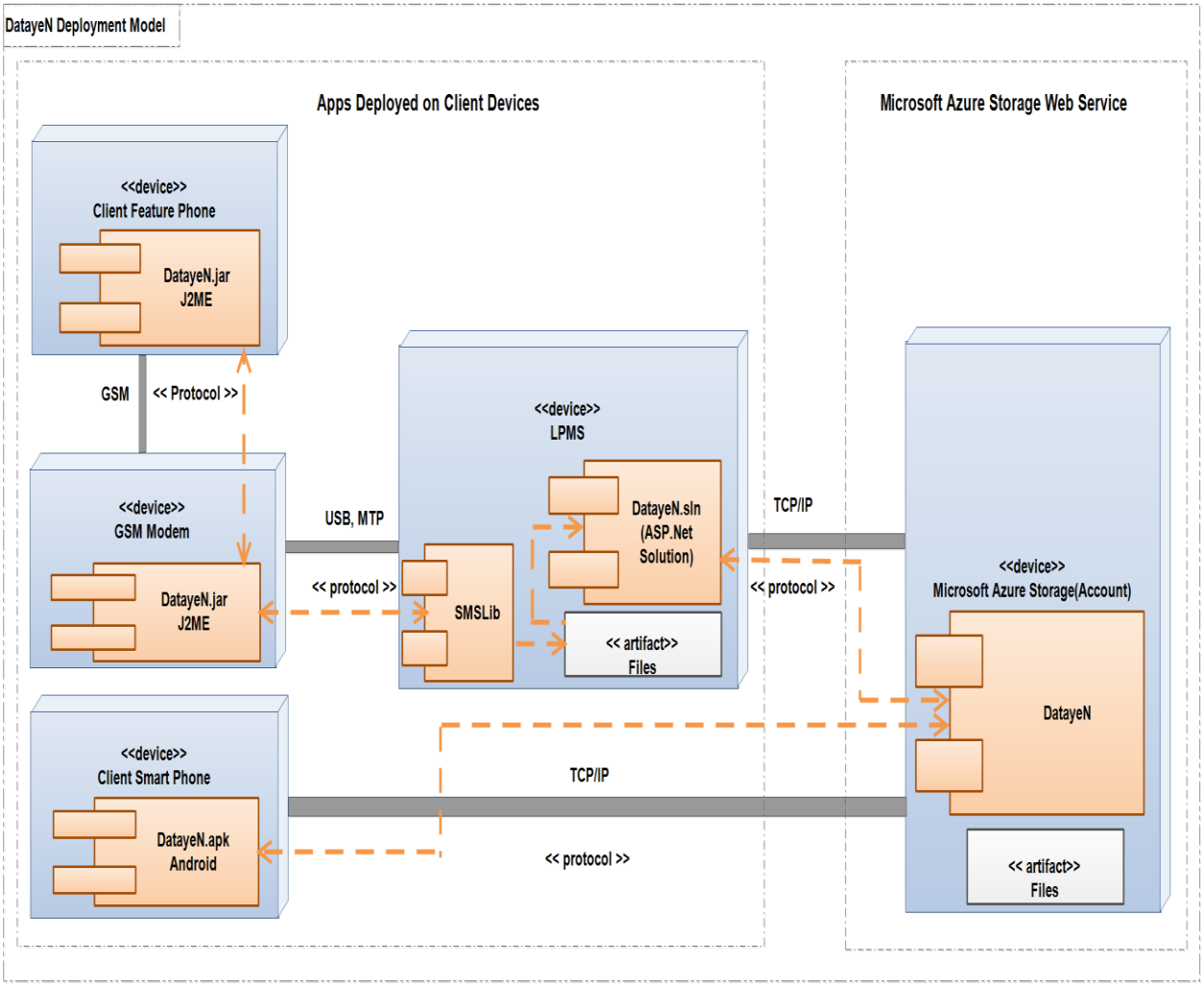

Figure 5-5: DatayeN Deployment Model

The second one is clients' smartphone which is going to be installed on an android application called DatayeN.apk. This app used for sending, storing and recovering device's data to/from Azure storage. The expected data to get pushed and recovered to/from Azure storage are contacts, messages, music, documents, pictures, and videos.

The third one is a GSM modem or device which accepts and manages cellular requests and forwards to the intended recipient. This device is going to be installed with a java jar app called DatayeN.jar. This app is used for accepting data through the cellular network to/from the LPMS, in which synchronizes with Azure storage. This app also performs compression and decompression of data. This modem is a connector, in between the feature phones and LPMS, like a middleman which transfers data back and forth.

The fourth one, is LPMS, a Local Partial Mirror Server, installed on windows operating system, running DatayeN.sln C# Project and SMSLib an SMS library which receives and extracts SMS contents, and sends SMS to the intended user. A server storage which stores users' only contact addresses and messages and enables feature phones to synchronize with Microsoft Azure Storage server. The LPMS, specifically the Datayen.sln project performs data de-duplication and synchronization with Azure storage, uploading users' contacts and messages, downloads and send to feature phones when required. This LPMS server is connected to GSM modem with USB, MTP protocol to connect with feature phones, and using TCP/IP to connect to Microsoft Azure storage.

The fifth one is Microsoft web service, called Microsoft Azure Storage, a storage account which enables cloud users to use storage webs service. According to the diagram above, inside the Microsoft Azure, we have created a storage account called DatayeN. This storage account, stores the users' contacts, music, pictures, documents, videos, and message. This component is only accessible through TCP/IP protocol.

To finally elaborate how this deployed component works, smartphones directly communicate with Azure storage for backup and recovery purpose, while feature phones need another GSM device to communicate with LPMS and then to the Azure Storage.

The detail explanation of the deployment diagram represented above in the Figure 5-4 is defined below based on two different endpoints.

- ***The feature phone endpoint:*** **-** feature phone is, users' endpoint for communicating with the Microsoft Azure storage. According to the diagram, Figure 5-4, the feature phone is installed with the J2ME app. This app enables users to store the data available in the device either on Microsoft Azure storage or on peer's device. The procedure of sending to the storage device is, the user selects required records (either contacts, or messages), and sends it using the GSM network to another receiver end which is GSM device(modem) which is installed in the J2ME app for decompressing records and connected directly with LPMS using USB, MTP protocol. Inside the LPMS server, there is a library call SMSLib which is used to extract SMS contents from the feature phone to LPMS storage device. And finally, the LPMS server performs de-duplication and synchronize with the Microsoft Azure storage service using TCP/IP protocol, the internet.
- ***The smartphone endpoint:*** **-** the device is also other users' endpoint for communicating with Microsoft Azure storage. According to the diagram, Figure 5-4, the phone device is installed with an android application which enables users to store the data available on Microsoft Azure storage. The data that can be pushed to the server is contacts, messages, pictures, videos, files, and music. The procedure of sending data to a storage device (Azure storage) is, users, select a specific record and opts to synchronize the data with Azure storage using TCP/IP protocol, the internet. Data de-duplication process is performed on the android phone.

5.2.4. Storage Server Interface

Microsoft Azure Blob storage is a web service that stores unstructured data in the cloud storage as objects/blobs. Blob storage can store any type of text or binary data, such as a document, media file, or application installer. Blob storage is also referred to as object storage. According to this solution, we have created a storage account called DatayeN, and containers and many blobs as shown in the Figure 5-5 below. Microsoft Azure Storage provides many different API interfaces for storage needing clients, applications, companies, enterprises. From the available interfaces, we picked the following interfaces for the simplicity our application communication with Microsoft Azure storage. The selected interfaces are described below:

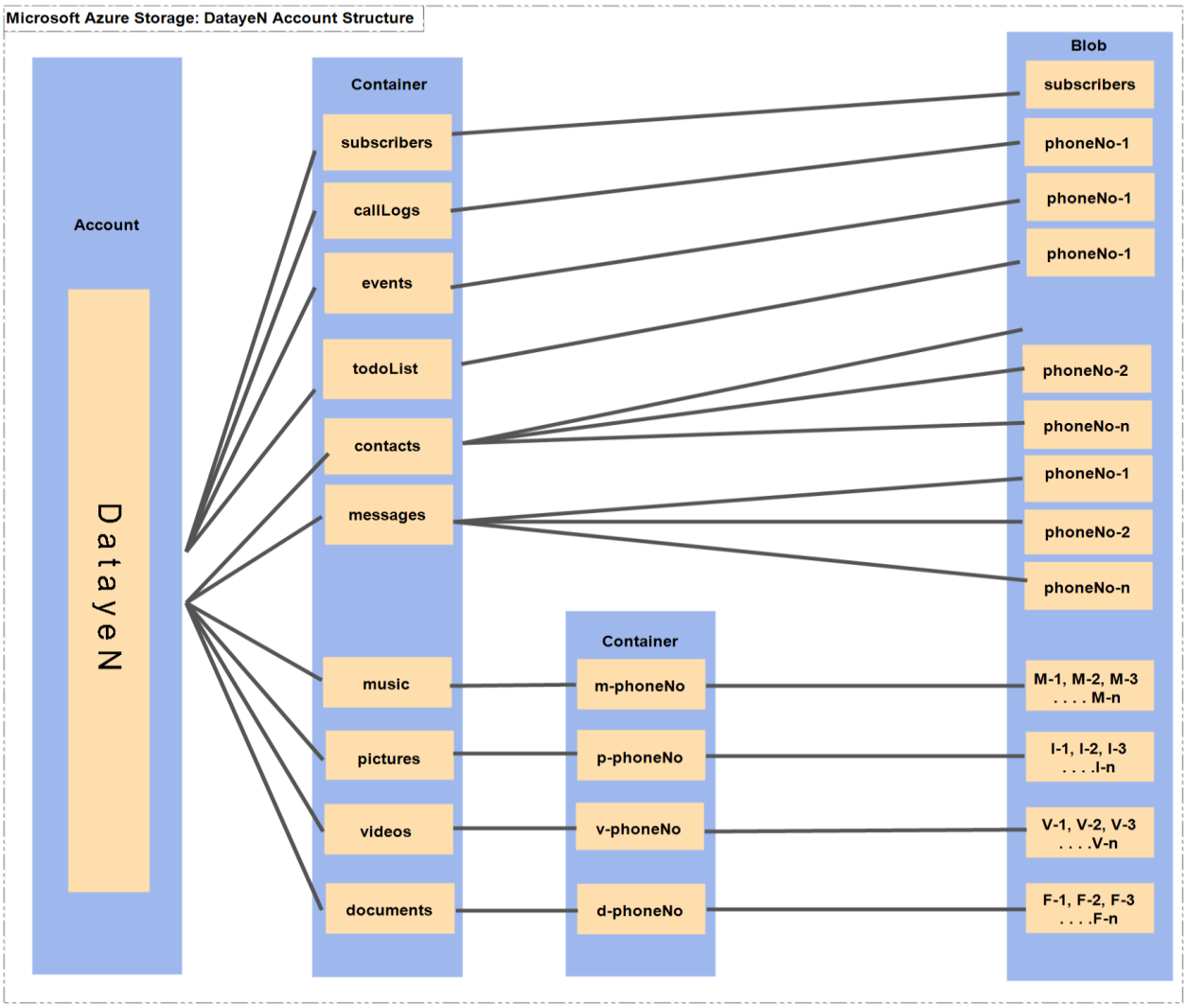

Figure 5-6: Microsoft Azure Storage Account Structure

- ***Download:*** - A Microsoft Azure storage endpoint or interface which enables clients to get or recover contents from the remote server.
- ***Upload:*** - A Microsoft Azure storage endpoint or interface which enables clients to store or push a file to the server.
- ***Search:*** - A Microsoft Azure storage endpoint or interface which enables client's app to search objects or blobs from the remote server.
- ***Delete:*** - A Microsoft Azure storage endpoint or interface which enables client's app to remove objects or blobs from a specific container of the remote server.

The detail explanation of the Microsoft Azure storage account structure diagram represented above in the Figure 5-5 is defined and explained below.

To describe the above diagram from left to right, the box identified with label DatayeN is a storage account in Microsoft Azure in which all clients of this solution store file. Inside this storage account, there are six containers or folders which store the actual user's data objects or blobs. The six containers are Contact, Message, Music, Picture, Video, and Document.

The container Contact is used for storing clients' contacts. This container stores contact blobs saved by the name of client's contact address, the phone number. The process of storing client's contacts addresses is that, when a user clicks a sync button on the user device, the system finds the storage account called DatayeN then again finds Contact container and finally stores the actual object by the name of phone number requested.

The container Message is used for storing clients' messages. This container stores message blobs saved by the name of clients' contact address, the phone number. The process of storing client's messages is that, when a user clicks a sync button on the user device, the system finds the storage account called DatayeN then again finds Message container and finally stores the actual object by the name of phone number requested.

The container Music is used for storing clients' music. This container contains other containers named by the corresponding clients' phone number that can store music blobs saved by music actual name. The process of storing client's music is that, when a user clicks a sync button on the user device, the system finds the storage account called DatayeN then again finds Music and phone specific container and finally stores the actual object.

The container Picture is used for storing clients' pictures. This container contains other containers named by the corresponding clients' phone number that can store picture blobs saved by image actual name. The process of storing client's picture is that, when a user clicks a sync button on the user device, the system finds the storage account called DatayeN then again finds Picture and phone specific container and finally stores the actual object. In this container, a container only for specific client stores multiple blobs called pictures. These pictures found in this container belongs to the specified client. This implies that every client has their own Picture container.

