m-Suvidha (FinTech App)

Submitted
in partial fulfillment of
the requirements of the degree of
Bachelor of Technology, Computer Engineering
2015-2016

Submitted by
Tejas Sankhe (121070012)
Hitesh Parmar (121070026)
Venali Sonone (121071050)
Priyanka Parle (121071059)

Under the guidance of
Prof. S. G. Bhirud

Department of Computer Engineering and Information Technology
Veermata Jijabai Technological Institute
(Autonomous Institute Affiliated to University of Mumbai) Matunga,
Mumbai - 400019

Declaration of the Student

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and references the original sources. We also declare that we have adhere to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any data in our submission. We understand that any violation of the above will be cause for disciplinary action by Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Tejas Sankhe
[121070012]

Hitesh Parmar
[121070026]

Venali Sonone
[121071050]

Priyanka Parle
[121071059]

Approval Sheet

This is to certify that Tejas Sankhe (121070012), Hitesh Parmar (121070026), Venali Sonone (121071050), and Priyanka Parle (121071049), students of B.Tech (Computer Engineering), VJTI, Mumbai - 400019 have successfully completed the initial report on "m-Suvidha (FinTech App)" under the guidance of Prof S.G. Bhirud as a part of Final Year Project.

____________	____________
Internal Examiner	External Examiner

Approval Sheet

This is to certify that Tejas Sankhe (121070012), Hitesh Parmar (121070026), Venali Sonone (121071050), and Priyanka Parle (121071049), students of B.Tech (Computer Engineering), VJTI, Mumbai - 400019 have successfully completed the initial report on "m-Suvidha (FinTech App)" under the guidance of Prof S.G. Bhirud as a part of Final Year Project.

Project Guide

Prof. S. G. Bhirud
Professor
Computer Engineering
IT Department,
VJTI, Mumbai - 19

Head of Department

Dr. G. P. Bhole
Assoc. Professor and Head
Computer Engineering
IT Department,
VJTI, Mumbai - 19

Date:

Place:

Acknowledgement

We would like to thank our project guide and mentor, Prof. S. G. Bhirud for contributing his time and effort to help us with our project. His constant support and encouragement, along with numerous inputs and suggestions he gave, have been invaluable and have been a driving force which has constantly motivated us to explore different aspects of our project. He has always been involved by discussing our project at each phase to make sure that the project is designed and implemented in an appropriate manner and that our conclusions are appropriate. We would also like to thank Computer and IT Department Head Dr. G. P. Bhole for his continuous support in our idea and the project. Lastly, we would like to thank our classmates for their numerous inputs and constant support in our project that has helped us to make the project better.

Tejas Sankhe
[121070012]

Hitesh Parmar
[121070026]

Venali Sonone
[121071050]

Priyanka Parle
[121071059]

Abstract

m-Suvidha is a smart fin-tech android application that ensures the smooth onboarding of customers through electronic KYC within the specifications laid down by Reserve Bank of India. It provisions for the instant creation of accounts by the customers by performing the e-KYC through electronically signed documents. The application fetches the details of the customers from the UIDAI servers by taking the Aadhaar number or by scanning the Aadhaar QR code. The application aims to prevent money-laundering and other frauds by using OTP verification. These methods ensure the successful identification of the Aadhaar enabled customers through the use of Aadhaar API 2.0. The e-KYC can then be performed either by getting the user uploaded documents digitally signed by e-Mudhra or by getting e-signed documents from the customers in first place. The application provides a seamless intelligent interface to ease the process of account creation and online banking.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and Background

1.1 Introduction

Banking services have evolved substantially over the past few years. Net banking has revolutionized the way banking happens in India and abroad. However, there is more to be done. Identification and Access Management is one branch which is currently evolving due to increased security concerns. Banks require that each person is uniquely identified and authenticated for availing services so that security of the system is ensured. In India, the compulsory issuance of aadhar card to every citizen has provided a solution to this problem. Aadhar card can be used along with the biometric / OTP to verify a person's identity. It will facilitate inclusive banking as also bring transparency, effectiveness to the entire process. m-Suvidha is an application that uses Aadhar services to render e-KYC while successfully onboarding the users.

1.2 Background And Motivation

Opening a savings account even today is a tedious process, which is unfortunate, as this process hasn't effectively changed in the past two decades. The process is even more difficult for opening a current account.In this process, the primary concern for a customer is to keep reiterating and proving their identity. The key demographic and early adopters, namely the millennial, find it extremely difficult to dedicate so much time and energy to open a simple savings account. Small mistakes amount to re-doing the entire process. It is physically taxing as well. The process becomes even more complicated for unbanked customers Typically, opening an account requires - Proof of address; a photo, a signature and the proof of identity. It still takes 8 – 10 days to open the account along with getting a credit/debit card, cheque book etc.

1.3 Problem Statement

Currently there is no single seamless way to on-board customers without getting paper documents for validating KYC as well as establishing proof of identity for opening a wallet/savings account based on RBI's requirements. A major aspect of this is the inefficiency of paper work in its physical form: both for the customer to submit and the operational overhead to collect, scan and archive. Hence a solution is needed to create an on-boarding process that fulfills the RBI requirements, leveraging UIDAI as well as eMudhra (for digital certificates).

The customer or user impacted by the problem: The user is anyone who is currently availing of the bank services or even potential customers who may be associated with the bank in the future, specifically the millennial customers who do not have much time and wish to be on-boarded quickly.

1.4 Research Gap Analysis

The current process poses many problems like - Increasing the cost to serve for the bank in the form of couriers and printing of multifarious forms. It consumes a lot of the users' time for scanning and sending documents. Mistakes and changes amount to repeating the process, which increases operational time. This hampers customer experience leading to customer dissatisfaction Despite having given the same details multiple times before, the users' data doesn't get pre-printed which wastes their time. The need to attach the same documents repeatedly also consumes time and is frustrating. There is lack of resources to walk the new users through the whole process.

1.5 Impact

One major impact in the absence of a quick, effective and seamless ID and V system is loss of customers as the millennial won't wait for banks that do not cater to their needs. New business models especially e-commerce systems (like Paytm wallet) will take away the customers from traditional banks following slower processes.

1.6 Potential solution and business benefits

1.6.1 Ideal futuristic scenario if the problem is solved.

1. A good way to visualize this would be if account opening involves these key attributes:
2. This would lead to Thumb imprint or voice pattern used as an identifier
3. All documents being made available automatically once identified
4. A common KYC requirement (like the Asset management industry), that follows regulatory restrictions
5. Access to this process in a channel agnostic, device agnostic way.
6. Increased productivity
7. Higher customer delight
8. Reduced operation costs
9. Lesser time to on-board and serve customers
10. There is some additional customer information (data as well as document capture), which the bank won't be able to get from UIDAI. It would be interesting if the solution is able to include this information so that this makes the customer experience seamless and at the same time helps the bank complete the on-boarding process in a cost and time effective manner.
11. All of the above requirements need to fit in one app which a user should be able to download on his mobile and authenticate themselves by giving their biometrics (finger prints) on their own device.

12. In a nutshell, “A new on-boarding process needs to be envisioned, wherein customer information and customer authentication should happen at the same time which will result into the customer getting on boarded into the wallet platform post which the user can start doing transactions. The authentication needs to happen through UIDAI and e-Sign (e-Mudhra).

1.6.2 Benefit your customer/user.

The key customers would have a better experience at getting onboarded into using banking services, thus increasing the overall signups and reducing the friction to join such platforms. The secondary users, who are the bank employees, would be able to have less delays and more seamless process, thus reducing the overall operational efforts. The onboarding experience has been the same for the past two decades. Once simplified, this could be a game changer in the banking industry as it would not just create a more positive outlook among new clients, but would also provide a sense of assurance that banks care about the customer experience.

1.7 Regulatory implications associated with the problem.

RBI has stringent KYC policy that must be followed by each bank. More details on the regulations can be found: **(https://rbi.org.in/scripts/FAQView.aspx?Id=82)**

1.8 Report Orientation

The report will compromise of nine chapters with different content and scenarios providing the complete details about the project. The report is completed in such a way that it first provides the background Knowledge about the project and then gives the thorough details about it. It also has methodology and the conclusion. The different chapters of the report are as follows:

Chapter 1: Introduction and Background
This chapter will provide introduction to the project and motivation for performing it. This chapter describes current scenario, previous attempts, approach, and Problem statement.
Chapter 2: Literature Review
This chapter will provide literature studied of the project. The focus would be on the technology going to be used. In this we give detailed explanation of KYC-Literature,e-KYC process along with Intelligent chat bot for banking system literature.
Chapter 3: System Analysis and Design
This chapter is basically divided into System analysis and System Design. In this chapter we have UML diagram like Use case, sequence diagram, collaboration diagram, activity diagram and component diagram. In it we also focus on detail information of the data sets and we also have time line chart.

Chapter 4: Security of m-Suvidha

This chapter is dedicated to the three level security incorporated in to the m-Suvidha and also the globally accepted industry standards that the app adheres to.

Chapter 5: Artificial Intelligence: Chatterbot

This chapter is complete description of Artificial element of the m-Suvidha. It is chapter detailing implementation and technology of m-Suvidha chat bot.

Chapter 6: Implementation.

This chapter will tell about different techniques used for financial application we developed and their result.In this comparison of these techniques is also given to see which is better. Finally we also show performance analysis and screenshots of application.

Chapter 7: Accomplishment

This chapter registers the achievement of the m-Suvidha and its participation/acceptance at various events and academic achievements

Chapter 8: Conclusion

This chapter gives the conclusion and the issues, advantages of the techniques to be used and the future Scope

Chapter 9: References

This chapter lists all the references and guidelines used for the thesis and work produced.

Chapter 2

Literature Review

Literature Review

2.1 MOBILE Banking Services Survey

2.1.1 Overview

Recent innovations in the telecommunication have proven to be a boon for the banking sector and its customers:one of these is Mobile Banking, where customers interact with the bank via mobile phones and banks provide them the services like short message services, fund transfers, account details, issue of cheque book etc. Presently almost all the banks in the world have started providing their customers "Mobile Banking" services. The main issue of this study is to understand the factors which contribute to user's intention to use the mobile banking services. The purpose of this review is to explore the factors that influence the adoption behavior of mobile banking services by Indian consumers.This section also discusses the various steps that mobile banking providers should take to increase their mobile banking services user's database.