The container Video is used for storing clients' videos. This container contains other containers named by the corresponding clients' phone number that can store video blobs saved by video actual name. The process of storing client's video is that, when a user clicks a sync button on the user device, the system finds the storage account called DatayeN then again finds Video and phone specific container and finally stores the actual object. In this container, a container only for specific client stores multiple blobs called videos. These videos found in this container belongs to the specified client. This implies that every client has their own Videos container.

The container Document is used for storing clients' documents. This container contains other containers named by the corresponding clients' phone number that can store documents or files blobs saved by document actual name. The process of storing client's document is that, when a user clicks a sync button on the user device, the system finds the storage account called DatayeN then again finds Document and phone specific container and finally stores the actual object. In this container, a container only for specific client stores multiple blobs, called documents. These documents found in this container belongs to the specified client. This implies that every client has their own Document container.

5.3. Programming Language and Operating System Used

5.3.1. Android

Android is a mobile operating system which is found on a variety of modern devices, the most popular being smartphones. This operating system enables Android applications to access devices functionality and performs required functionality. According to the survey we took, 60% of the participants were using Android mobile devices, 2% of the participants were iPhone users, windows phone users were also 2%, and the rest of users were feature phone users. Based on the survey result most of the users are Android phone users and comes second feature phone users. In total 96% the participants are included for use in this solution.

5.3.2. J2ME

Java is an object-oriented programming language designed for use in the distributed environment of the internet. It is the most popular computer programming language for the development of Android smartphone applications and is among the most favored for edge device and internet of things development. Java provides three key platform editions, Java SE(J2SE) for standalone applications, Java EE(J2EE) for client-server components and internet applications, and Java ME(J2ME) for embedded applications.

J2ME is a java micro-edition for small devices, such as devices with low processing capability, less memory, and low performance. Feature phones are devices with that above-listed constraint. According to our survey from the participants' feature phone types and in the mobile market, the feature phones available are Nokia, Samsung, Techno, Blackberry, Oking and Smadle phones. These above-listed phones support java.

5.3.3. C#

C-Sharp(C#) is an object-oriented programming language provided by Microsoft where developers can build a variety of secure and robust applications that run on the .NET Framework. With the help of C# programming language, developers can develop windows applications, web applications, distributed applications, web service applications database applications. In related to our solution, the C# project is designed to work with Microsoft web service storage called Microsoft Azure Storage **[29]**.

5.4. Algorithms

5.4.1. Backup

As we have noted above in chapter one, backup is the process of storing a copy of data inside the secondary device so that it will get recovered whenever it is needed. The diagram shown in the Figure 5-6 below represents the flow of activities that are required to perform backing up data from the user device to the remote web server storage, Microsoft Azure Storage or to the peer device.

As shown in the diagram below, the backup algorithm initiated by the end user to communicate with secondary storage either with peer storage (specifically for feature phones) or with Microsoft Azure Storage.

The backup algorithm is described below: -

1. A user connect internet with a remote server.
2. The system checks whether the user is already subscribed or not.
3. Arranges local device phone records, go to step 10 if the user is a new subscriber.
4. If the user is already subscribed go to step 5, otherwise register the user and go to 10.
5. The server downloads the intended file to the client device.
6. Client device splits the file into blocks.
7. Check if the record is empty, if Yes go to end 10 else go to step 8.
8. Check if the record exists in the local client device as a record, if **Yes**, save record existence frequency increase record index and go to step 7.
9. Write a record to file, increase record index and go to step 7.
10. Save and prepare local record to a file.
11. Push file.

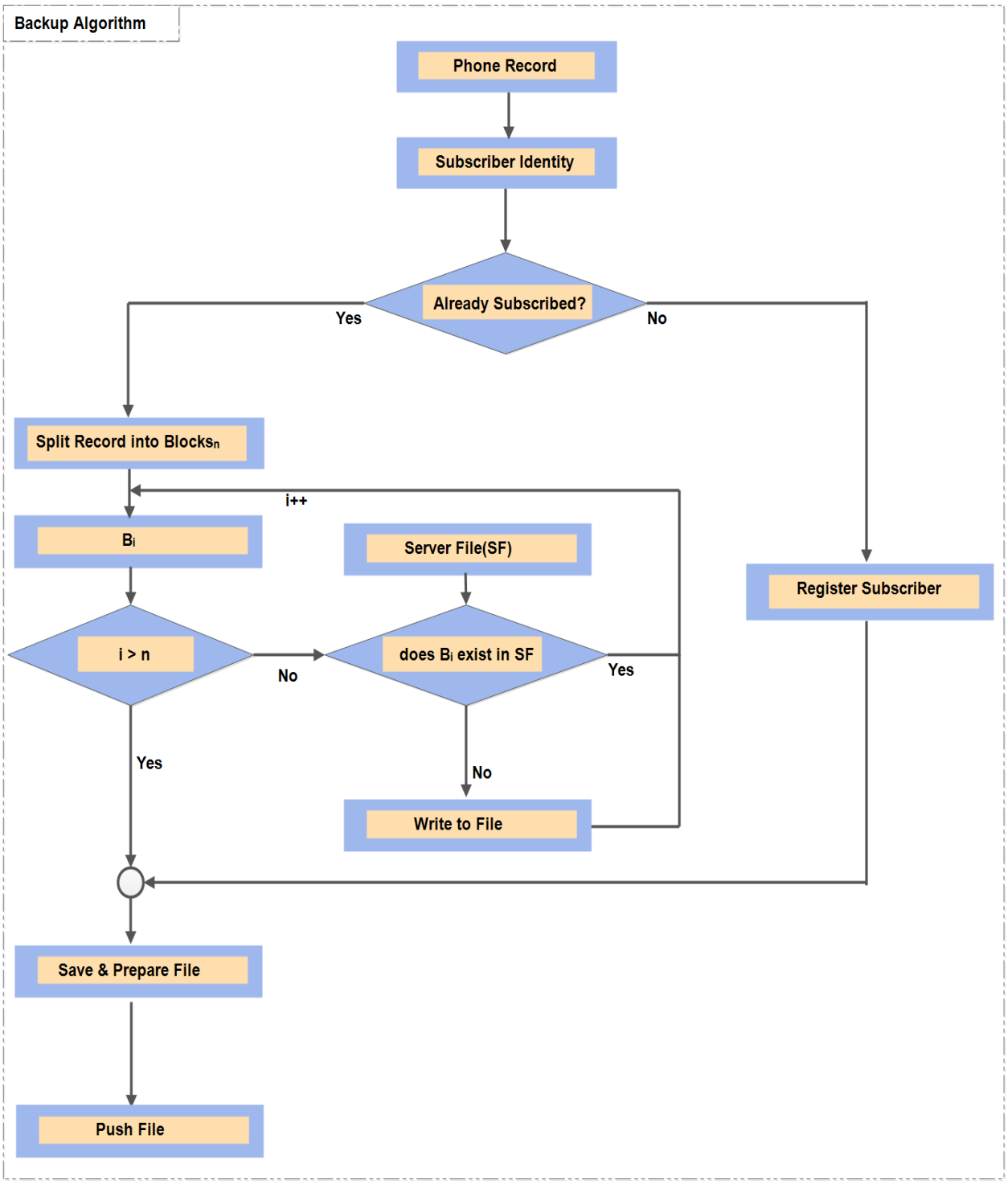

Figure 5-7: Backup Algorithm

5.4.2. Recovery

As we have noted above in chapter one, recovery is the process of getting back previous data and restoring back from the backup secondary storage, either form Microsoft Azure Storage or peer's device. The diagram shown in the Figure 5-7 below represents the flow of activities that are required to perform recovering back data from the secondary storage.

This recovery process is allowed only for those who subscribed before and wants to restore it back to their local device. The user has the opportunity to restore the data backed up before, either from the peer storage or Microsoft Azure Storage, the same as to the option provided while making backing up.

Like represented the process of recovering in the above diagram, the algorithm is: -

1. A user connects internet with a remote server.
2. The system checks whether the user is already subscribed or not. If the user is not subscribed yet, instruct the user to subscribe.
3. The system arranges local device phone record.
4. The server downloads the intended file to the client device.
5. Client device splits the file into records.
6. Check if the record is empty, if Yes go to end 10 else go to step 8.
7. Check if the record exists in the local client device as a record, if **Yes**, save record existence frequency increase record index and go to step 7.
8. Write a record to file, increase record index and go to step 7.
9. Save and prepare the record for the drive.

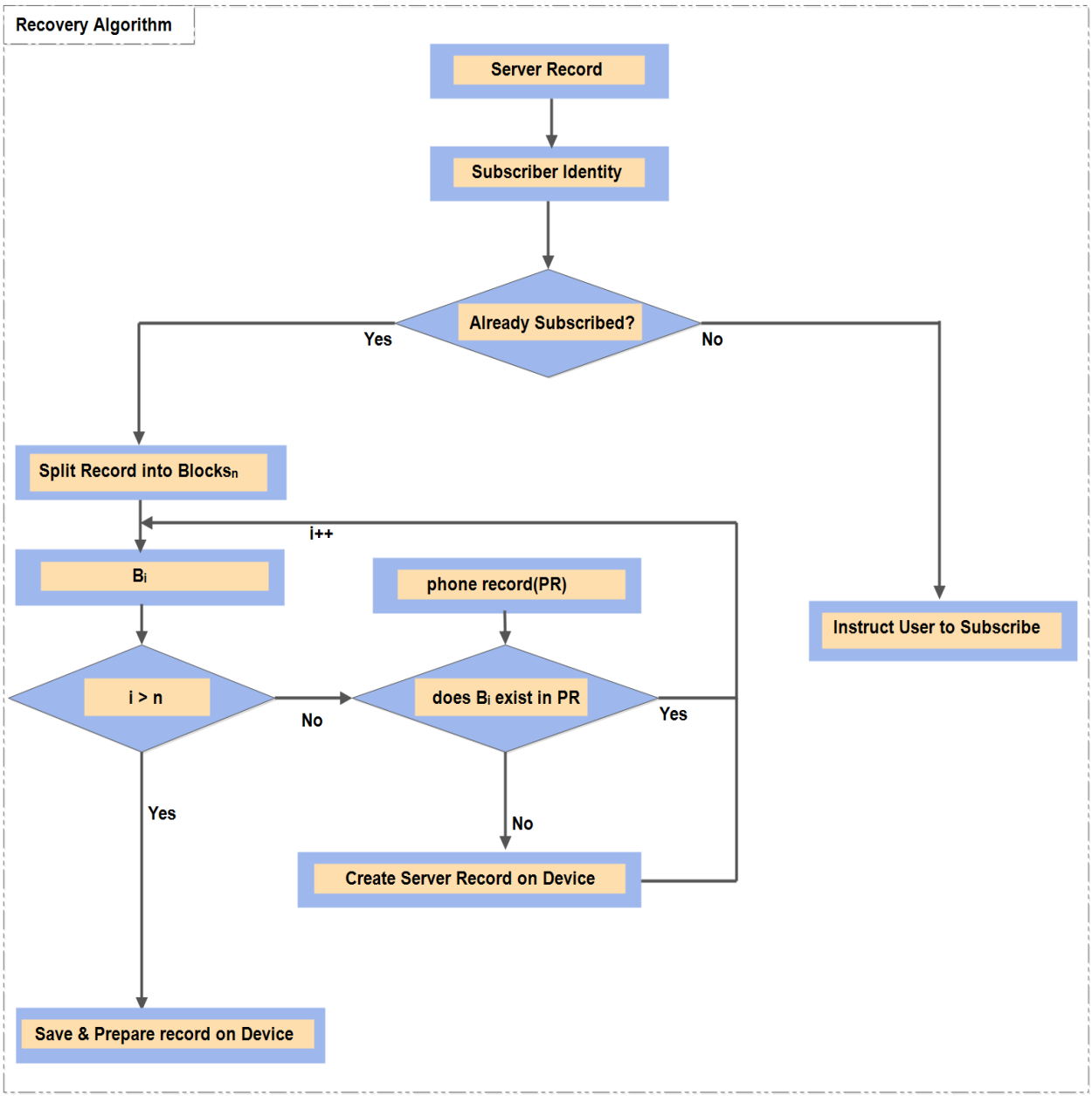

Figure 5-8: Recovery Algorithm

5.4.3. Compression and Decompression

Communication is the process of connecting and interacting sender with the receiver. In information technology era, communication is based on transmitting data or information from sender to receiver. The communication process should be accurate, on-time, effective and efficient. Compression and decompression using Huffman encoding and decoding to support a quality communication between sender and receiver.

Compression is a process of converting information's bits into equal representation but with smaller bits. The main objective of compression is to reduce the size of the information so that it can be transmitted in a shorter period of time and takes a smaller size than the actual data inside storage device. To utilize the advantage of compression, decompression should be available at the other end.

Decompression is a process of converting back the compressed data to its original content so that it can readable by end users. While the compression process is performed at the sender end, the decompression process is performed at the receiver end. Regarding our research, the compression technique is needed for representing more characters while users sending SMS messages to the receiver.