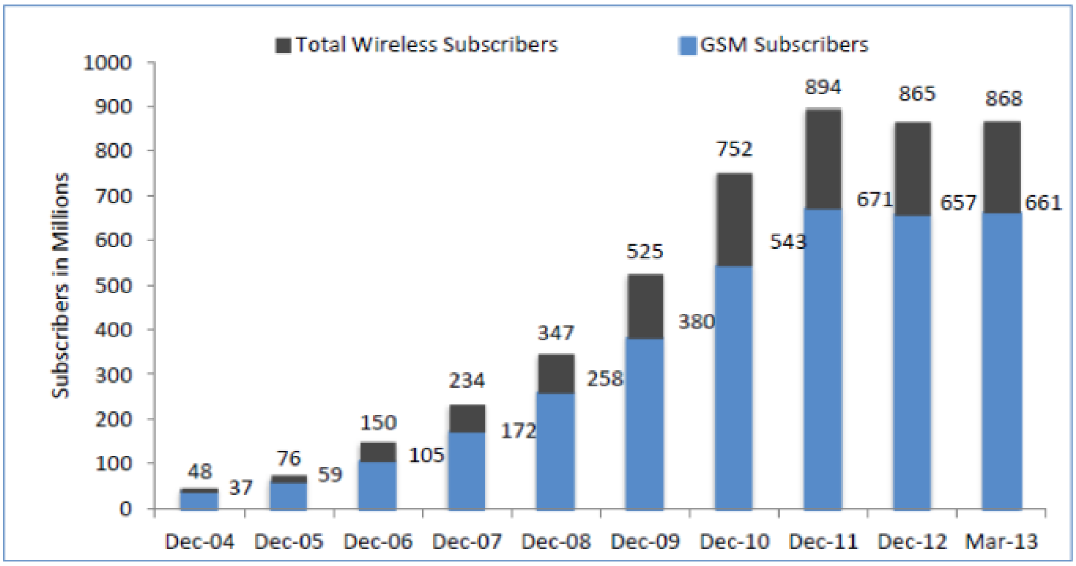
2.1.2 Introduction

According to TRAI, mobile banking involves the use of mobile phones for banking transactions like fund transfer,balance check, etc. As per the extant guidelines of RBI, banks that are licensed, supervised and have a physical presence in India, are permitted to offer mobile banking services. Mobile Banking policies in India aim to enable funds transfer from an account in any bank to any other account in the same or any other bank (interoperability) on a real time basis irrespective of the mobile network the customer has subscribed to (TRAI, 2013). The Mobile phone plays a very important role in the development of mobile commerce and mobile banking.

2.1.3 History of Mobile Phones in India

A report of the Cellular Operators Authority of India (CAOI), regarding the entry of cell phones into India,depicts that it was in the year 1992 that telecommunication Sector in India liberalized to bridge the gap through Government spending and to provide additional resources for the nation's telecom target and the private sector was allowed to participate. In the year 1994 India was licensed to provide cellular mobile services granted by the government of India for the Metropolitan cities of Delhi, Mumbai, Kolkata and Chennai. Kolkata became the first metro to have a cellular network in 1995. TRAI was set up in the year 1997 for the regulation of telecommunication sector in India. In March 1999 National Telecom Policy (NTP) was announced. In 2003 CDMA network was launched. In 2004 Broadband policy was announced. Mobile phone subscribers had reached 100 Million by 2006. In 2008, RBI issued operative guidelines for banks for mobile banking transactions in India. By the year 2009, wireless subscriber base crossed 400 million. At present wireless mobile phone subscribers are 867 Million i.e. it has almost doubled in the last four years. With the advancement in the operating systems of the mobile phones and mobile technology like 2G, 3G,4G has brought a significant change in the way of working of mobile banking services providers.

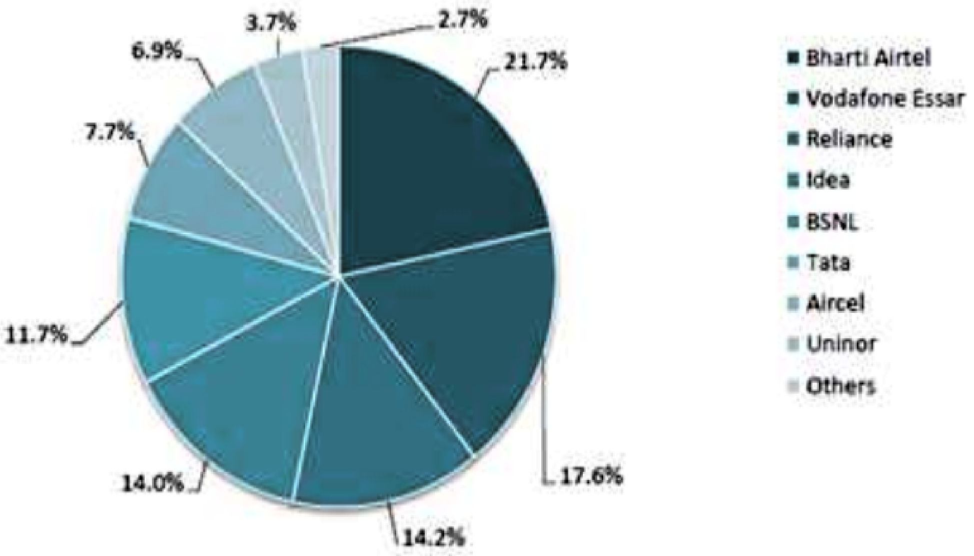
Since the introduction of 2G and the subsequently 3G, the demand for mobile phone has increased many folds. This can be interpreted by a rapid increase in the number of mobile phone subscribers.Refer Fig: 2.1



Source: TRAI & COAI Annual Report, 2013

Figure 2.1: All India Total Cellular and GSM Cellular Subscriber Base

There are many wireless operators in India but Bharati Airtel has got the maximum share of 21.7 percent after the Vodafone Essar (17.6 percent). Refer Fig: 2.2

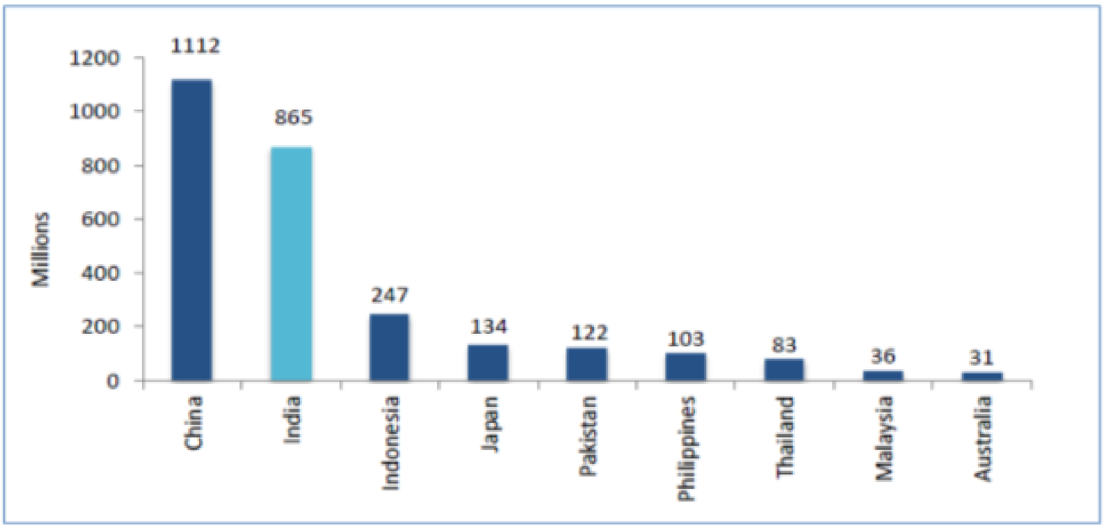


Source: The Indian Telecom Services Performance Indicators, TRAI March, 2013

Figure 2.2: Market Share of Wireless Operators

2.1.4 Top Ten Countries in Mobile Phone Subscribers Base

Mobile phone technology has become very common in all the countries of the world. According to Merrill Lynch Global research report 2011, China has the maximum number of mobile phone subscribers i.e. 1112 million and India stands on the Second position with 865 mobile phone subscribers. Refer Fig: 2.3



Source: Merrill Lynch Global Research Dec- 2013

Figure 2.3: International Trend of Subscriber Base

2.1.5 Evolution of Mobile Commerce

Mobile Commerce in India is increasing at a very fast pace. According to TRAI (2013), subscribers who access the internet through wireless phones are 143.2 Million. Mobile commerce has emerged after the introduction of electronic commerce. A simple definition of E-Commerce describes it as: “ the buying and selling of products and services over the Web” (Kalakota and Robinson, 2001). E-Commerce has gained importance in the last few years. E-Commerce applications developed so far, assume basically fixed users with wired infrastructure such as PC Connected with internet using a LAN (Local Area Network). Many new E-Commerce applications are possible using wireless and mobile networks. These applications are termed as ‘Wireless E-commerce’ or ‘Mobile Commerce’. With the increase in the number of wireless internet subscribers and advancement in the operating systems of mobile phones, mobile commerce has reached to every nook and corner of the world.

M-Commerce is an area which is rapidly changing the way people conduct their financial transactions. Tiwari, Buse and Herstatt (2006) discussed the features of mobile Commerce. According to the author M- commerce is characterized by many unique features as compared to the conventional form of commercial transactions like: Ubiquity, Immediacy, Localization, Instant Connectivity, Proactive Functionality and Simple Authentication Procedure.

- Ubiquity: It means users can avail the services and carry out transactions independent of the geographical location (‘anywhere’ feature)
- Immediacy: This feature is attractive in the way users can buy the goods anytime, i.e. without a wait (‘anytime’ feature).
- Localization: Positioning technologies i.e. GPS (Global Positioning Services) allows companies to offer goods and services to the user as per his/her current location.
- Instant Connectivity: With the introduction of the GPRS (General Packet Radio Service) mobile users are constantly online. This feature brings convenience to the users.

- Localization: Positioning technologies i.e. GPS (Global Positioning Services) allows companies to offer goods and services to the user as per his/her current location.
- Instant Connectivity: With the introduction of the GPRS (General Packet Radio Service) mobile users are constantly online. This feature brings convenience to the users.

- Pro-Active Functionality: M-commerce brings opportunities for the companies like push marketing, where users can opt for ‘Opt-in advertising’ so that they are i nformed about new products and services in the form of SMS.
- Simple Authentication Procedure: With the help of Subscriber Identity Module (SIM) and Personal Identification Number (PIN) the authentication process has become very simple.

2.1.5.1 Mobile Commerce Applications

Mobile services of similar nature can be bundled together as mobile applications.(Refer Table: 2.1) This study has been specifically focused on only one of the Mobile commerce application i.e. Mobile Banking.