Huffman encoding is a lossless data encoding algorithm, information compression, and decompression algorithm. This algorithm is used in our research to increase the number of characters to get sent using SMS in GSM network. The algorithm is implemented using Java, J2ME app which is going to get installed on feature phones. The compression process is applied at the sender side(compressing the number of characters to be sent, either a list of contacts or a list messages), and the decompression process is performed at the receiver end, at LPMS server attached GSM modem or device if it is sent to the cloud storage otherwise it remains compressed because it is stored inside peer's device and must be hidden from unauthorized user.

The process of Huffman compression is that a user of feature phone selects specific record contacts or messages when they tried to push to the storage service, the client device compresses the selected record and pushes to the storage.

5.4.4. De-duplication

Data de-duplication is a process of removing duplicated data to ensure data uniqueness, minimizes storage usage, bandwidth utilization, and environment protection by reducing needed for processing more duplicated data [1]. A unique data backup in a cloud storage can be maintained using the data de-duplication technology, which enables a unique copy of data on the cloud storage server. It states that data de-duplication is a data reduction process and technique that will be stored as part of data backup [20].

Regarding this research, we have decided to perform data de-duplication as shown in the Figure 5-8 below, on users' contacts and messages to remove duplicated contact information, and message. This process is only performed for those who were using before and tried to synchronize. In our research point of view, Microsoft Azure storage web service is a thin cloud in which it doesn't have to manipulate tasks except storing clients' data so that clients' device performs other tasks, like de-duplication process, is performed at the client device.

The de-duplication algorithm is: -

1. Users try to connect internet from the client device.
2. The server downloads the intended file to the client device.
3. Client device splits the file into records.
4. Check if the record is empty, if Yes go to end 7 else go to step 5.
5. Check if the record exists in the local client device as a record, if **Yes**, save record existence frequency increase record index and go to step 4.
6. Write a record to file, increase record index and go to step 4.
7. Save the file and push to Microsoft Azure Storage.

In this solution, de-duplication is performed in two ways for those who have subscribed already, tried to sync with the server. Based on the solution we are providing, there are two different platforms that are going to use the backup and recovery service, feature phones through GSM network, and Android phones through the internet. According to the platforms available, feature

phones' record de-duplication process is performed on LPMS server, implemented using C# project. This C# project tries to remove duplicated records, registers their existence frequency, and finally uploads to the server, and Android phones' record de-duplication is performed on each Android phone as their performance much better than the feature phones.

According to our solution, the de-duplication process is performed only on device's phone contact addresses, and messages as the records are easy to detect duplicated information or record while performing backup and doesn't require a special performance or hardware requirement.

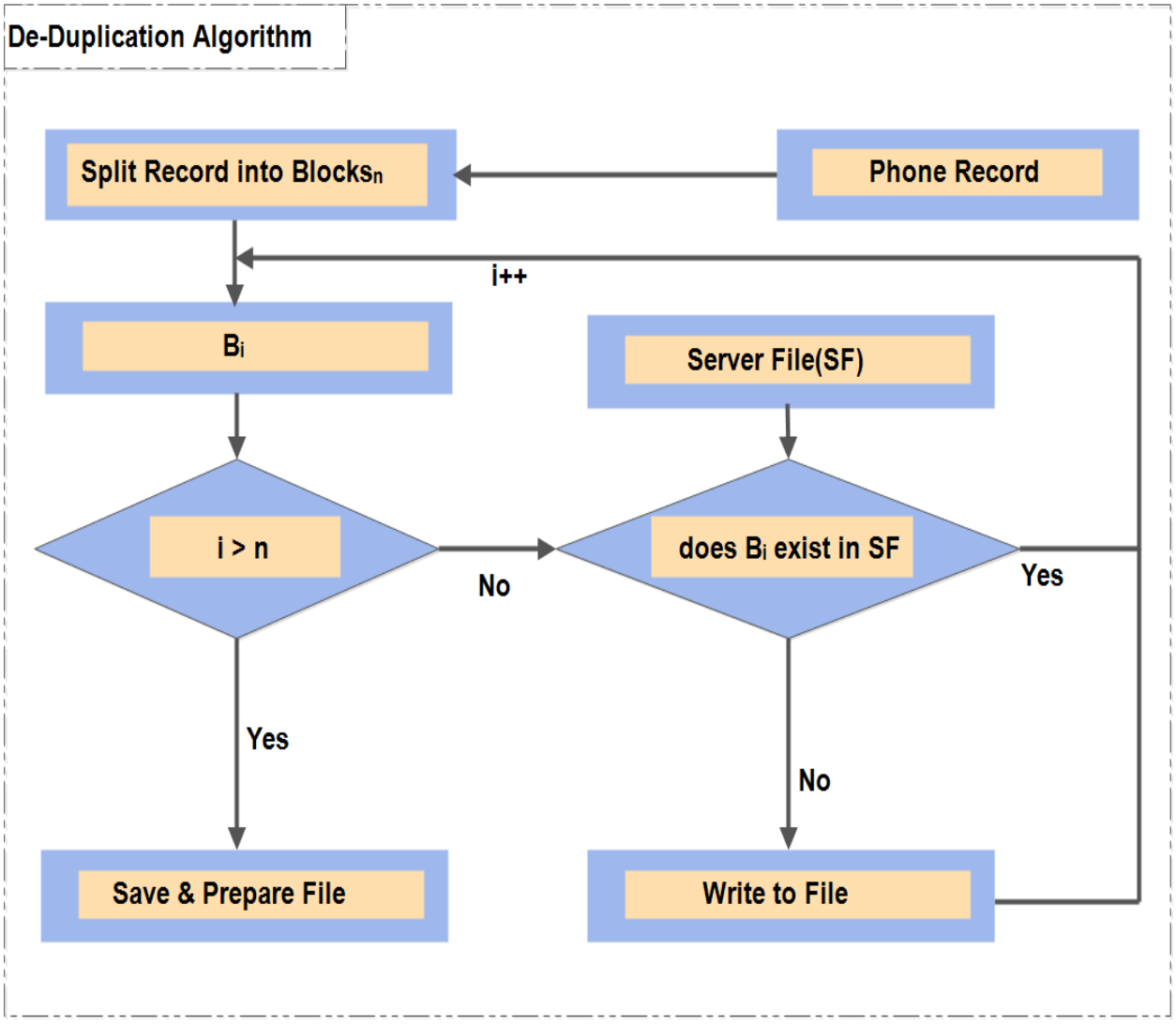

Figure 5-9: De-Duplication Algorithm

5.5. Tools Used for Development

NetBeans IDE 6.9: **-** is an integrated development environment (IDE) for Java, C, C++ and another programming language. But Specifically, in this study, it is used for developing the J2ME application.

Wireless Toolkit 2.5.2: **-** a toolkit which enables J2ME apps to communicate wireless with the external environment.

Java 2 Micro Edition 3.0: **-** a minimized version of Java SE for a small device, with limited resources, such as processor and memory.

Android Studio IDE 2.3: **-** a development environment for Android application, which includes other necessary libraries for the app development to minimize developer's effort for configuration.

Microsoft Azure Storage SDK: **-** a library package which exposes Microsoft Azure Storage web services API. Specific to this solution, Microsoft Azure Storage Android SDK version 2.0.0 is needed for communicating Android application with Microsoft Azure Storage, and Microsoft Windows Azure Storage version 9.1.1.0 is needed for communicating LPMS with Microsoft Azure Storage.

Microsoft Visual Studio 2015: **-** is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. It supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) in nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI, Visual Basic.NET, C#, JavaScript, TypeScript, XML, XSLT, HTML and CSS. But in this study, it is used for writing C# codes that can communicate with the web service [29].

SimpleUML: **-** android studio IDE plugin, used for generating UML class diagram and their relationship with each other from java classes of the project or application.

Microsoft Visio 2016: **-** Microsoft Software for creating UML class diagrams and their relationship with each other.

Chapter 6 : EVALUATION, RESULTS, AND DISCUSSION

8.1. Result

In this chapter, we discussed the results and the implication of this study final result, and explained this study solution usage scenario.

8.1.1. Result Usage Scenario

The final product of the study is that to provide a solution which answers the above-defined research questions and users need based on the survey answer. Thus, a software solution has developed as result. In this section, we are going to discuss solution usage and use scenario.

8.1.1.1. Scenario 1

In this scenario, both client's device is a feature phone, as shown in the Figure 6-1 below. If for the first time, an uploading client should subscribe to the system to use the backup and recovery service. After the subscription is succeeded, the client can upload device's contacts and messages to the server. Then, whenever the device is lost and replaced with another feature phone, the already uploaded data can be recovered with the same mobile phone device category.

N.B: - in this scenario, data backing up and recovering only user's contacts and messages.

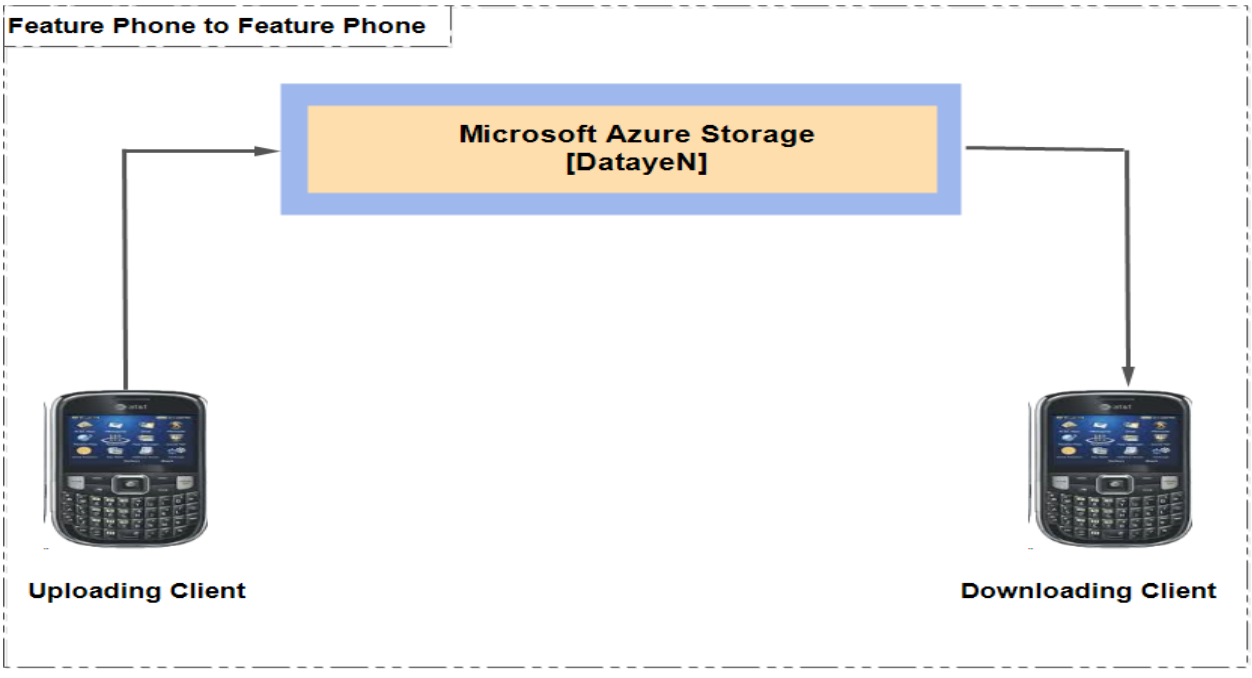

Figure 6-1: Feature Phone - Feature Phone BR

8.1.1.2. Scenario 2

In this scenario, the mobile phone devices' category is different, as shown in the Figure 6-2 below in which the uploading device client is feature phone while the downloading client device is a smartphone. If for the first time, an uploading client should subscribe to the system to use the backup and recovery service. After the subscription is succeeded, the client can upload device's contacts and messages to the server. Then, whenever the device is lost and replaced with a smartphone, the already uploaded data can be recovered using the replaced smartphone, which is with the same phone number using the lost device number.

N.B: - in this scenario, the data type that can be backed up using the feature phone is user's contacts and messages, and data types that be recovered using the smartphone is user's contacts, messages, music, videos, pictures and files if any.

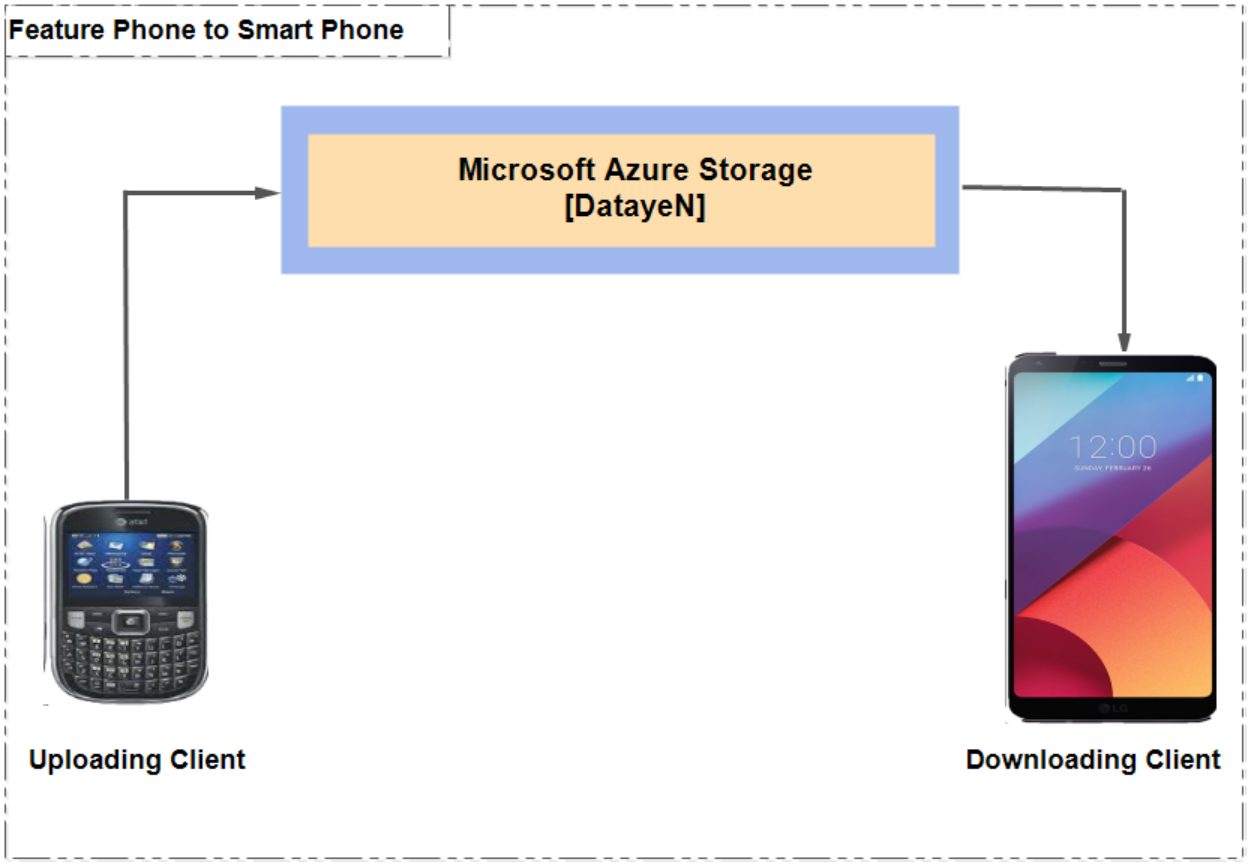

Figure 6-2: Feature Phone - Smart Phone BR

8.1.1.3. Scenario 3

In this scenario, the mobile phone devices' category is different, as shown in the Figure 6-3 below, in which the uploading device client is smartphone while the downloading client device is a feature phone. If for the first time, an uploading client should subscribe to the system to use the backup and recovery service. After the subscription is succeeded, the client can upload device's contacts, messages, music, videos, pictures, and files if any to the server. Then, whenever the device is lost and replaced with a feature phone, the already uploaded data can be recovered using the replaced feature phone, which is with the same phone number with the lost device number but only contacts and messages.

N.B: - in this scenario, the data type that can be backed up using the smartphone is user's contacts, messages, music, videos, pictures, and files if any, and data types that be recovered using the feature phone is user's contacts and messages.

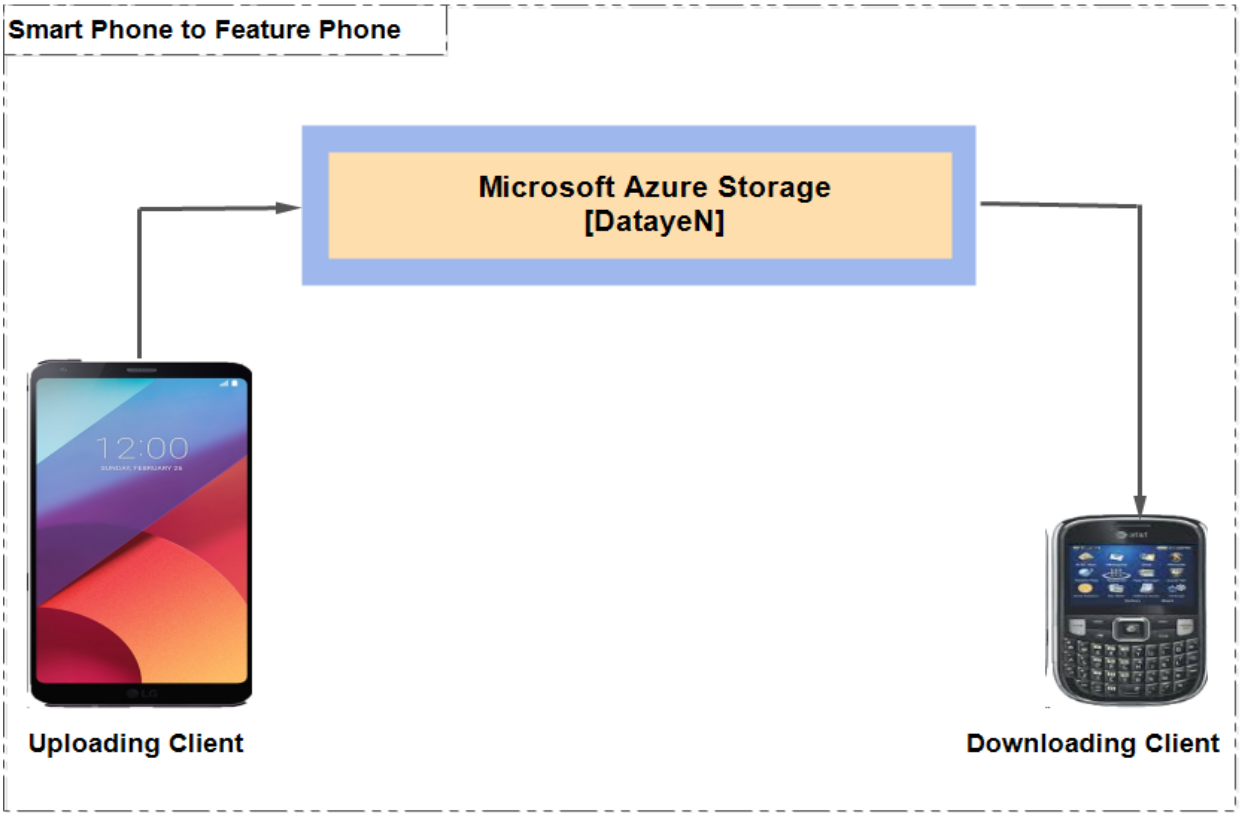

Figure 6-3: Smart Phone - Feature Phone BR

8.1.1.4. Scenario 4

In this scenario, both client's mobile device is a smartphone as shown in the Figure 6-4 below. If for the first time, an uploading client should subscribe to the system to use the backup and recovery service. After the subscription is succeeded, the client can upload mobile device's contacts, messages, pictures, music, videos and files to the server. Then, whenever the device is lost and replaced with another smartphone, the already uploaded data can be recovered with the replaced smartphone with the same phone number using the lost device number.

N.B: - in this scenario, the data type that can be backed up and recovered using the smartphone is user's contacts, messages, pictures, music, videos, and files.

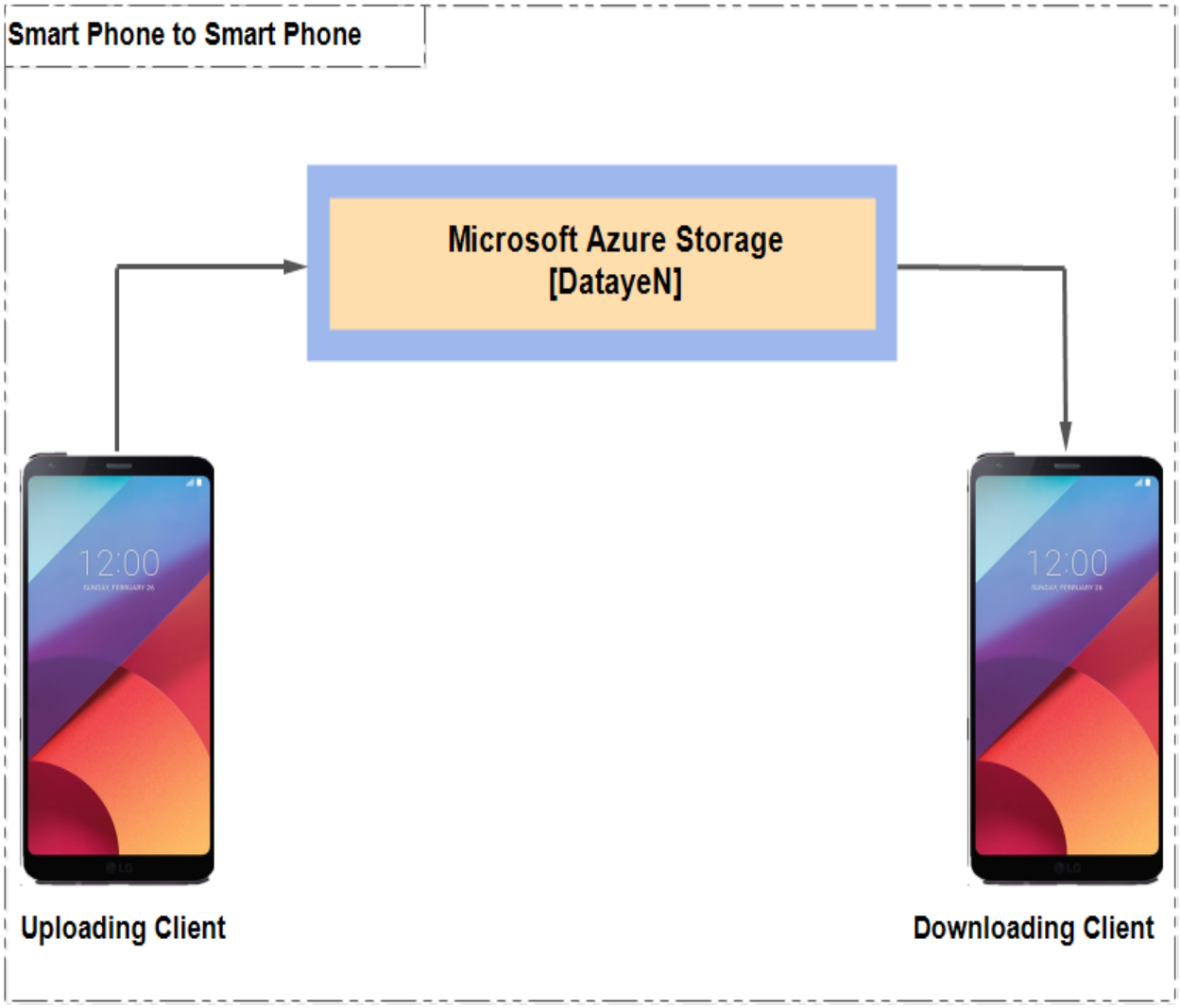

Figure 6-4: Smart Phone - Smart Phone BR

8.1.1.5. Scenario 5

In this scenario, both client's device is a feature phone, as shown in the Figure 6-5 below. The client can upload/push device's contacts and messages to the peer's mobile device only using Ethio-Telecom GSM network. The pushed or uploaded data of uploading user resides on peer's device. Then, whenever the device is lost and replaced with another feature phone, the already uploaded data to the peer device can be recovered with the same mobile phone device category, which is a feature phone with the same phone number with the lost device number.

N.B: - in this scenario, the data type that can be backed up and recovered using the feature phone is user's contacts and messages.

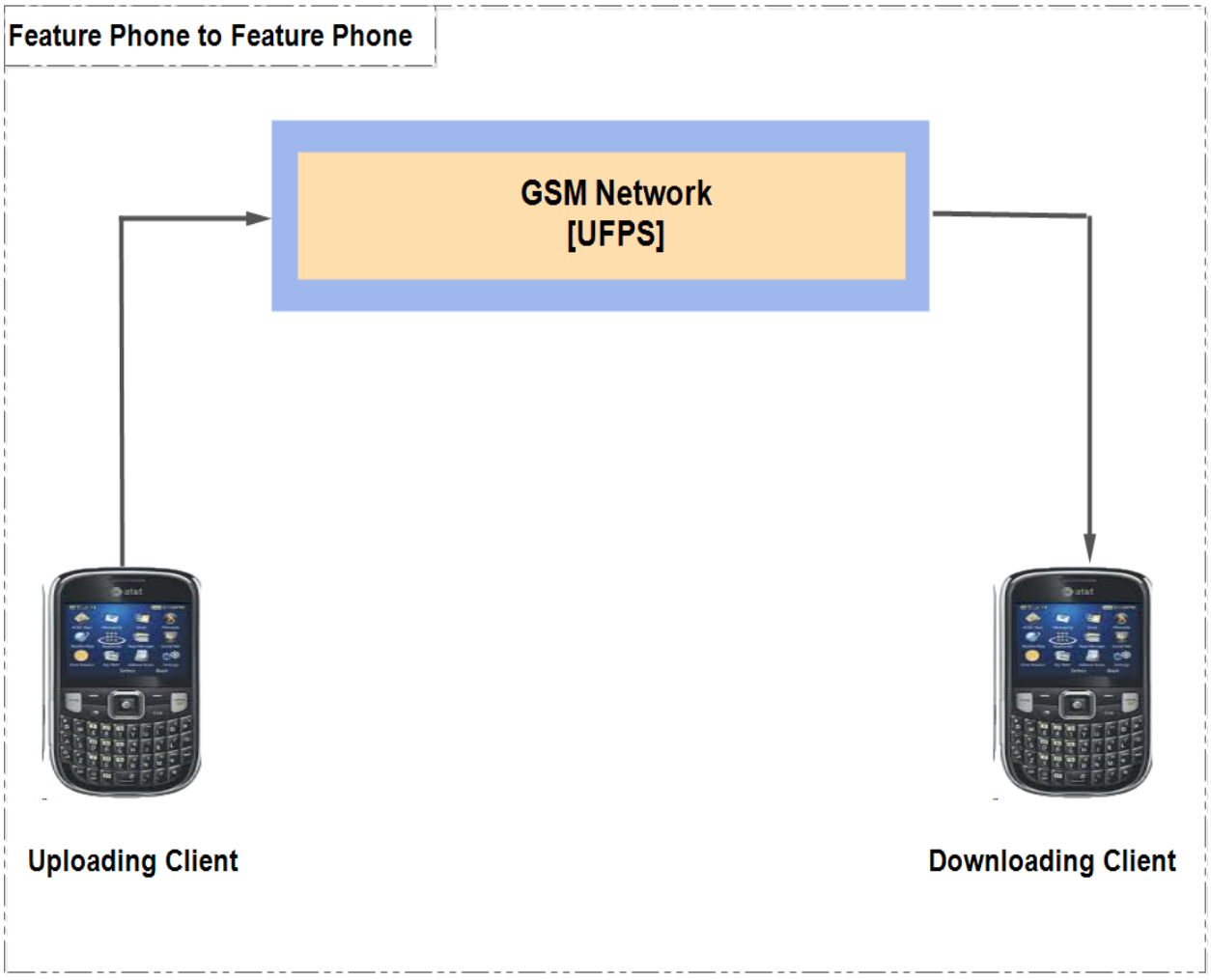

Figure 6-5: Feature Phone - Feature Phone P2P BR

8.2. Discussion

According to the collected survey, the participants' responded to the need for data backup and recovery, and other related questions summarized to answer the research questions represented below.