2.1.5.2 Mobile Banking

Mobile Banking services were first offered by Kenya and Philippines in the world. M-PESA – Kenya: M-PE SA is the first mobile banking solution in the year 2007 by the telecom operators Safaricom and Vodafone. It has captured the majority of the market in Kenya and is very popular among the customers. SMART Money and G-Cash Philippines: Philippines launched SMART money, which is an electronic wallet and users do most of its banking transactions through mobile only.

There is a great scope of mobile banking in India as the number of mobile users is increasing. This is because of an increase in the number of wireless internet user subscriber base in India i.e. 143.2 Million (TRAI, 2013). In the year 2008, 3G was launched by MTNL (Mahanagar Telephone Nigam Ltd.) and IMPS (Immediate Payment Service) was also launched in 2010. After these initiatives and developments by

Table 2.1: M-commerce Applications

M-commerce Applications	
Application	**Examples of services offered**
Mobile banking	Mobile accounting Mobile brokerage Mobile financial information
Mobile entertainment	Mobile gaming Download of music and ring tones Download of videos and images Location-based entertainment services
Mobile information services	Currrent affairs(financial, sports and other news) Travel information Tracking services (personal and objects) Mobile search engines and dictionaries Mobile offices
Mobile marketing	Mobile couponing Direct (context sensitive) marketing Organization of mobile events Mobile newsletters
Mobile shopping	Mobile purchasing of goods and services
Mobile ticketing	Public transport sports and cultural events Air and rail traffic Mobile parking
Telematics services	Remote diagnosis and maintenance of vehicles Navigation services Vehicle tracking and theft protection Emergency services

RBI, mobile banking services have increased many folds and RBI issued the guidelines for banks to provide mobile banking services in India in the year 2008. These are:

- While opening an account in a bank
- Only such banks which are licensed and supervised in India and have a physical presence in India will be permitted to offer mobile payment services to residents of India.
- The services should be restricted to only to bank accounts/ credit card accounts in India which are KYC/AML compliant.
- Only Indian Rupee based services should be provided.
- Banks may use the services of business correspondents for extending this facility, to their customers.
- The guidelines with regard to use of business correspondent would be as per the RBI circulars on business correspondents issued from time to time.
- The 'Risks and Controls in Computers and Telecommunications' guidelines will equally apply to mobile payments.
- The "Know Your Customer (KYC)" and "Anti Money Laun dering (AML)" as prescribed by RBI from time to time would be applicable to customers opting for mobile based banking service.

2.1.6 Transaction Limits in Mobile Banking

- Only Indian rupee transactions and these transactions are allowed within India only.
- Per day transaction cap of Rs.50000 has been removed by RBI, and every bank can change this cap depending upon their risk.
- Transaction without end-to-end encryption is Rs.5000/- (SMS Based).

2.1.7 Security and Authentication

The highlights of security and authentication guidelines provided by the RBI on Mobile Banking:

- The M PIN or higher standard of mechanism should be used to authenticate the mobile banking customer.
- End-to-end secure encryption mechanism should be followed in transactions.
- The bank should conduct regular information security audits on the mobile banking systems to ensure complete security.

Despite many initiatives taken in the field of mobile banking there are only 12 percent (17 million) users out of 143.2 million mobile phone internet subscribers who are using banking services on their mobile phones (Alpesh Patel, 2013). So, the main issue of research is to understand the factors which contribute to users intention to use the mobile banking services. The purpose of this review paper is to explore the factors that influence the adoption behaviour of mobile banking services by Indian consumers.

2.1.8 Research Methodology

This section reviews the literature by identifying different articles, reports and research papers related to mobile banking. Different models are being used by many researchers like Technology Acceptance Model (TAM), Theory of Panned Behaviour (TPB) and Innovation Diffusion Theory (IDT) and these models are very helpful in determining the adoption of mobile banking services.

2.1.9 Literature

Mobile Banking, also known as M-Banking, can perform various functions like mini statement, checking of account history, SMS alerts, access to card statement, balance check, mobile recharge etc. via mobile phones (Vinayagamoorthy and Sankar, 2012). Banks are constantly updating their technology and want to increase their customer base by reaching to each and every customer. There are many advantages of using mobile banking, such as people in the rural or remote areas can also get an easy access to mobile banking whenever required. Vinayagamoorthy and Sankar, (2012) have discussed about the mobile banking and according to them it is a term that is used for performing various banking transactions like fund transfer, balance check, payments etc. via mobile phones.

First mobile banking transaction services in India were offered by ICICI bank in January 2008

(Mr. V. Vaidyanathan, 2008) but SMS alerts started in 2005-06 (Alpesh Patel, 2013). Wireless phone subscribers in India crossed 867.8 Million in 2013, as per TRAI (Telecom Regulatory Authority of India Act, 1997) as compared to 261.07 in March 2008. So there is approximately 4 times increase in the number of subscribers. However, according to this report, subscribers who access the internet through wireless phones are 143.2 Million. Almost 16.5 percent of wireless mobile phone subscribers are using the Internet over their mobile phones. According to a Mobile banking report by Deloitte (Alpesh Patel, 2013), 17 Million Indians are using mobile phones for banking transactions. So, approximately 2 percent of wireless phone subscribers are using banking services on their mobile phones. Mobile banking is still in its nascent stage in India. Therefore, identifying and understanding the factors influencing the behaviour of mobile phone subscribers is one of the fundamental requisite for development of mobile banking services in India.

Research in the field of mobile banking is at the introductory stage in India. It started in the year 2005-2006, with the introduction of short message services (SMS) of mobile alerts for transactions. Then in the year 2008, Reserve Bank of India (RBI) issued the guidelines for mobile banking transactions.

In the same year MTNL (Mahanagar Telephone Nigam Ltd.) launched 3G in India.In

2010-2011 India launched its first IMPS (Immediate Payment Service (IMPS) which is an instant interbank (similar to NEFT) transaction that can be initiated only through mobile phones or online or through SMS. In the year 2011-12, Vodafone and HDFC bank launched m-paisa and Airtel launched Airtel Money in 5 cities in India. In 2012-13 Airtel-Axis Bank launched a mobile banking service for financial inclusion and money transfer. According to operative guidelines for banks by RBI, only those banks which are licensed and supervised in India and have a physical presence in India will be permitted to offer mobile banking services (Chugh, 2014). According to RBI report, there are 82 banks that are permitted by RBI to provide mobile banking services throughout the India (Reserve bank of India, 2014) as compared to 21 Banks in the year 2010.

During the last four years, the numbers of banks providing mobile banking services in India have increased four times. But numbers of mobile banking users have not increased at the same pace. There are many challenges that Indian banks are facing for increasing the mobile banking user database like Handset operability, Security, Scalability and Reliability, Application Distribution etc. Acceptance and adoption of this innovative technology is very complex and this 'complexity' attribute is studied by various resear chers and they have suggested that banks should make these services easy to use by the Indian population because Indian population is not very well versed with this upcoming technology (Chaipoopirutana, Combs, Chatchawanwan and Vij (2009); Lin (2010); Sahin (2006).

To understand the adoption behaviour of users, many researchers have done research on the factors that helps in determining the acceptance and the attitude of users towards mobile banking. TAM (Technology Acceptance Model), TPB (Theory Planned Behaviour), IDT (Innovation Diffusion Model) (see Figure 5,6,7) have been discussed by Bhatti (2007) and Sadi and Noordin (2011) and they claimed that all the 13 factors i.e. Perceived Usefulness, Perceived ease of use, Personal Innovativeness, Perceived Trust, Perceived Cost, Subjective Norm, Social Influence, Self-Control, Perceived Behavioural Control, Facilitating condition, Self-Efficacy, Attitude towards use, and Intention to use M-commerce are statistically significant and by using exploratory factor analysis they concluded that the mere introduction of M-commerce is not sufficient but focus should be laid on the improvement of attributes that effect the M-Commerce adoption. Out of all the factors, perceived usefulness is found to be the critical factor thus, the service provider should take care that customers should perceive their services as valuable and useful to keep up with their fast paced lifestyle. This research also found that trust is also an important factor and should be taken into consideration by the Service providers; if consumers do not feel secure they will be reluctant to use the services. (Kim, Shin, and Lee 2007). It is also found that people have less trust in the mobile banking services and personal disposition to trust played a positive role in developing initial usage in mobile banking. To some extent the success of acceptance of M-commerce transactions depends on the customer as well as vendor's trust (Singh, Srivastava, and Srivastav, 2010). Kim, Shin, and Lee (2007) and AL-Majali and Mat (2011) also discussed that if customers believe that a mobile banking firm is able to develop effective service delivery strategies and provide adequate protection from fraud and violation of privacy, then adoption (or continue-to-use) intentions of the mobile phone users will increase.

Facilitating Condition is also an important attribute of consumer behavioural control towards intention to use; therefore it is necessary to improve the facilitating conditions of mobile application services like connection speed, secure systems and easy transaction method (sadi and Noordin, 2011).

Bhatti (2007), used all the three models TAM, TPB and IDT and found out that the perceived ease of use, perceived usefulness, subjective norm, personal innovativeness and perceived behavioural control are strong determinants of the intention to adopt M-commerce. The study has revealed that subjective norms and perceived behavioural control impact perceived ease of use and intention to adopt mobile commerce. Perceived control of users can be increased by offering them free use of service for a short period of time. Rapid adoption of technology, because of its social influence, is studied in terms of subjective norms and it is found to be a significant factor as the behavioural intention is very much affected by peer group influence.

Chaipoopirutana, Combs, Chatchawanwan, and Vij (2009) and Lin (2010), claimed that the adoption of mobile banking is 'complex' as it has the negative relatio n with intention to adopt mobile banking. In this paper they have discussed the Roger's (1995) innovation diffusion model's attributes: complexity, compatibility, relative advantage and trialability and found that Relative advantage, compatibility, ease of use (opposite of complexity) has a significant effect on attitude to adopt mobile banking services. They have also suggested that complexity must be reduced in order to increase the number of adopters in internet banking and compatibility has a positive relation with the adoption of internet banking. It implies that banks should start advertising their internet banking services to the consumers so that they can relate it to their values, beliefs and experiences of the adopters. Customers have a favourable attitude towards adopting mobile banking services, if they have positive belief about the relative advantage of mobile banking. Relative advantage refers to the degree to which a technology provides more benefits than its precursor (Rogers, 2003).