RQ1. What are the possible mechanisms for a feature phone to communicate with a web service, cloud storage, and enable those devices to store a backup data in the cloud storage, recover the data back from the cloud storage?

- In existing solution, feature phones were unable to communicate with a web service, but now in this study, we have developed system J2ME app which performs GSM request and receives GSM request and compresses the SMS characters to increase the characters by 28%. This GSM request connects to the web service through LPMS which is developed in C# and integrated with SMSLib that can send SMS and read mobile phone SMS messages. This LPMS server performs both GSM request an HTTP request.

RQ2. How to enable data portability among different mobile phones platform, especially in between feature phones and smartphones through the cloud?

- In existing solution, moving data like contacts, and messages from one platform to another platform is impossible as the file format different differs from one to another. This consideration is only available if the data holding device is available. In this study, we have implemented that a data that is stored in the cloud storage is easily readable in either feature phones or smartphones platforms.

RQ3. What are the possible compression algorithms that can increase the number of characters of that can be sent using GSM SMS service?

- The number of characters that are going to be represented in one SMS message is 140 characters. As we already defined that this study uses the SMS service to send a selected record through GSM network to our LPMS server. Therefore, finding a solution for increasing the number of SMS characters was our primary objective so that we discovered the SMS characters in one SMS message can be uplifted to 28% using the Huffman encoding.

The following questions are some filtered questions asked of the participants of this survey, and their answer is analyzed and written below each question

- ***RQ4.*** In what way will get influenced users if they lost data found on their mobile?

  Answer: - the participant's response was, 86% of the participants rated that they will get highly affected if they lost the contact addresses found on their mobile device, 14% of them said its impact is medium. 68% of the participants replied that losing images found on their mobile devices affects them highly, and 50% of the participants said losing files and documents affects them highly, while 45% of the participants replied the impact is medium. 59% of the participants rated losing music found on their mobile phone device has low impact, 55% of the participants replied that call logs have low impact, while 41% of the participants replied losing video has low impact. 27% of the participants have replied losing messages affects highly.
- ***RQ5***. How many users are using cloud computing back as a service?

  Answer: - the participant's response was, 17% of the participants do not know about cloud storage, 52% do not use cloud storage service, in total 69% of the participant do not use cloud storage, and 26% of the participants use cloud storage.
- ***RQ6***. How much need exist for local repository data storage of mobile data?

  Answer: - the participant's response was, 95% of them replied backup and recovery solution using the internet is good but fails to address feature phones while 100% of the participants replied that data backup and recovery solution addresses all mobile users.
- ***RQ7***. To what extent is the data valued found on mobile devices by users?

  Answer: - the participants response was, 72% of the participants describes contact addresses found in their mobile device is the most and highly important digital information (ranked first), 33% of the participants said pictures found on their mobile phones their third highly valuable and most important data(ranked third), 20% of the participants described as file is the most valuable and important data. Overall contact addresses, files, and pictures are highly valuable.
- ***RQ8***. What is the perception of mobile users towards cloud computing, cloud backup storage and backup as service in a cloud?

Answer: - the participant's response was, 52% of the participants was replied that they didn't know what backup and cloud is, but after the survey, all participants were aware of the cloud service, the need of backup and recovery.

- ***RQ6***. What would be mobile users thought and opinion on using a local cloud?
  Answer: - the participant's response was, regarding using local cloud repository, they said would be great, as the repository will be responsible, near to users or clients.

As we have stated in chapter 2, most of the study focuses smartphone backup and recovery solution in the cloud storage, while some study focuses on manual feature phone backup and recovery solution which needs additional cost for extra secondary storage and software configuration. This feature phone backup and recovery solution need always human interaction to backup to external storage and same recover it, which is tiresome, and may not be the latest record while recovering when the primary storage accidentally lost, and also the secondary storage is prone to error as it in the hand of the owner.

This study enables users to store data available on their mobile devices either using feature phone or smartphone in Microsoft Azure storage and can recover it back from either mobile device. As the backup data is in a cloud storage, it is accessible whenever, and wherever the requests it.

Our solution for the feature phone is different from literature solution for feature phone provided which requires additional hardware request and software configurations. First, our solution doesn't need any additional resources except the phone which is in use by the user. Second, our solution provides data portability opportunity in which the records on the cloud storage are readable by each platform, the feature phone, and smartphone so that they can be represented easily on any of the platforms. And one more feature we have created is that a user of feature phone can store the records found on the device on peer's feature phone device so that the owner of the data can recover it back.

Chapter 7 : CONCLUSION AND FUTURE WORK

7.1. Contribution

This study contributes a design of framework for backup and recovery service that supports both feature and smart mobile phone to the existing knowledge.

Characteristics	This Solution	Existing Solutions
Supports smartphone	Yes	Yes
Sync feature phone with cloud	Yes	No
Requires no extra hardware and software for feature phone data backup and recovery(not expected from individual clients)	Yes	No, for some
Supports data portability in between platforms through the cloud storage	Yes	No
P2P BR	Yes	No

7.2. Conclusion

Cloud computing is an efficient and growing fast information technology infrastructure utilization technology that can provide many services through the web, such as storage service, software service, database service and many more services. Cloud storage service is a service which used to store customer's data so that it will get accessed everywhere and also used for backup and recovery of customer's data. Backup and recovery are the most important thing in information system management. The lack of data portability among platforms, in between feature and smartphone, and a lack of backup and recovery solution for a feature phone in the cloud or the need of providing costly hardware and software configuration make it difficult users of feature phone to recover lost information.

Most of the research papers we have reviewed focus on providing data backup and recovery service for only smartphones while some papers focus on feature phones which is manual backup and recovery service to/from the owner external storage or secondary device. The aim of this study is to explore and use web service storage for two different mobile phone platforms, feature phones, and smartphones and synchronize the data found on the devices and store in the cloud so that it can be recovered from either device. In this study, data collection finding identifies that first some of the feature phone users were using a manual backup, implying that they use exporting the data found on their device to secondary storage which makes it tiresome and the backed-up information is not updated and not recent, second moving data from feature phone to smartphone or vice versa was difficult, and thirdly according to our data collection the need backup and recovery solution which considers the above two listed fining was highly rated.

In this study, we designed and implemented two different applications J2ME app called *DatayeN.Jar* and android app called DatayeN.apk for feature phones and smartphones respectively, and a dot Net project(LPMS installed with SMSLib, Microsoft Windows Azure Storage SDK) which manages feature phones request and connects feature phones to the web service storage called Microsoft Azure storage. The feature phones send a request to the GSM device installed with DatayeN.jar app which is directly connected to LPMS using GSM network. The android application directly connects using internet with the cloud storage, Microsoft Azure storage and synchronizes the data found in the mobile device per the owner's request. In this solution, we have added a feature the supports user feature can store the backup data on peers' device(P2P BR) so that can recover it back. This study has provided insight that feature phones can be connected with web service and store data on cloud storage, and the data can be accessed on either platform, from a feature phone or smartphone, which is called data portability.

Finally, during this study, we have explored the cloud computing technology, the services provided by the cloud computing. We have gained a great experience working the J2ME technology, exploring the specification of feature phones, understanding using the Huffman encoding a compression and decompression algorithm for optimizing the number of characters to be sent using the GSM network, and working with the Android environment. We have explored and compared many cloud storage service providers, explored the dot Net framework and the

cloud storage integration, and finally, we have gained a great experience on how to work research.

7.3. Future Work

In this study, we have developed a solution for both platforms, feature phone and smartphone, which synchronizes the data available on the devices to the cloud storage services, Microsoft Azure storage. Due to scope, and the feature phones capability, the proposed approach works only by pushing the actual data to the cloud storage without any encryption and decryption. However, to provide a more realistic solution the following points are needed to be addressed in the future but, other students and researchers also can explore one or more of these topics.

- In the near future, we will implement a new algorithm for feature phone data compression as the Huffman encoding optimizes only by 28% characters.
- We will explore a better solution for data encryption and decryption which considers feature phones capability.
- We will work on data de-duplication as data de-duplication is important for managing big data.

REFERENCES

[1] Michael Vrable, Stefan Savage, and Geoffrey M. Voelker. Cumulus: Filesystem Backup to the Cloud.

[2] Guo-Zi Sun, Yu Dong, Dan-Wei Chen, and Jie Wei, Data backup and recovery based on data de-duplication, International Conference on Artificial Intelligence and Computational Intelligence, 2010.

[3] Debabrata Sarddar and Rajesh Bose. A Mobile Cloud Computing Architecture with Easy Resource Sharing", Vol 4, No 3, June 2014.

[4] MS. Shriwas. A Proficient Scheme for Backup and Restore Data in Android for Mobile Devices, Vol.1, Issue.1, October 2013.

[5] MCIT, E-Systems Africa Ethiopia, Cloud Computing Strategy and Roadmap Project Readiness Assessment, May 2014.

[6] Ch.anilkumar and Dr.R.Kirshnaiah. A Novel approach for unique Data Backup in Cloud Storage, Vol.2, No.4, July 2013.

[7] GE Liang, YANG Junduo, ZHU Qingsheng, LV Kang, WANG Qianqian, LUO Xin, Cloud backup Scheduling Algorithm Based on Cloud State Table and Cloud Resources Table, Ninth Web Information Systems and Applications Conference, 2012, China.

[8] Jing Nie, Design the Desktop Backup System Based on Cloud Computing, Eighth International Conference on Computational Intelligence and Security, 2012.

[9] Kiatchumpol Suttisirikul, and Putchong Uthayopas, Accelerating the Cloud Backup using GPU based Data Deduplication, IEEE 18th International Conference on Parallel and Distributed Systems, Thailand.

[10] Hugo E. Camacho, J. Alfredo Brambila, Alfredo Pena, and Jose M. Vargas, A Cloud Environment for Backup and Data Storage, Engineering Information Technology, Polytechnic University of Altamira, Altamira Tamaulipas, Mexico.

[11] What is a Data Backup? "n.d", 25 June 2017, http://kb.winzip.com/kb/entry/12/

[12] What is cloud backup? "n.d", 25 June 2017, https://www.carbonite.com/en/cloud-backup/

personal/what-is-cloud-backup/

[13] Ethiopian Mobile Subscriber: Mobile Subscribers Reached Fifty-Three Million, March 2017, http://www.2merkato.com/news/alerts/5008-ethiopia-mobile-subscribers-reached-53-million.

[14] Stack Overflow, https://stackoverflow.com/

[15] D. Geer: Reducing the Storage Burden via Data De-duplication, Computer, vol.41, pp.15-17. Issue.12, Dec.2008.

[16] Enrico Bocchi, Marco Mellia, and Sofiane Sarni: Cloud Storage Service Benchmarking: Methodologies and Experimentations, IEEE 3rd International Conference on Cloud Networking (CloudNet), 2014.

[17] V. Spoorthy, M. Mamatha, B. Santhosh Kumar: A Survey on Data Storage and Security in Cloud Computing, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.6, June- 2014.

[18] What is cloud computing: https://aws.amazon.com/what-is-cloud-computing/

[19] Pratap P. Nayadkar. A Survey on Different Backup and Restore Techniques used in Mobile devices, Vol.5, Issue.6, 2014.

[20] Ch.anilkumar and Dr.R.Kirshnaiah, A Novel approach for unique Data Backup in Cloud Storage, Vol.2, No.4, July 2013.

[21] S. Collesei, P. di Tria, G. Morena, Short Message Service-Based Applications in the GSM Network.

[22] AMAZON WEB SERVICES, Amazon Simple Storage Service, http://aws.amazon.com/s3/.

[23] Ahmad Affandi, Saparudin, and Erwin, The Application of Text Compression to Short Message Service Using Huffman Table, Vol.6, No.1, 2011.