S.Samudra and Phadtare (2012) used the UTUAT model (see Figure) to investigate the adoption of mobile banking services and findings suggests that mobile banking services should be promoted to middle level managers whose salaries are in the range of 1-6 lacs and the age group is 25-30 as this is the most active age groups of 3G mobile. In UTUAT model, five factors are used to study the adoption of mobile banking: Performance expectancy, Effort expectancy, Social Influence, Facilitating Conditions and Voluntariness. Facilitating conditions seem to dominate in this study. As we make easy to use services the adoption rate will increase. Creating awareness about the services is also important as discussed by many other researchers (Safeena, Date, Kammani, and Hundewale, 2012; Lin, 2010)

Cost as an attribute has been studied by (Sadi and Noordin, 2011), this study found out that perceived cost is also an important factor and has negative relation with the intention to adopt mobile banking services; therefore, this study suggests that the creative promotional and pricing strategies, including cost reduction should be implemented to attract more price-conscious customers. Singh, Srivastava, and Srivastav (2010), also argued that the financial cost incurred has a negative effect on the intention to use mobile banking.

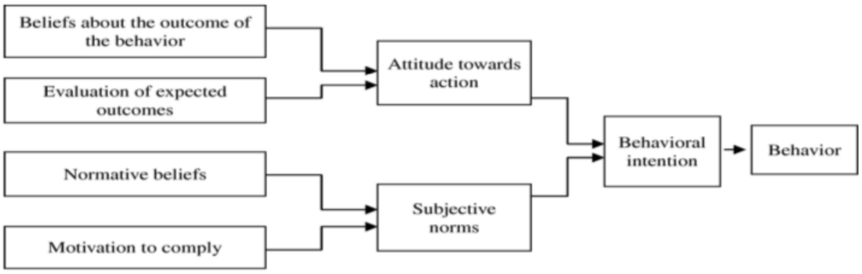
Researchers have come across many different models that help them in determining the important factors that affect the attitude and intention of the mobile banking users. In the next section those models have been discussed.

2.1.9.1 TRA, TAM, TPB, IDT and UTUAT Model

There are various models that help in study of adoption behaviour of mobile banking services. These models include various attributes that judge the intention of the mobile banking user and his/her attitude towards it. These models are: 1) Theory of Reasoned Action (TRA) 2) Technology Acceptance Model (TAM) 3) Theory of Planned Behaviour (TPB) 4) Innovation Diffusion Theory (IDT) 5) Unified Theory of Acceptance and Use of Technology Model (UTUAT).

2.1.9.2 Theory of Reasoned Action (TRA)

In the model proposed by Fishbein and Ajzen (1975) (see Figure) it was suggested that person's actual behaviour can be determined by the behavioural intention along with the belief and subjective norms that the person has for the behaviour. Subjective norms refer to "an individual's perception of other's opinion abo ut his/her particular behaviour, if he should perform a particular behaviour or not" and attitude towards action is defined as a person's positive or negative attitude towards this performed behaviour. Thus, TRA is a useful model that can explain the actual behaviour of an individual. In 1985 Davis took the same model and extended it to the TAM and linked it to the user acceptance of an information system.



Source: Fishbein and Ajzem, 1975

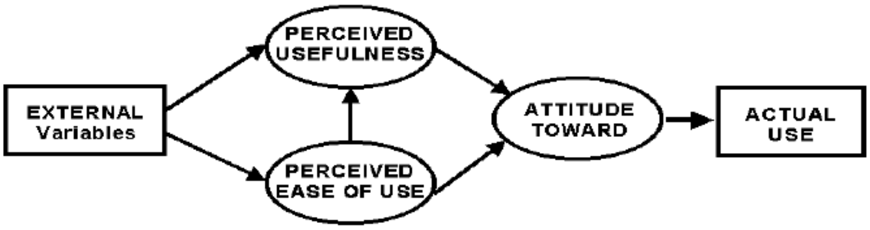
Figure 2.4: Theory of Reasoned Action (TRA)

2.1.9.3 Technology Acceptance Model (TAM)

Technology Acceptance Model (TAM) proposed by Fred Davis in 1986 (see Figure 6). Davis (1986) defined Perceived usefulness as " The degree to which an individual believes that using the particular system would enhance his or her performance" and Perceived ease of use is defined as "the degree to which a person believes that using a particular system would be free of effort" . According to him attitude of the user towards the acceptance of new technology or information system is determined by perceived usefulness and perceived ease of use.

2.1.9.4 Theory of Planned Behaviour (TPB)

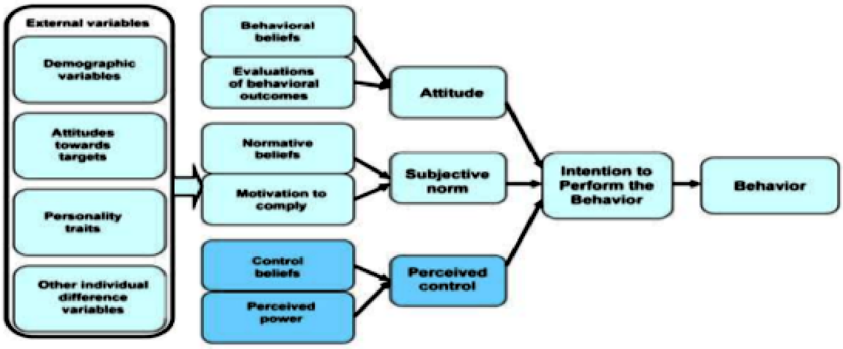
Theory of Planned Behaviour is an extension to TRA, it (see Figure 7) has taken into account one additional construct i.e. Perceived Behavioural Control (PBC). Perceived behavioural control refers to the people's perceptions of their ability to perform a



Source: Davis 1986, p. 24

Figure 2.5: Technology Acceptance Model (TAM)

given behaviour in a controlled manner. PBC is further influenced by control beliefs and perceived Power or perceived facilitation. Control beliefs refer to the perceived presence of those factors that may facilitate or impede the performance of behaviour. Perceived power specifies the power to have the resources that are required to use a specific system.



Source: Ajzen, 1991

Figure 2.6: Theory of Planned Behaviour (TPB)

2.1.9.5 Innovation Diffusion Theory (IDT)

Rogers (2003) described the innovation-diffusion process as “an uncertainty reduction process” (p. 232) and he proposes attributes of innovations that help to decrease uncertainty about the innovation.

Attributes of innovations include five characteristics of innovations:

- Relative advantage
- Compatibility
- Complexity
- Trialability

- Observability

Rogers (2003) stated that “individual’s perceptions of these characteristics predict the rate of adoption of innovations” (p. 219). Rogers (2003) defined the ra te of adoption as “the relative speed with which an innovation is adopted by members of a social system” (p. 221), Re lative advantage as “the degree to which an innovat ion is perceived as being better than the idea it supersedes” (p. 229), “compatibility is the degree to which an innovatio n is perceived as consistent with the existing values, past experiences, and needs of potential adopters” (p. 15), compl exity as “the degree to which an innovation is perceived as relatively difficult to understand and use” (p. 15), “trialability is the degree to which an innovation may be experimented with on a limited basis” (p. 16), observability as “the degree to which the results of an innovation are visible to others” (p. 16). To summarize, Roger argued that innovations that offer a more relative advantage, compatibility, simplicity, trialability, and observability will be adopted much faster as compare to others.

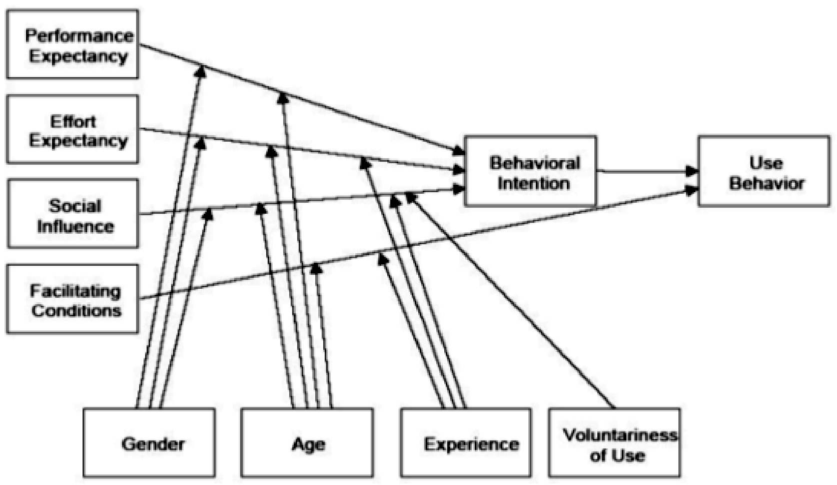
2.1.9.6 Unified Theory of Acceptance and Use of Technology Model (UTUAT) Model

This model is based on the theories of individual acceptance that are synthesized by Venkatesh, Morris, Davis, and Davis, (2003), include the Theory of Reasoned Action (TRA), Technology Acceptance Model (TAM), Motivational Model (MM), Theory of Planned Behaviour (TPB), Model Combining the Technology Acceptance Model and Theory of Planned Behaviour (C-TAM-TPB), Model of PC Utilization (MPCU), Innovation Diffusion Theory (IDT), and Social Cognitive Theory (SCT). Venkatesh (2003), (see Figure 8) defined Performance expectancy as the degree to which an individual believes that using the system will help him/her to attain gains in job performance, Effort Expectancy as the degree of ease associated with the use of the system, Social Influence as the degree to which an individual perceives that important others believe he or she should use the new system and Facilitating Conditions as the degree to which an individual believes that an organizational and technical infrastructure exists to support use of the system.

2.1.10 Discussion and Conclusion

In the backdrop of above reviewed literature, it can be seen that the adoption of mobile banking services in India is just 2 percent. So it becomes important for the service providers to increase the rate of adoption of mobile banking users. Through the literature review some important points have been highlighted. It includes:Banks should create awareness about the mobile banking services through Advertisements, Pamphlets, Demo Fares, Campaigning etc. so that the customer feel informed and it may create interest among them. S.Samudra and Phadtare (2012), claimed that the footfalls at ATM centres is likely to be very high, the campaigns may be carried out at these locations to attract more customers towards these services.

- Trust is also an important point of concern. Trust between the customers and the service provider is very important, without security and privacy users will not use mobile for financial transactions.



Source: Venkatesh et al. (2003)

Figure 2.7: Unified Theory of Acceptance and Use of Technology Model (UTUAT) Model

- Perceived ease of use and perceived usefulness are found to be important factors to influence the consumer intention to adopt mobile banking. Hence, the main attention of management should be focused on the development of usefulness of system, trust building and cost reduction.
- Perceived cost is also an important factor; therefore, this study suggests that the creative promotional and pricing strategies, including cost reduction should be implemented to attract more price-conscious customers.
- It is also found that customers will adopt mobile banking if they find it easy to use and understand.