[24] Anna Isakov and Hao Shi, Review of J2ME and J2ME-based Mobile Applications, International Journal of Computer Science and Network Security, VOL.8 No.2, February 2008.

[25] Hobin Yoon, Ada Gavrilovska, and Karsten Schwan, Interactive Use of Cloud Services: Amazon SQS and S3, 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2012.

[26] Kirti Madhukar Battase, Neha Vijay Barve, Parinita Bajirao Kandekar, Lata Vitthal Sanap, Android Application for SMS Compression, International Journal of Research in Advent Technology, Vol.2, No.4, April 2014.

[27] Java ME Archive https://billday.com/javame/

[28] GMS, https://searchmobilecomputing.techtarget.com/definition/GSM

[29] Microsoft Visual Studio 2015, C# and .Net Framework, https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework.

[30] Manoj Patil, and Prof Vinay Sahu, A Survey of Compression and Encryption Techniques for SMS, International Journal of Advancements in Research & Technology, Volume 2, Issue 5, May -2013.

[31] Nigam Sangwan, Text Encryption with Huffman Compression, International Journal of Computer Applications, Volume 54– No.6, September 2012.

APPENDICES

Appendices A: Questioner

THE NEED OF DATA BACKUP & RECOVERY QUESTIONNAIRE

Statement of Informed Consent

You are being asked to participate as a volunteer in a research study conducted by ***Kelele Gebremedhin***, a master's student at Adama Science & Technology University (ASTU) in Software Engineering. ***This study is designed to gather information and explore the need for mobile data backup & Recovery (Contacts, Call Logs, & Messages) for Ethio-Telecom mobile customers using cellular (GSM) network.***
The research is being conducted under the supervision of ***Dr. Mesfin Haile. Abebe***

You will be one of many participating in this study by completing this questionnaire.

1. Your participation in this project is voluntary; You may withdraw from the study at any time without penalty or harm of any type. If you decline to participate in or choose to not complete the questionnaire, the researcher will not inform anyone of your decision, and no foreseeable negative consequences will result.
2. Completing the questionnaire will require approximately 15 - 20 minutes. There are no known risks associated with completing the questionnaire. If, however, you feel uncomfortable in any way during this process, you may decline to answer any question.
3. The researcher will not identify you by name in any report using information obtained from your questionnaire; your confidentiality as a participant in this study will remain secure. Subsequent uses of data generated by this questionnaire will protect the anonymity of all individuals.
4. This research effort and this questionnaire have been reviewed and approved by the ASTU ***Department of Graduate Council(DGC)***, which functions as the Institutional Review Board for graduating students work.

NB:

- ***BY COMPLETING AND SUBMITTING THIS QUESTIONNAIRE, YOU ARE INDICATING THAT YOU UNDERSTAND THE STATEMENTS ABOVE, AND CONSENT TO PARTICIPATE IN THIS STUDY.***
- ***DO NOT PUT YOUR NAME ON THE QUESTIONNAIRE; YOUR SIGNATURE ACKNOWLEDGING THAT YOU UNDERSTAND THE INFORMATION PRESENTED ABOVE IS NOT REQUIRED.***
- ***PLEASE FILL OUT THIS QUESTIONNAIRE. IF YOU HAVE COMPLETED THIS QUESTIONNAIRE BEFORE, PLEASE DO NOT FILL IT OUT AGAIN.***

THANK YOU

For Ethio-Telecom Customer Service Agents

1. Do you use a mobile phone? (**Please tick one**)

A ☐ Yes B) ☐ No

2. If your answer to question number **1** is **A** what kind of mobile phone? (**Please tick one**)

A) ☐ Nokia D) ☐ BlackBerry

B) ☐ iPhone E) ☐ Android

C) ☐ Windows F) ☐ Another Phone

3. Do you use remote cloud backup storage? (**Please tick one**)

A) ☐ Yes C) ☐ I Know Nothing

B ☐ No D) ☐ No, because there is no local responsible repository

4. What kind of data do you store on your mobile device? (**Tick on the checkbox, Multiple selections is possible**)

A) ☐ Contacts E) ☐ Photos

B) ☐ Call Logs F) ☐ Videos

C) ☐ Message G) ☐ Music

D) ☐ Files/Documents

5. How do you rank the importance of the data found on your mobile devices? Please write your rank from highest **(1)** to lowest **(7)** in the checkbox below?

A) ☐ Contacts E) ☐ Photos

B) ☐ Call Logs F) ☐ Videos

C) ☐ Message G) ☐ Music

D) ☐ Files/Documents

6. If your answer to question number **3** is **A,** please write down the service name and the reason you choose this service.

_____

_____

_____

7. Have you ever lost your mobile device?

A) ☐ Yes B) ☐ No

8. If your answer to question number **7** is **A**, did you replace it with another one? If you did, with which mobile device is replaced? **(Please tick one)**

A) ☐ Android D) ☐ BlackBerry

B) ☐ iPhone E) ☐ Nokia

C) ☐ Windows F) ☐ Another Phone

9. If your answer to question number **7** is **A**, your answer to question number **3** is **A** and you did replace it with another phone, did you recover your file from the remote server.

A) ☐ Yes B) ☐ No

10. If your answer to question number **9** is **B**, why?

_____

_____

_____

11. As a customer service agent; have ever faced any request from customers that someone who has lost phone and needs to recover it and asks you if there is any Ethio-Telecom service for recovering lost device data.

A) ☐ Yes B) ☐ No

12. What do you think of using local repository? a data backup repository for maintaining your mobile data using the internet.

13. What do you think of using local repository? for maintaining backup and recovery of mobile data using the normal cellular network, independent of the internet?

14. What do think will be the possible challenges or the things needs to be considered for providing this solution using the GSM network?

For Ethio-Telecom Customers/Mobile Users

1. Do you use a mobile phone? (**Please tick one**)

A ☐ Yes B) ☐ No

2. If your answer to question number **1** is **A** what kind of mobile phone? (**Please tick one**)

A) ☐ Nokia D) ☐ BlackBerry

B) ☐ iPhone E) ☐ Android

C) ☐ Windows F) ☐ Another Phone

3. Do you use remote cloud backup storage? (**Please tick one**)

A) ☐ Yes C) ☐ I Know Nothing

B ☐ No D) ☐ No, because there is no local responsible repository

4. What kind of data do you store on your mobile device? (**Tick on the checkbox, Multiple selections is possible**)

A) ☐ Contacts E) ☐ Photos

B) ☐ Call Logs F) ☐ Videos

C) ☐ Message G) ☐ Music

D) ☐ Files/Documents

5. How do you rank the importance of the data found on your mobile devices? Please write your rank from highest **(1)** to lowest **(7)** in the checkbox below?

A) ☐ Contacts E) ☐ Photos

B) ☐ Call Logs F) ☐ Videos

C) ☐ Message G) ☐ Music

D) ☐ Files/Documents

6. If your answer to question number **3** is **A,** please write down the service name and the reason you choose this service.

_____

_____

_____

7. Have you ever lost your mobile device?

A) ☐ Yes B) ☐ No

8. If your answer to question number **7** is **A**, did you replace it with another one? If you did, with which mobile device is replaced? **(Please tick one)**

A) ☐ Android D) ☐ BlackBerry

B) ☐ iPhone E) ☐ Nokia

C) ☐ Windows F) ☐ Another Phone

9. If your answer to question number **7** is **A**, your answer to question number **3** is **A** and you did replace it with another phone, did you recover your file from the remote server.

A) ☐ Yes B) ☐ No

10. If your answer to question number **9** is **B**, why?

_____

_____

_____

11. What do you think of using local repository? a data backup repository for maintaining your mobile data using the internet.

12. What do you think of using local repository? for maintaining backup and recovery of mobile data using the normal cellular network, independent of the internet?

For Ethio-Telecom Management

1. What are the existing Ethio-Telecom service packages?
2. Have ever face any request for mobile data back-ups, like a need to recover lost or failed device contacts and messages.
3. Do you have any cloud service, cloud environment or have a plan adopting cloud environment?
4. What are the possible options and mechanisms for data transferring using GSM network?
5. Do you have a plan for providing backup and recovery service through GSM or internet or both?
6. What could be the possible challenges for providing this solution, do you think it is feasible? In terms of
    a) Security, Bandwidth and network speed, Infrastructure

Appendices B: Sample User Interface

Android Starting Up: **-** the user interface, **figure Appendix C-1: Android Starting Up** below represents the UI when the android app is starting.

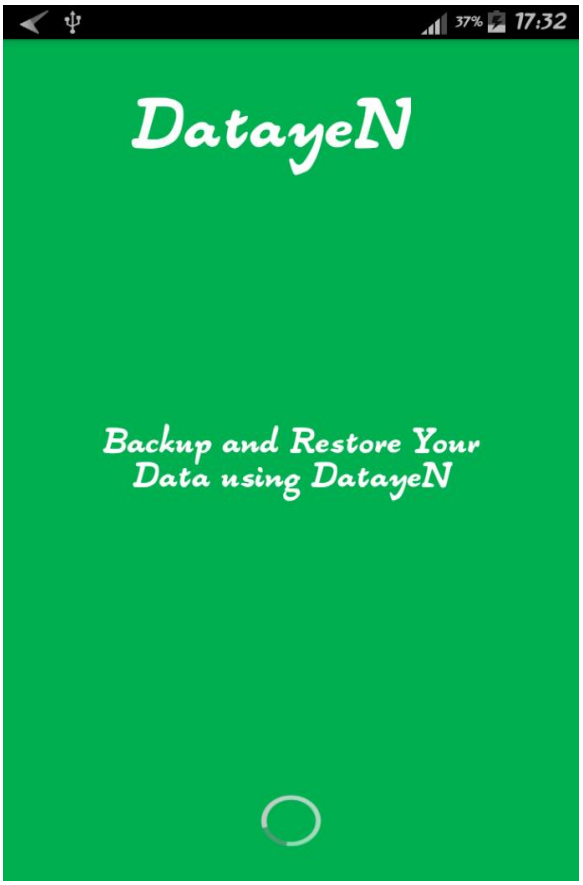

Figure Appendix C- 1: Android Starting Up

Android Started: **-** the user interface, **figure Appendix C-2: Android Started,** represents the UI when the android app started and shows the phone data that can be synchronized, like Contacts, Messages, pictures, videos, music and documents, and other additional features.

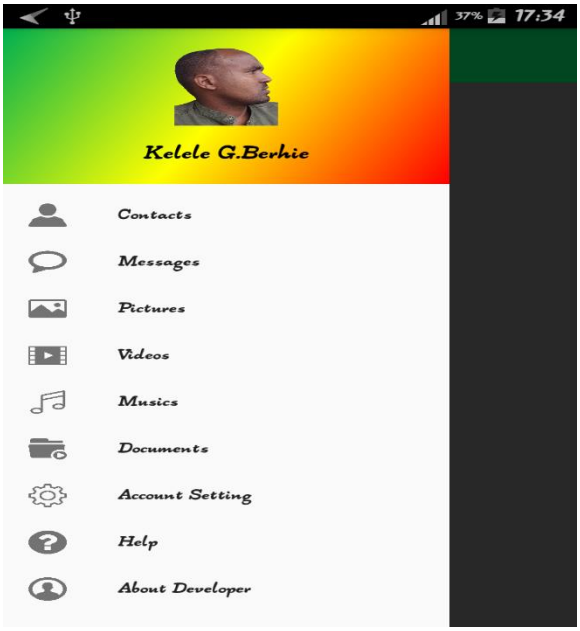

Figure Appendix C- 2: Android Started

Synchronize Contacts: **-** the user interface, **figure Appendix C-3: Synchronize Contacts,** represents the UI where phone contacts synchronize.

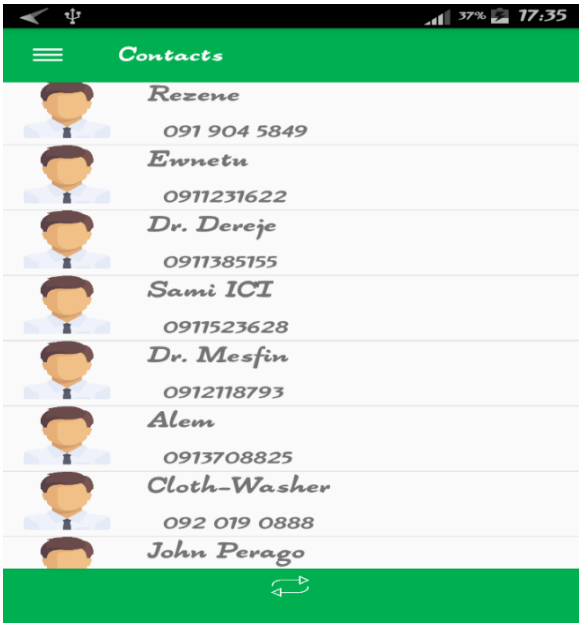

Figure Appendix C- 3: Synchronize Contacts

Synchronize Messages: - the user interface, **figure Appendix C-4: Synchronize Messages,** represents the UI where phone messages synchronize.