The users who are using banking services on their mobiles are highly satisfied ones, because of several reasons. The first reason is the availability of facilities of balance checking, access to account and card statement, checking recent transactions, ordering of cheque books, blocking of lost cards, etc. In the earlier times customers used to stand in the long queue in banks for money transfer, money deposit etc. but now mobile banking is providing facilities of anytime and anywhere banking. Security in the mobile banking services is also enhanced by the introduction of OTP. Before the completion of any transaction you need to enter the OTP that is generated by the bank while the user is trying to initiate any mobile banking transaction and it is generated for one time use only as it expires after single use.The above review shows that to fulfill the expectations of the consumers and to increase the mobile banking users, mobile banking service provider needs to increase the awareness about the mobile banking services. Banks and the mobile service providers need to come together to bring a revolution in the field of mobile banking.

2.1.11 Comparison of Models

Table 2.2: Models and Theories

Models and Theories	Constructs
Theory of Reasoned Action (TRA) by Fishbein and Ajzen (1975) derives from psychology to measure behavioural intention and performance.	Attitude Subjective norm
Technology Acceptance Model (TAM) by Davis (1989) develops new scale with two specific variables to determine user acceptance of technology.	Percieved Usefulness Percieved Ease of Use
Theory of Planned Behaviour (TPB) by Ajzen (1991) extends TRA by including one more variable to determine intention and behaviour.	Attitude Subjective norm Percieved Behavioural Control
Combined TAM and TPB (C-TAM-TPB) by Taylor and Todd (1995).	Percieved Usefulness Percieved Ease of Use Attitude Subjective norm Percieved Behavioural Control
Innovation Diffusion Theory (IDT) by Rogers (1962) is adapted to information systems innovations by Moore and Benbasat (1991). Five attributes from Roger's model.	Relative Advantage Compatibility Complexity Observatibility Trialiability
Unified Theory of Acceptance and Use of Technology Model (UTAUT) by Venkatesh et al. (2003) integrates above theories and models to measure user intention and usage on technology.	Performance Expectancy Effort Expectancy Attitude toward Using Technology Social Influence Facilitating Conditions Self-Efficacy Anxiety

2.2 KYC-Literature Survey

2.2.1 Introduction

Know Your Customer (KYC) is the due diligence and bank regulation that financial institutions and other regulated companies must perform to identify their clients and ascertain relevant information pertinent to doing financial business with them. The Prevention of Money Laundering Act, 2002 and rules notified there under impose obligation on banking companies, financial institutions and intermediaries to verify identity of clients, maintain records and furnish information to Financial Intelligence Unit- India (FIU-IND). The goal of KYC is to enable banks to know and understand their customers better and help them manage their risks prudently against Anti Money Laundering (AML) and Combating Financing of Terrorism (CFT). KYC being a regulatory and legal requirement, the policies are framed by the financial institutions around the key elements such as Customer Acceptance Policy, Customer Identification Procedures, Monitoring of Transactions and Risk management in accordance with the Reserve Bank of India's directive in 2004.

2.2.2 KYC and Banking

We wish to review a study of various scholarly articles that helped us gather insights on the KYC requirement and its impact on Banking. As discussed in a paper Research on Know Your Customer (KYC) published by Prof. Venkatesh U. Rajput, the KYC guidelines of RBI mandate banks to collect three proofs from their customers. They are as follows:

1. Recent Photograph of the customer
2. Proof of Identity
3. Proof of Address

The study shows that KYC procedure needs to be adhered to by a customer during following instances:

- While opening an account in a bank
- While applying for a credit card or loan
- While opening a subsequent account
- Opening a locker facility
- When there are not enough documents with the bank in existing account
- When there are changes in signatories, beneficial owners, etc.
- When the bank feels it necessary to obtain additional information from existing customers based on conduct of the account
- While investing in a mutual fund

- Financial institutes may ask for a mandatory KYC process in other instances too

The research paper talks about the Enhanced Due Diligence (EDD) for successful on-boarding of the customers through the regulatory checks and Continuous Due Diligence (CDD) to monitor the activities of the customers continuously to see if they change markedly over time. This is indeed essential in Combating Financing of Terrorism (CFT) and preventing theft as also money laundering. Although the physical KYC is reliable, it is tedious, inefficient and poses various limitations in internet banking. It can be replaced with a more effective KYC process in the electronic mode, e-KYC.

2.2.3 e-KYC Process flow

The Aadhaar e-KYC service provides an instant, electronic, non-repudiable proof of identity and proof of address along with date of birth and gender. In addition, it also provides the resident's mobile number and email address to the service provider, which helps further streamline the process of service delivery. E-KYC may be performed at an agent location using biometric authentication, as well as remotely using an OTP on a website or mobile connection. The process flow is as given below:

(a) The interested resident authorizes UIDAI (through Aadhaar authentication) to provide their basic demographic data for PoI (Proof of Identity) and PoA (Proof of Address) along with their photograph (digitally signed) to service providers.

(b) The resident's record is first selected using the Aadhaar Number and then the demographic/biometric inputs are matched against the stored data which was provided by the resident during enrolment/update process. Another option for authentication can be done on the basis of the OTP sent to the registered mobile number.

(c) KYC front-end application captures Aadhaar number + biometric/OTP of resident.

(d) KUA forms the Auth XML using the PID block, signs it, and uses that to form KYC XML and signs it (if this is delegated to KSA, KSA also could form the KYC XML and sign it) sends to KSA

(e) KSA forwards the KYC XML (if KSA forms the KYC XML on behalf of KUA, KSA needs to form the KYC XML, and sign it) to Aadhaar KYC API

(f) Aadhaar KYC service authenticates the resident and if successful responds with digitally signed and encrypted demographic and photograph in XML format

(g) Demographic data and photograph in response is encrypted with either KSA or KUA public key (based on the setup at CIDR)

(h) KSA sends the response back to KUA enabling paper-less electronic KYC

(i) For security reason data collected for Aadhaar KYC must not be stored in the devices or log files. It's essential for ASA and AUA to maintain audit records for all the authentication request metadata along with the response

(j) KYC front-end application must ensure it takes an explicit "resident consent" authorizing the AUA to retrieve the resident data. Only if the resident has provided the consent (in the application UI, either in self-service mode or operator should prompt the resident and get consent), this should be populated as "Y". No other values are valid.

(k) The process can be for confirmation of proof of identity or confirmation of the information provided by the resident.

(l) Resident's privacy is of utmost importance, hence in the Aadhaar authentication service can only respond with a 'yes/no' nothing more, nothing less.

(m) No Personal Identity Information is returned as part of the response.

2.2.4 Response

- The encrypted response is just "0" or "-1".
- If the status is "0", it means that the encrypted response data is valid.
- If the status is "-1", it means the data should not be decrypted and used
- There will be a unique alphanumeric response code for each request received by CIDR.
- The AUA is expected to store this for future reference for handling any disputes.
- Aadhaar KYC server will retain KYC trail only for a short period of time as per UIDAI policy.

Like all digital interaction, there are failure chances too. In case of a failure, an error code is generated. Typical failure codes are:

- "K-100" – Resident authentication failed
- "K-200" – Resident data currently not available
- "K-540" – Invalid KYC XML
- "K-541" – Invalid KYC API version
- "K-542" – Invalid resident consent ("rc" attribute in "Kyc" element)
- "K-543" – Invalid timestamp ("ts" attribute in "Kyc" element)
- "K-544" – Invalid resident auth type ("ra" attribute in "Kyc" element does not match what is in PID block)
- "K-545" – Resident has opted-out of this service
- "K-551" – Invalid "Txn" namespace

- “K-569” – Digital signature verification failed for KYC XML (means that authentication request XML was modified after it was signed)
- “K-570” – Invalid key info in digital signature for KYC XML (it is either expired, or does not belong to the AUA or is not created by a well-known Certification Authority)
- “K-600” – AUA is invalid or not an authorized KUA
- “K-601” – ASA is invalid or not an authorized KSA
- “K-602” – KUA encryption key not available
- “K-603” – KSA encryption key not available
- “K-999” – Unknown error

The UIDAI released API– VERSION 1.0 can primarily be used in the following scenarios:

2.2.4.1 New customer/beneficiary

(a) KUA captures resident authentication data and invokes the Aadhaar e-KYC API through a KSA network;

(b) KYC data returned within the response of the e-KYC API is digitally signed and encrypted by UIDAI;

(c) Using the resident data obtained through this KYC API, the agency can provision the service instantaneously.

2.2.4.2 Existing customer/beneficiary

(a) The KUA captures resident authentication data and invokes the Aadhaar e-KYC API through a KSA network;

(b) The KYC data returned within the response of the e-KYC API is digitally signed and encrypted by UIDAI;

(c) Since the resident is already a customer/beneficiary, the KUA can use a simple workflow to approve the Aadhaar linkage by comparing data retrieved through the e-KYC API against what is on record (in paper or electronic form);

(d) Once verified, the existing customer/beneficiary record can be linked to the Aadhaar number.

The Aadhaar e-KYC API returns data along with a unique transaction code. The fact that the data is digitally signed by UIDAI and that every transaction has a unique code makes it possible to perform an electronic audit at a later point in time for any particular transaction. The Aadhaar e-KYC service does not compromise security for inclusion, and instead offers a solution that is secure as well as inclusive and protects data privacy by eliminating paper trail on the field. The policy note highlights the salient features of the service: Paperless, Consent based, Eliminates Document Forgery, Inclusive, Secure and compliant with the IT Act, Non-repudiable, Low cost, Instantaneous, Machine Readable, and Regulation friendly. The following figure 2.1 demonstrates e-KYC process flow

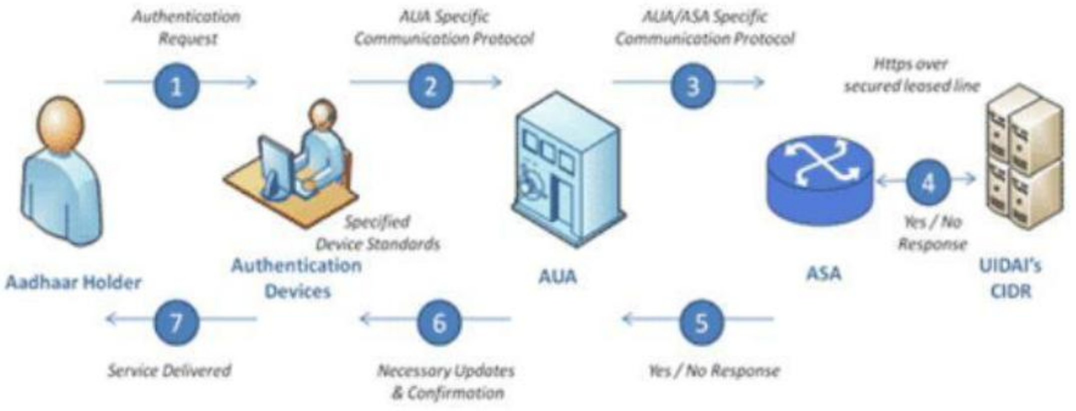

Authentication Request
1
AUA Specific Communication Protocol
2
AUA/ASA Specific Communication Protocol
3
Https over secured leased line
4
Yes / No Response
Specified Device Standards
Aadhaar Holder
Authentication Devices
AUA
ASA
UIDAI's CIDR
7
Service Delivered
6
Necessary Updates & Confirmation
5
Yes / No Response