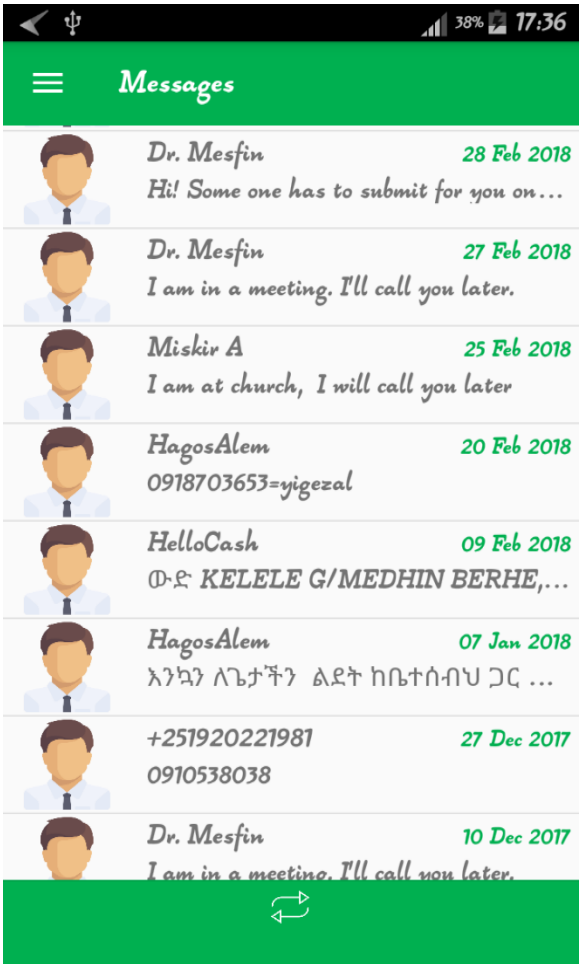

Figure Appendix C- 4: Synchronize Messages

Synchronize Pictures: - the user interface, **figure Appendix C-5: Synchronizing Pictures,** represents the UI where phone pictures synchronize

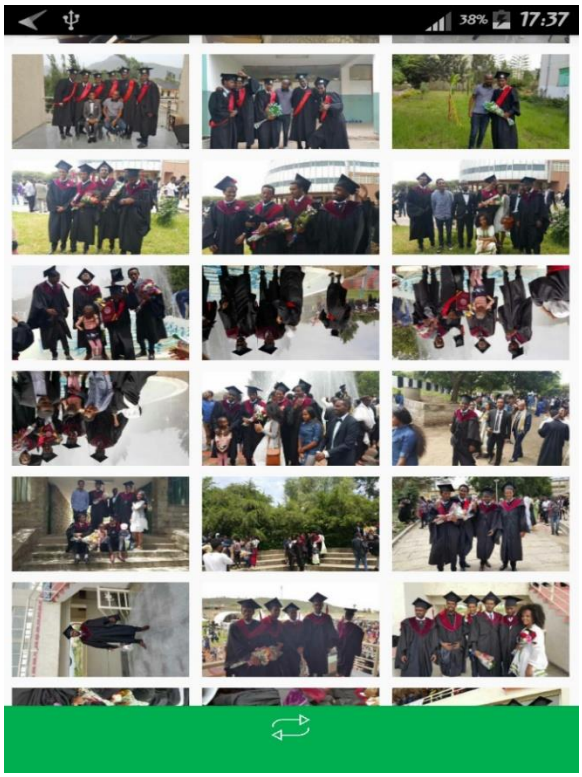

Figure Appendix C- 5: Synchronize Pictures

Synchronize Music: - the user interface, **figure Appendix C-6: Synchronize Music,** represents the UI where phone music synchronizes.

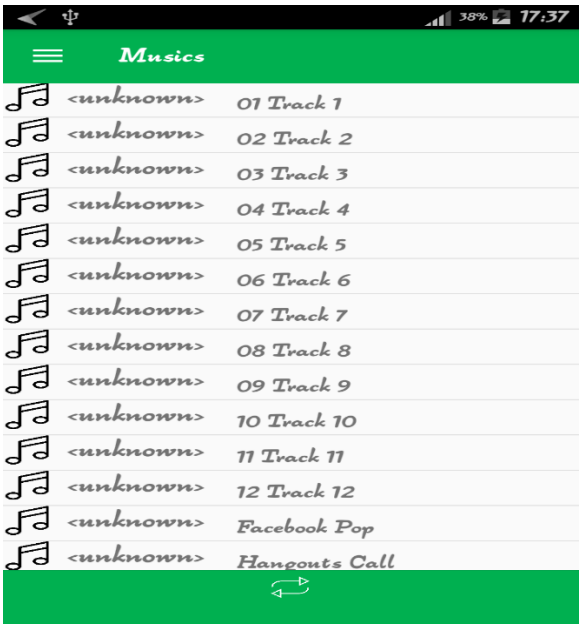

Figure Appendix C- 6: Synchronize Music

J2ME app Started: **-** the user interface, **figure Appendix C-7: J2ME Started,** represents the UI where the J2ME is started and displays the data options that can be synchronized.

Figure Appendix C- 7: J2ME started

Synchronize Contacts: **-** the user interface, **figure Appendix C-8: Synchronize Contacts,** represents the J2ME UI where the synchronizes phone contacts.

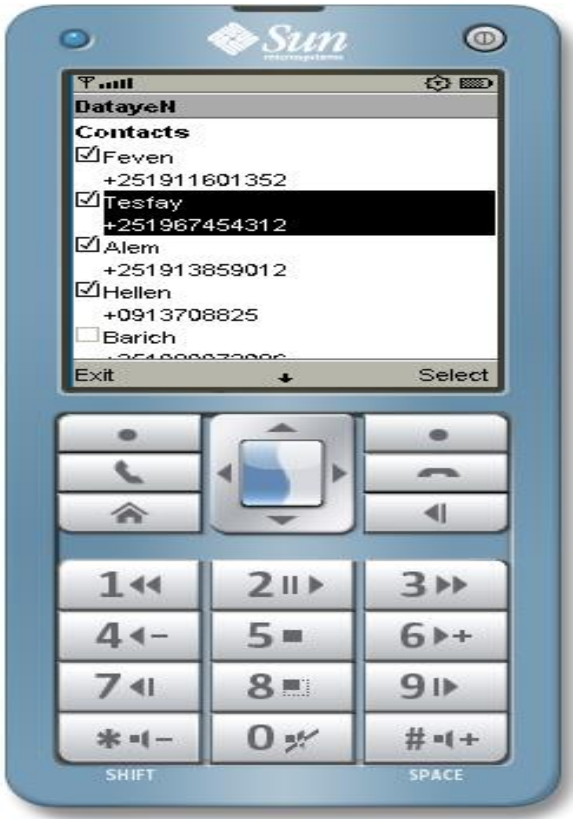

Figure Appendix C- 8: Synchronize Contacts

Confirmation: **-** below confirmation window.

Figure Appendix C- 9: Pushing Contacts Confirmation

The following section represents snapshots of the containers and the blobs inside each container in Microsoft Azure cloud storage.

The Contacts Container: - this Azure storage container is a virtual folder inside the Azure storage which contains user's contact blobs. In this container, each user's contacts address book is maintained as a single file named after the user mobile phone number.

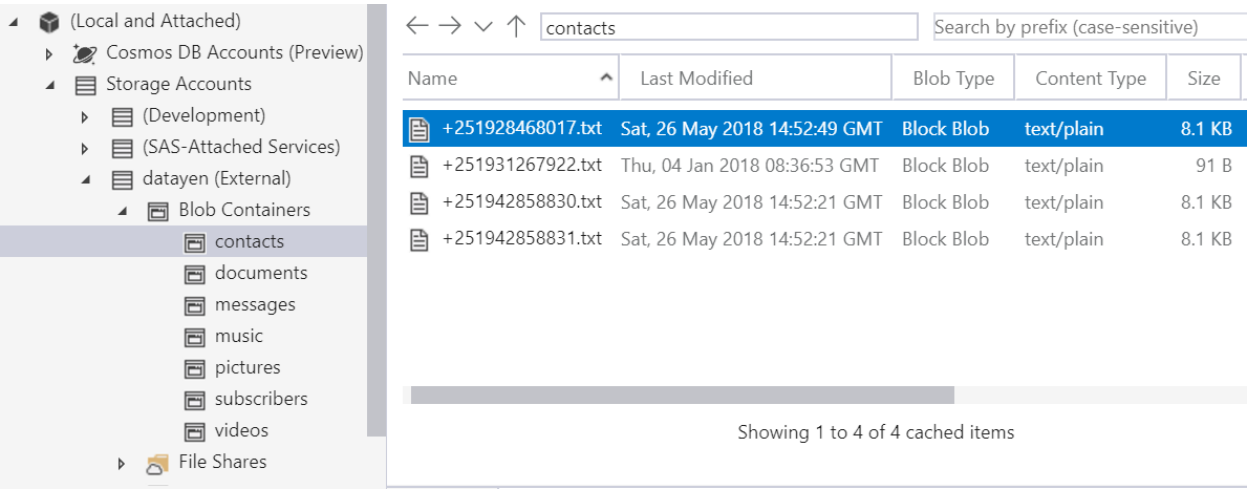

Figure Appendix C- 10: Contacts Container

The Messages Container: - this Azure storage container is a virtual folder inside the Azure storage which contains user's message blobs. In this container, each user's messages is maintained as a single file named after the user mobile phone number.

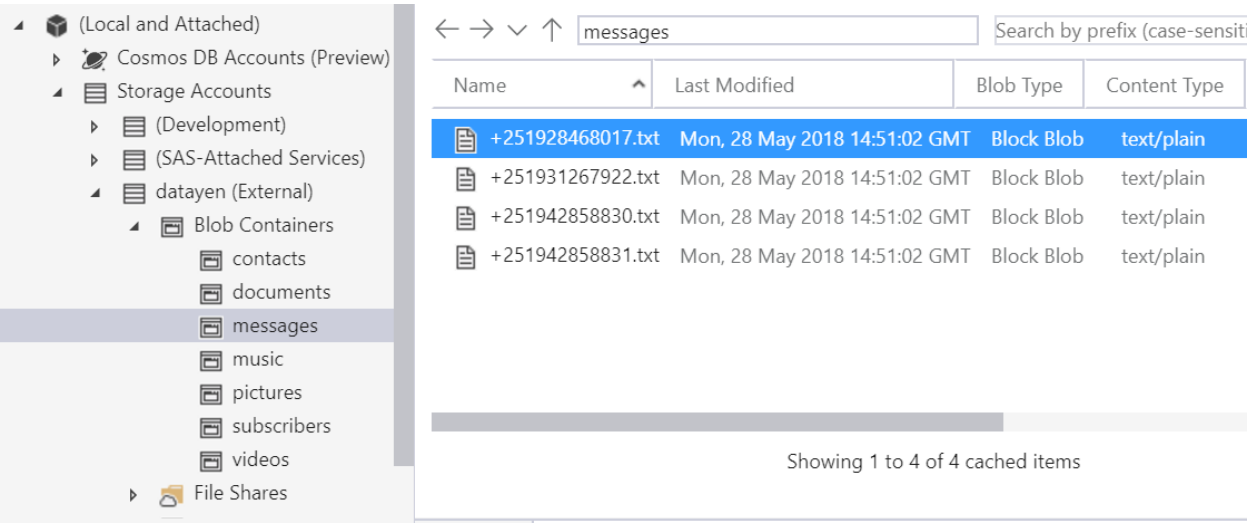

Figure Appendix C- 11: Messages Container

The Pictures Container: **-** this Azure storage container is a virtual folder inside the Azure storage which contains other containers named after the user mobile phone number that can contain user's picture blobs with their actual name.

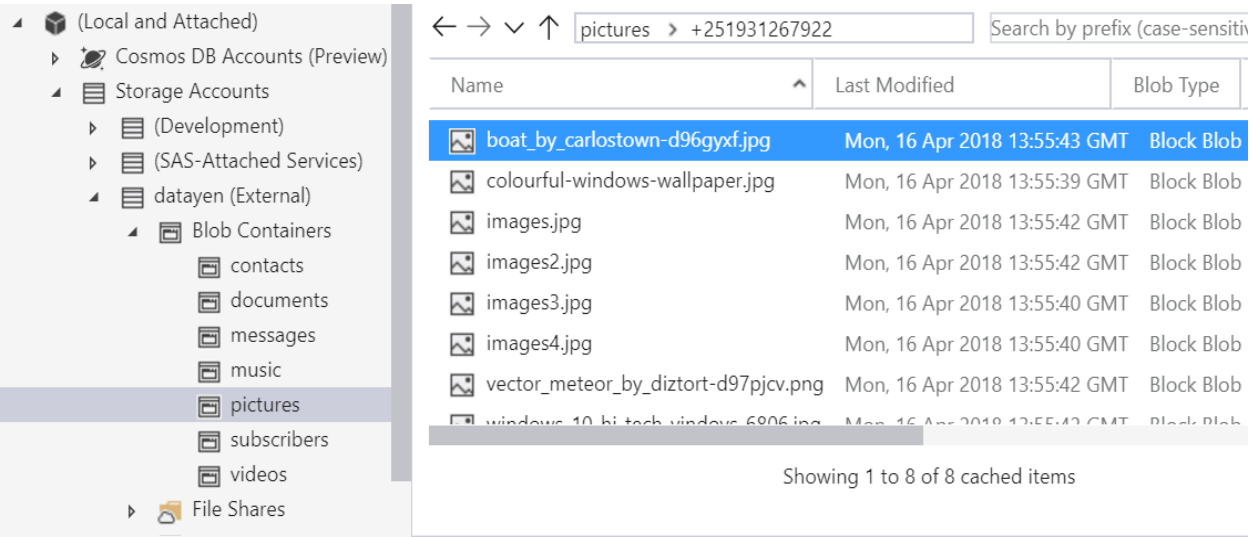

Figure Appendix C- 12: Pictures Container

The Music Container: **-** this Azure storage container is a virtual folder inside the Azure storage which contains other containers named after the user mobile phone number that can contain user's music blobs with their actual name.

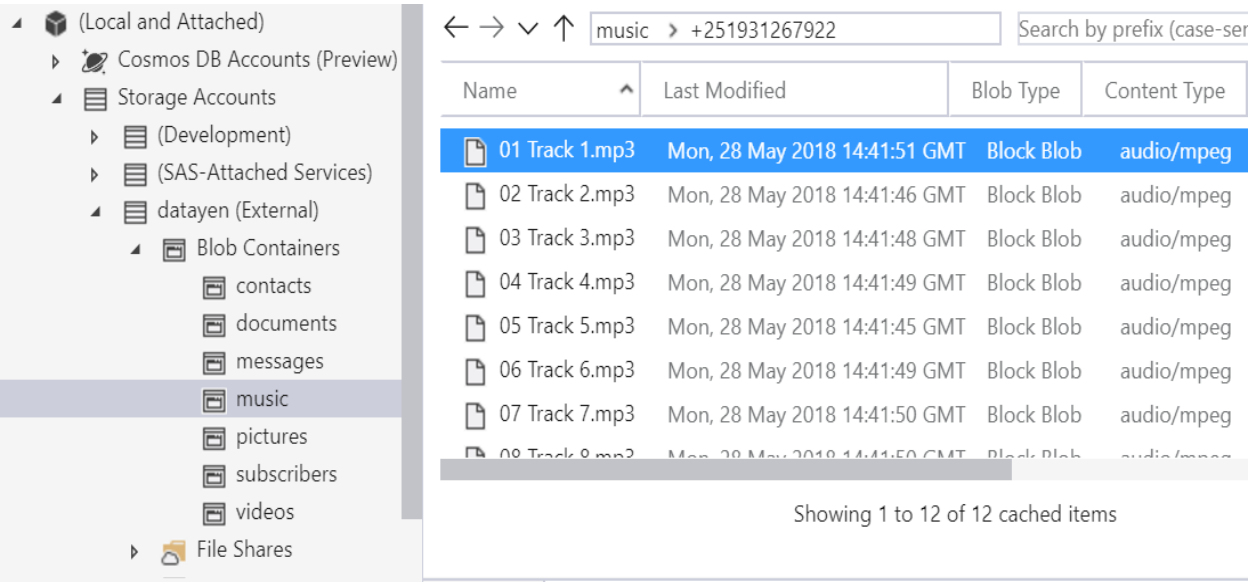

Figure Appendix C- 13: Music Container

The Videos Container: **-** this Azure storage container is a virtual folder inside the Azure storage which contains other containers named after the user mobile phone number that can contain user's video blobs with their actual name.

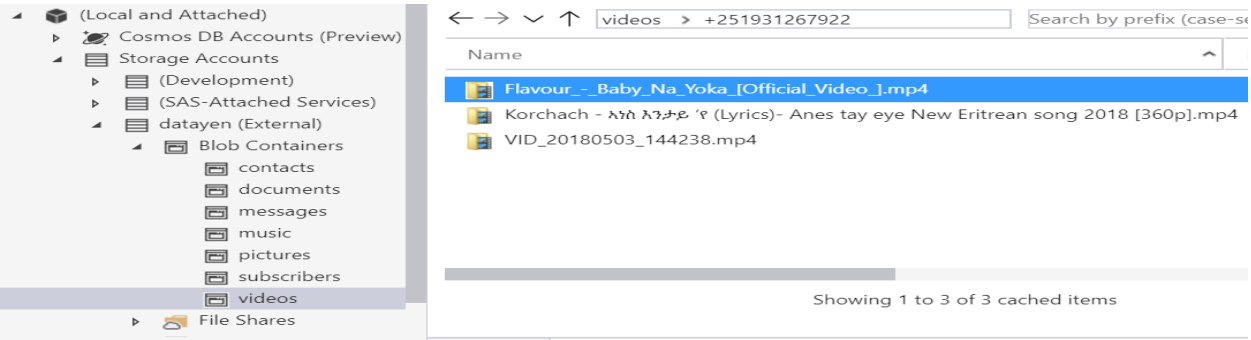

Figure Appendix C- 14: Videos Container

The Documents Container: **-** this Azure storage container is a virtual folder inside the Azure storage which contains other containers named after the user mobile phone number that can contain user's file blobs with their actual name.

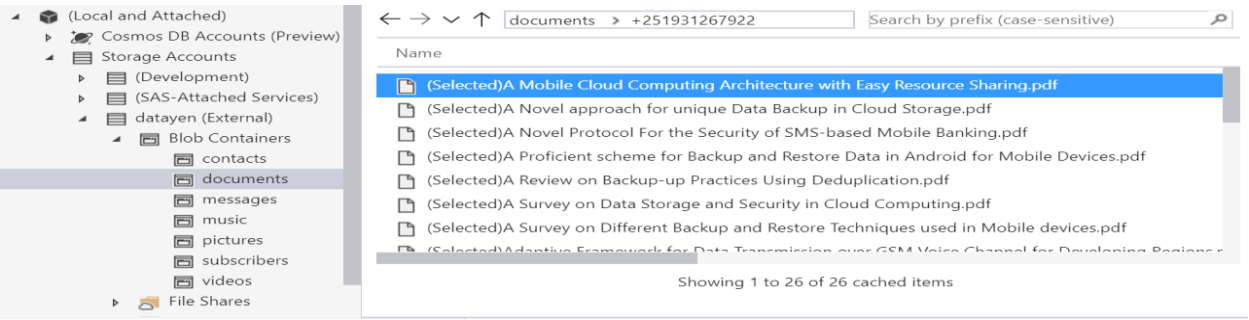

Figure Appendix C- 15: Documents Container

The Subscribers Container: **-** this Azure storage container is a virtual folder inside the Azure storage which contains subscribers(phone number) information as file blobs.

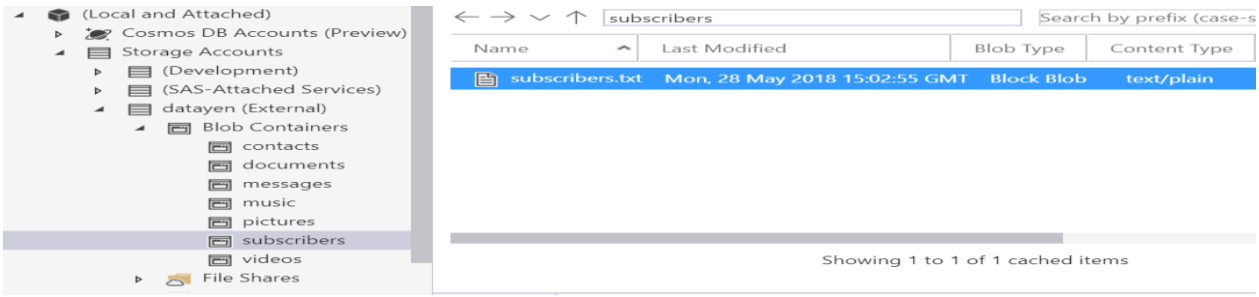

Figure Appendix C- 16: Subscribers Container

Appendices C: Sample Source Code

Reading Contacts and Sync: - the following represents reading the phone contacts and synchronizing with Azure storage.

```
/**
 * Created by DatayeN on 30-Oct-17.
 */

ArrayList<ContactModel> android_contacts = new ArrayList<ContactModel>();

public void GetContactsList() {
    new Thread(new LoadContacts()).start();
    ContactModel_Adapter adapter = new ContactModel_Adapter(this, android_contacts);
    listView.setAdapter(adapter);
    synchronizeContacts(android_contacts);
}
class LoadContacts implements Runnable {
    @Override
    public void run() {
        Cursor contact_item =
getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
null, null, null, null);
        ContactModel contact_data;
        while (contact_item.moveToNext()) {
            contact_data = new ContactModel();
            contact_data.Name =
contact_item.getString(contact_item.getColumnIndex(ContactsContract.CommonDataKinds.Pho
ne.DISPLAY_NAME));
            contact_data.Phone_Number =
contact_item.getString(contact_item.getColumnIndex(Phone.NUMBER));
            contact_data.ID =
contact_item.getString(contact_item.getColumnIndex(ContactsContract.Contacts._ID));
```

```
            android_contacts.add(contact_data);
        }
        contact_item.close();
    }
}

class ContactModel {
    String Name = "";
    String ID = "";
    String Phone_Number = "";
}
class ContactModel_Adapter extends BaseAdapter {
    Context context;
    List<ContactModel> android_contacts;
    public ContactModel_Adapter(Context context, List<ContactModel> contacts) {
        this.android_contacts = contacts;
        this.context = context;
    }
    @Override
    public int getCount() {
        return this.android_contacts.size();
    }
    @Override
    public Object getItem(int position) {
        return this.android_contacts.get(position);
    }
    @Override
    public long getItemId(int position) {
        return position;
    }
```

```
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        View view = View.inflate(context, R.layout.contact_item, null);
        TextView contact_name = (TextView) view.findViewById(R.id.contact_name);
        TextView contact_number = (TextView) view.findViewById(R.id.contact_number);
        contact_name.setText(android_contacts.get(position).Name);
        contact_number.setText(android_contacts.get(position).Phone_Number);
        return view;
    }
}
```

Reading Messages and Sync: **-** the following represents reading the phone messages and synchronizing with Azure storage.

```
/**
 * Created by DatayeN on 30-Oct-17.
 */

public class Message extends BaseActivity {
    ListView listView;
    ArrayList<MessageModel> android_messages= new ArrayList<>();

    public Message(){}
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        FrameLayout contentFrameLayout = (FrameLayout) findViewById(R.id.content_frame);
        getLayoutInflater().inflate(R.layout.messages_layout, contentFrameLayout);
        listView = (ListView) findViewById(R.id.message_list);
        Uri uriSms = Uri.parse("content://sms/inbox");
        Cursor cursor = getContentResolver().query(uriSms, new String[]{"_id", "address",
```

```
"date", "body"},null,null,null);
        cursor.moveToFirst();
        MessageModel message_data;
        DateFormat formatter = new SimpleDateFormat("yyyy/MM/dd");
        SimpleDateFormat postFormater = new SimpleDateFormat("dd MMM yyyy");
        Date dateConvertor = new Date();
        while(cursor.moveToNext())
        {
           try
           {
              String recievedDate = cursor.getString(2);
              dateConvertor = formatter.parse(formatter.format(Long.parseLong(recievedDate)));
           }
           catch (Exception e)
           {
              e.printStackTrace();
           }
           message_data = new MessageModel();
           String address = cursor.getString(1);
           String body = cursor.getString(3);
           message_data.address_phone = address;
           message_data.address_name = getContactName(getApplicationContext(),
cursor.getString(cursor.getColumnIndexOrThrow("address")));
           message_data.content = body;;
           message_data.date = postFormater.format(dateConvertor).toString();
           android_messages.add(message_data);
        }
        MessageModel_Adapter adapter = new MessageModel_Adapter(this, android_messages);
        listView.setAdapter(adapter);
     }
```

```
public  void synchronizeMessages(android_messages);
public String getContactName(Context context, String phoneNumber) {
   ContentResolver cr = context.getContentResolver();
   Uri uri =
Uri.withAppendedPath(ContactsContract.PhoneLookup.CONTENT_FILTER_URI,
         Uri.encode(phoneNumber));
   Cursor cursor = cr.query(uri,
         new String[] { ContactsContract.PhoneLookup.DISPLAY_NAME }, null, null, null);
   if (cursor == null) {
      return null;
   }
   String contactName = null;
   if (cursor.moveToFirst()) {
      contactName = cursor.getString(cursor
            .getColumnIndex(ContactsContract.PhoneLookup.DISPLAY_NAME));
   }
   if (!cursor.isClosed()) {
      cursor.close();
   }
   return contactName;
}
class MessageModel{
   String address_name = "";
   String address_phone ="";
   String content = "";
   String date = "";
}
class MessageModel_Adapter extends BaseAdapter {
   Context context;
   List<MessageModel> android_messages ;
```

```
    public MessageModel_Adapter(Context context, List<MessageModel>
android_messages){
        this.context = context;
        this.android_messages = android_messages;
    }
    @Override
    public int getCount() {
        return this.android_messages.size();
    }
    @Override
    public Object getItem(int position) {
        return this.android_messages.get(position);
    }
    @Override
    public long getItemId(int position) {
        return position;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
      View view = View.inflate(context, R.layout.message_item, null);
      TextView address_value = (TextView) View.findViewById(R.id.message_item_address);
      TextView content_value = (TextView) view.findViewById(R.id.message_item_content);
      TextView date_value = (TextView) view.findViewById(R.id.message_item_date);
      if(android_messages.get(position).address_name != null)
          address_value.setText(android_messages.get(position).address_name);
        else
         address_value.setText(android_messages.get(position).address_phone);
        content_value.setText(android_messages.get(position).content);
        date_value.setText(android_messages.get(position).date);
        return view;   } } }
```