Figure 2.8: e-KYC process flow

The important stakeholders in the process as seen in the Fig: 2.8 are:

- Unique Identification Authority of India (UIDAI)
- Authentication Service Agency (ASA)
- Authentication User Agency (AUA)
- Sub Authentication User Agency (AUA)
- Authentication Device Technology Service Provider
- Aadhaar holders

Refer Fig: 2.9 for the e-KYC data flow as mentioned in the policy note

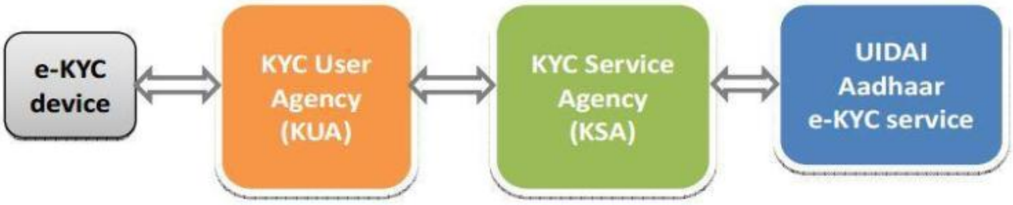


Figure 2.9: e-KYC data flow

2.2.5 Conclusion

Secure Service Delivery is pivotal to the risk management within financial institutions. The Aadhaar e-KYC service can revolutionize service delivery in the public and private sector as it does not trade-off security for convenience and inclusion and instead provides a solution that is secure, convenient, and inclusive.

2.3 Intelligent Chat Bot for Banking System

2.3.1 Overview of chat bot in m-suvidha.

An intelligent chat bot will be used to give information or answers to any question asked by user related to bank. Our Intelligent system will first take input from bank customer. This input will be taken as text in written format. According to input, intelligent system will processes the query and give response to user. An artificial intelligence is most important and helpful part of our project. Intelligent system is automation of activities associated with human thinking, decision making, and problem solving process. This system will be available on android. Our system will represent the design and development of an intelligent chat bot. It will present a technology demonstrator to verify a proposed framework required to support such a bot (an android service). While a black box approach is used, by controlling the communication structure, to and from the android-service, the android-service allows all types of clients to communicate to the server from any platform. The service provided will be accessible through a generated interface which allows for seamless Java processing; whereby the extensibility improves the lifespan of such a service. By introducing an artificial brain, the android-based bot generates customized user responses, aligned to the desired character. Questions asked to the bot, which will not be understood, are further processed using a third-party manual system, and the response will be archived, improving the artificial brain capabilities for future generation of responses

2.3.2 Technologies involved.

- ICB (Intelligent Chat Bot)
- Java core library.
- AI(Artificial Intelligent).

2.3.3 Brief introduction to chat bot for banking system

This banking bot project is built using artificial algorithms that analyses user's queries and understand user's message. We are going to design system for banks where users can ask any bank related questions like ekyc process,ekyc rules,account, policy etc. This application will be developed for android users. The system will recognizes user's query and understands what he wants to convey and simultaneously answers them appropriately. Even if the user does not frame sentence properly system will understands the query and answer accordingly. There is no specific format the user must follow to ask questions. The built in artificial intelligence system realizes users requirements and provides suitable answers to the user. The purpose of a chat bot system is to simulate a human conversation; the chat bot architecture integrates a language model and computational algorithm to emulate information chat communication between a human user and a computer using natural language.

With the improvement of data-mining and machine-learning techniques, better decision-making capabilities, availability of corpora, robust linguistic annotations/processing tools standards like XML and its applications, chat bots have become more practical in daily life applications such as help desk tools, information retrieval tools, automatic telephone answering systems, advertising, tools to aid in education, business and Ecommerce. In E-commerce, chat bot helps in information retrieval tasks, such as for searching and browsing, as menu based navigation poses difficulties in locating the appropriate information. The dialogue system provides additional information on products and simplify decision making process to find a product that satisfy customer's requirements.According to Dr. Wallace, perhaps, the biggest market of chat bot is Entertainment Markets, in which, we can imagine that chat bots can act as a talking book for children and provide foreign language instruction or can be a tutor in Intelligent Tutoring system. One such study used an ALICE system to help Chinese university students practice their conversational English skills. The study was qualitative in nature and used pre-existing conversational English skills.

The review focused more on user attitudes rather than on chatter bot efficiency. It was discovered that 62 percent of users chatted for 10 lines or less, and that 8.5 percent of the time ALICE bot has no specific pattern to match the given input and had to rely on rootlevel generic responses. In all of these conversational entities, one thing is common; and that is, they are having the difficulty of maintaining dialogue for sustainable period of time. Another tutoring study focused on using ALICE as a course enhancement tools with Social and Political Theory knowledge .Chatter bot development is reasonably well studied ever since the Turing Imitation Game (TIG) was first proposed. Eliza was the first famous chat bot, and ALICE was another milestone. The Loebner Prize and The Chatterbox Challenge are both annual competitions which have their roots in TIG. However, these are typically text only experiments, although some limited visual components are often added. This focus is on; however, whether with the text exchange alone, we can replicate human “behaviour” .

This literature review found that most subjects used the system as a search engine rather than as a conversation partner. It was further concluded that their system was unable to function as a stand-alone tutor. Dialog system can adequately carry out the conversations with the user and can log the conversations which can be good source for knowledge acquisition for domain specific topic . Therefore, techniques of knowledge acquisition were rightly used in their system AZ-ALICE chat bot that is an extension in ALICE chatter bot.

Chapter 3

System Analysis and Design

3.1 Analysis and Design of the System

3.1.1 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. The class diagram is the main building block of object oriented modelling. It is used both for general conceptual modelling of the systematics of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed.

In the diagram, classes are represented with boxes which contain three parts:

- The upper part holds the name of the class
- The middle part contains the attributes of the class
- The bottom part gives the methods or operations the class can take or undertake.

In the design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the static relations between those objects. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses.
In order to further describe the behavior of systems, these class diagrams can be complemented by state diagram or UML state machine.

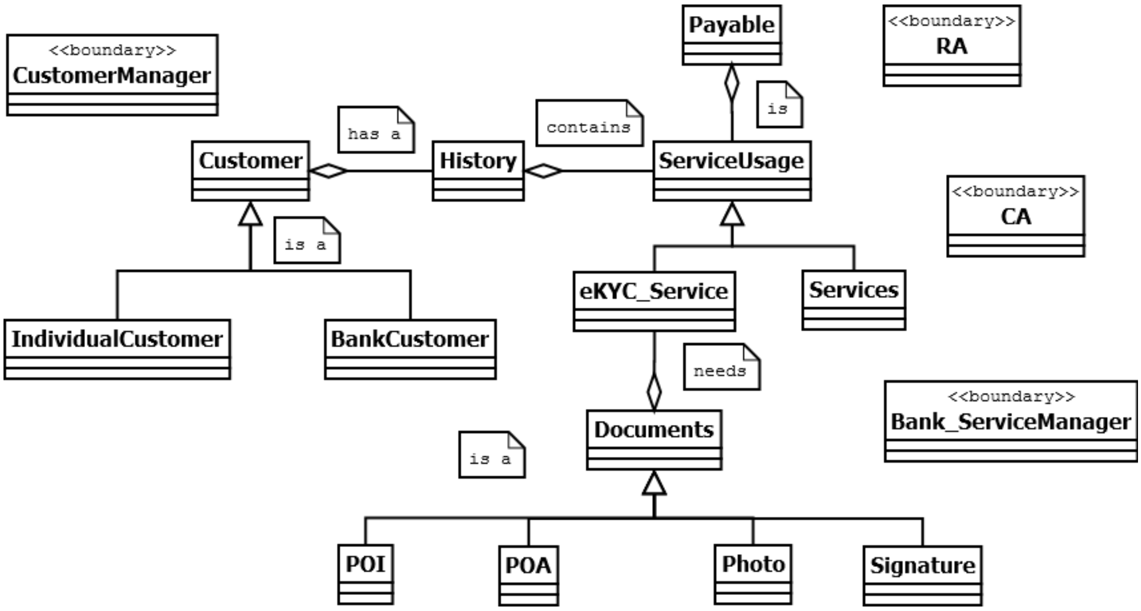
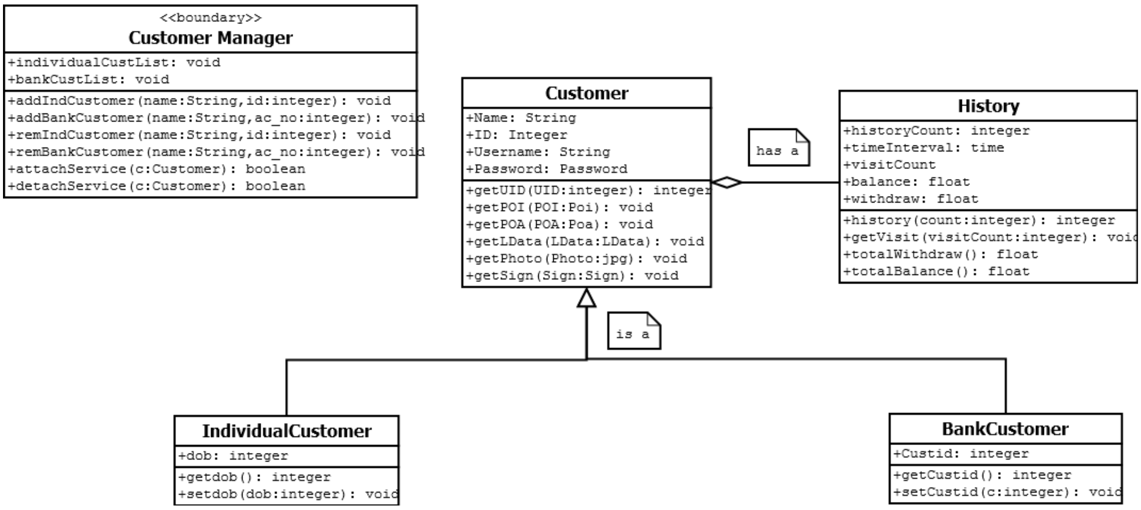


Figure 3.1: Class Overview



<<boundary>>
Customer Manager
+individualCustList: void
+bankCustList: void
+addIndCustomer(name:String,id:integer): void
+addBankCustomer(name:String,ac_no:integer): void
+remIndCustomer(name:String,id:integer): void
+remBankCustomer(name:String,ac_no:integer): void
+attachService(c:Customer): boolean
+detachService(c:Customer): boolean
Customer
+Name: String
+ID: Integer
+Username: String
+Password: Password
+getUID(UID:integer): integer
+getPOI(POI:Poi): void
+getPOA(POA:Poa): void
+getLData(LData:LData): void
+getPhoto(Photo:jpg): void
+getSign(Sign:Sign): void
has a
History
+historyCount: integer
+timeInterval: time
+visitCount
+balance: float
+withdraw: float
+history(count:integer): integer
+getVisit(visitCount:integer): voi
+totalWithdraw(): float
+totalBalance(): float
is a
IndividualCustomer
+dob: integer
+getdob(): integer
+setdob(dob:integer): void
BankCustomer
+Custid: integer
+getCustid(): integer
+setCustid(c:integer): void

Figure 3.2: Detailed Customer Class Diagram

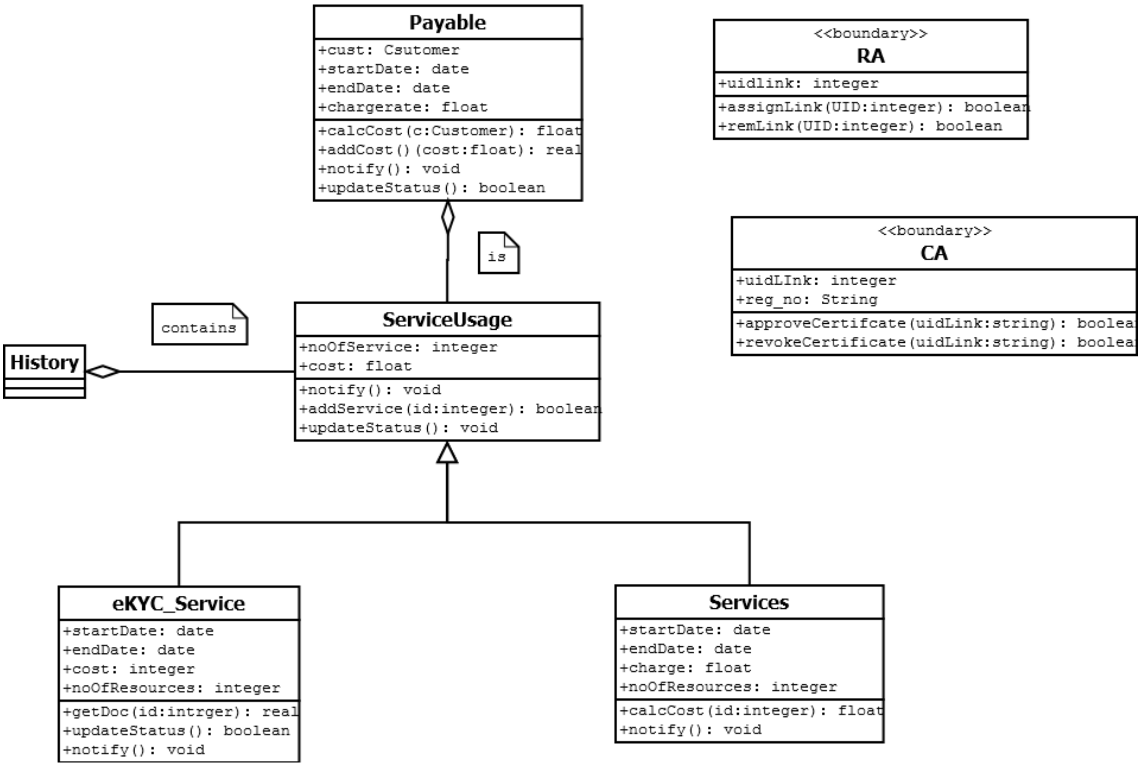


Figure 3.3: Detailed Service Class Diagram

3.1.2 Use case Diagram

Use Case Diagram: Use case diagrams model the functionality of a system using actors and use cases. Use cases are services or functions provided by the system to its users.

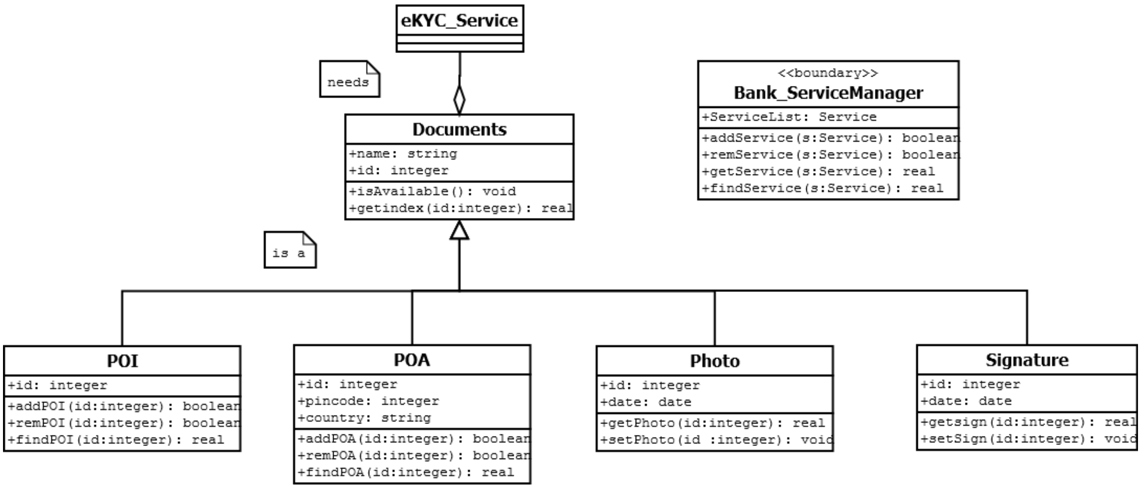

eKYC_Service
needs
<<boundary>>
Bank_ServiceManager
+ServiceList: Service
+addService(s:Service): boolean
+remService(s:Service): boolean
+getService(s:Service): real
+findService(s:Service): real
Documents
+name: string
+id: integer
+isAvailable(): void
+getindex(id:integer): real
is a
POI
+id: integer
+addPOI(id:integer): boolean
+remPOI(id:integer): boolean
+findPOI(id:integer): real
POA
+id: integer
+pincode: integer
+country: string
+addPOA(id:integer): boolean
+remPOA(id:integer): boolean
+findPOA(id:integer): real
Photo
+id: integer
+date: date
+getPhoto(id:integer): real
+setPhoto(id :integer): void
Signature
+id: integer
+date: date
+getsign(id:integer): real
+setSign(id:integer): void

Figure 3.4: Detailed Documents Class Diagram

The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified. So in brief, the purposes of use case diagrams can be as follows:
1. Used to gather requirements of a system.
2. Used to get an outside view of a system.
3. Identify external and internal factors influencing the system.
4. Show the interacting among the requirements are actors.

Notations used:
System:
Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.

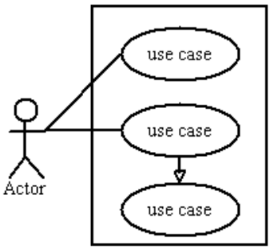


Figure 3.5: system

Use Case:
Draw use cases using ovals. Label with ovals with verbs that represent the system's functions.

Figure 3.6: use case

Actors:
Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.

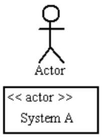


Figure 3.7: actor

Relationships:
Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either ”uses” or ”extends.” A ”uses” relationship indicates that one use case is needed by another in order to perform a task. An ”extends” relationship indicates alternative options under a certain use case.

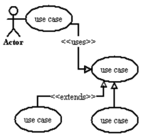


Figure 3.8: relations

The following is the e-KYC Use case Diagram:

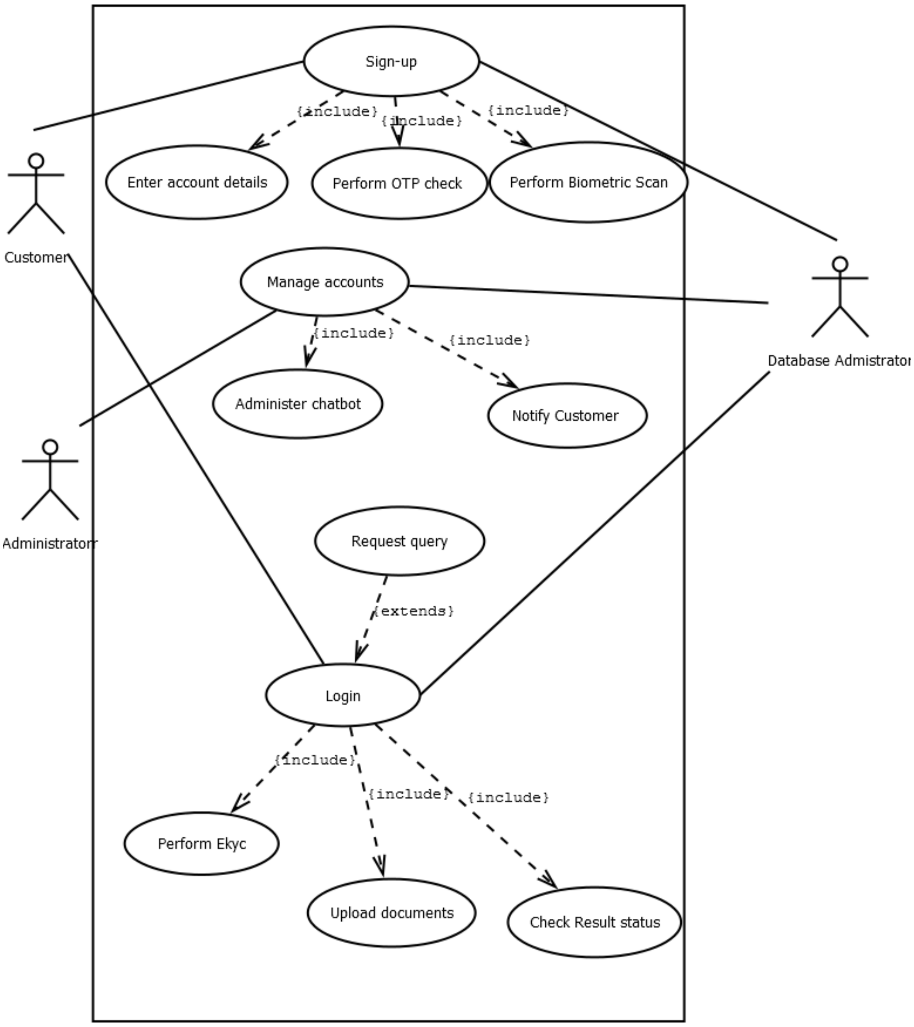


Figure 3.9: e-KYC Use case Diagram

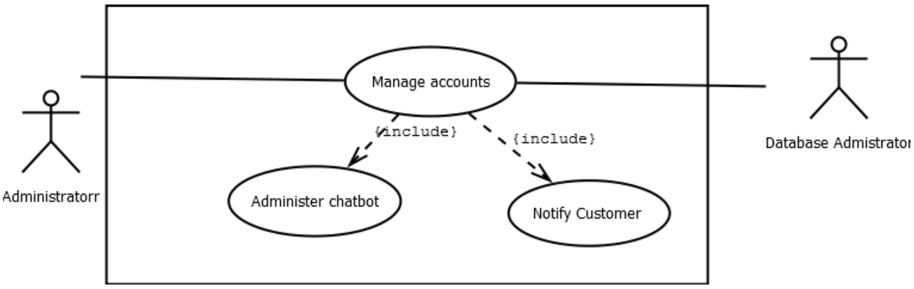


Figure 3.10: e-KYC Use case DB Admin Diagram

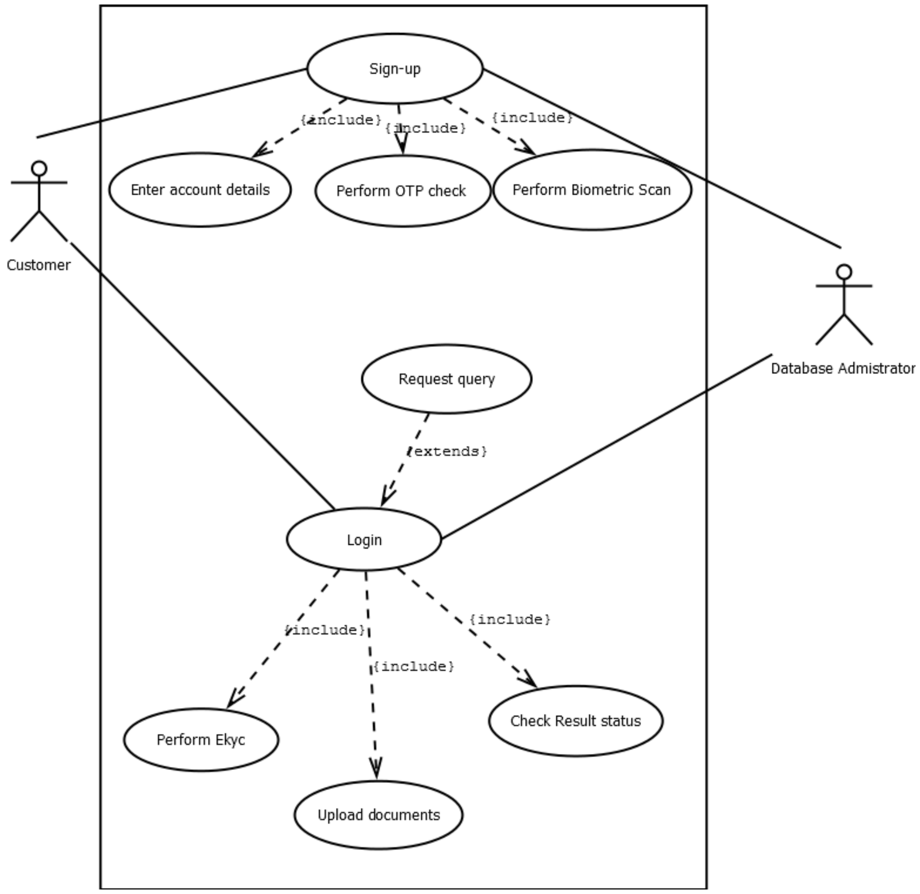


Figure 3.11: e-KYC Use case Customer DB Admin Diagram

3.1.3 Sequence Diagrams

3.1.3.1 Sequence diagram

Sequence Diagrams: Sequence diagram is the most common kind of interaction diagram, which focuses on the message interchange between a numbers of lifelines. Sequence diagram describes an interaction by focusing on the sequence of messages that are exchanged, along with their corresponding occurrence specifications on the lifelines. Sequence diagram emphasizes on time sequence of messages. Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. The following nodes and edges are typically drawn in a UML sequence diagram: lifeline, execution specification, message, combined fragment, interaction use, state invariant, continuation, destruction occurrence. Before drawing the sequence diagram, below things needs to be identified:
-Objects taking part in the interaction.
-Message flows among the objects.
-The sequence in which the messages are flowing.
-Object organization.

Notation used:
1. Class Roles
Class roles describe the way an object will behave in context. Use the UML object

symbol to illustrate class roles, but don't list object attributes.

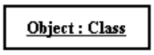


Figure 3.12: class roles

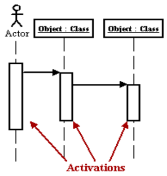


Figure 3.13: Activation

2. Message

Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.

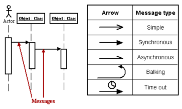


Figure 3.14: Message

3. Life Lines

Lifelines are vertical dashed lines that indicate the object's presence over time.

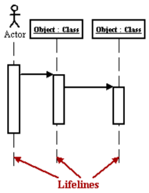


Figure 3.15: Life Line

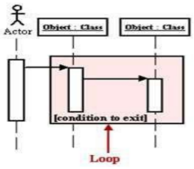


Figure 3.16: Loops

4. Loops
A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets [ ].

3.1.3.2 Sequence diagram of app signup

The following is the Sequence diagram app signup for e-KYC:

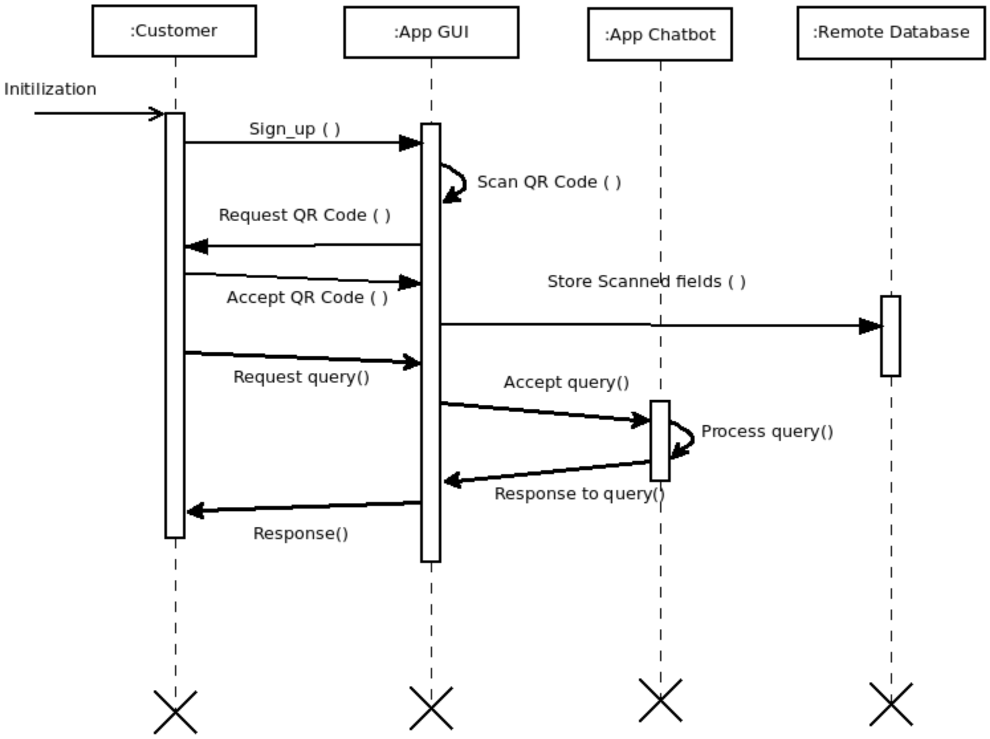


Figure 3.17: Sequence diagram app signup for e-KYC

3.1.3.3 Sequence diagram for E-KYC

The following is the Sequence diagram for E-KYC:

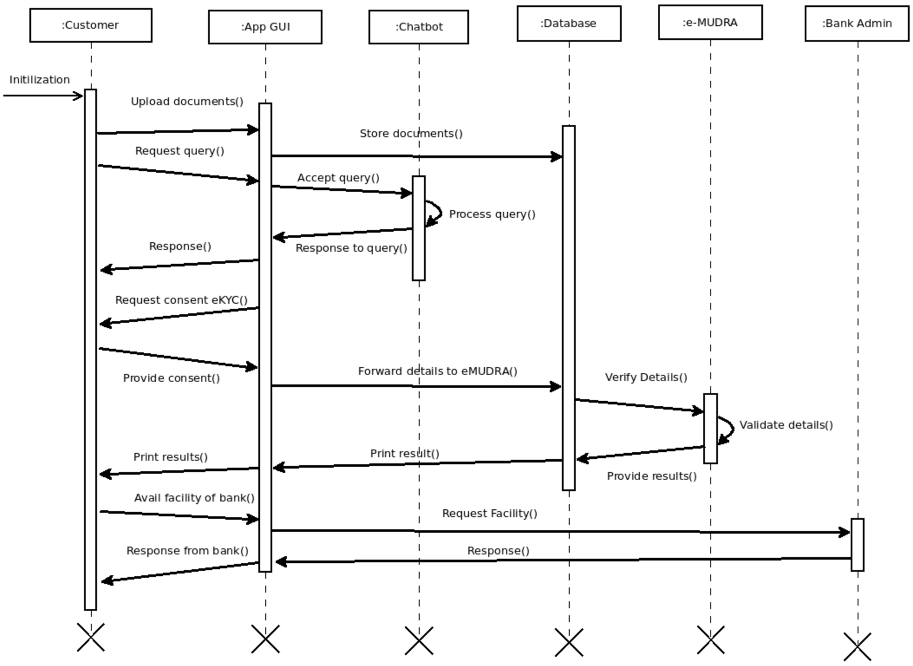


Figure 3.18: Sequence diagram for E-KYC

3.1.3.4 Sequence diagram for Login of e-KYC

The following is the Sequence diagram for Login of e-KYC:

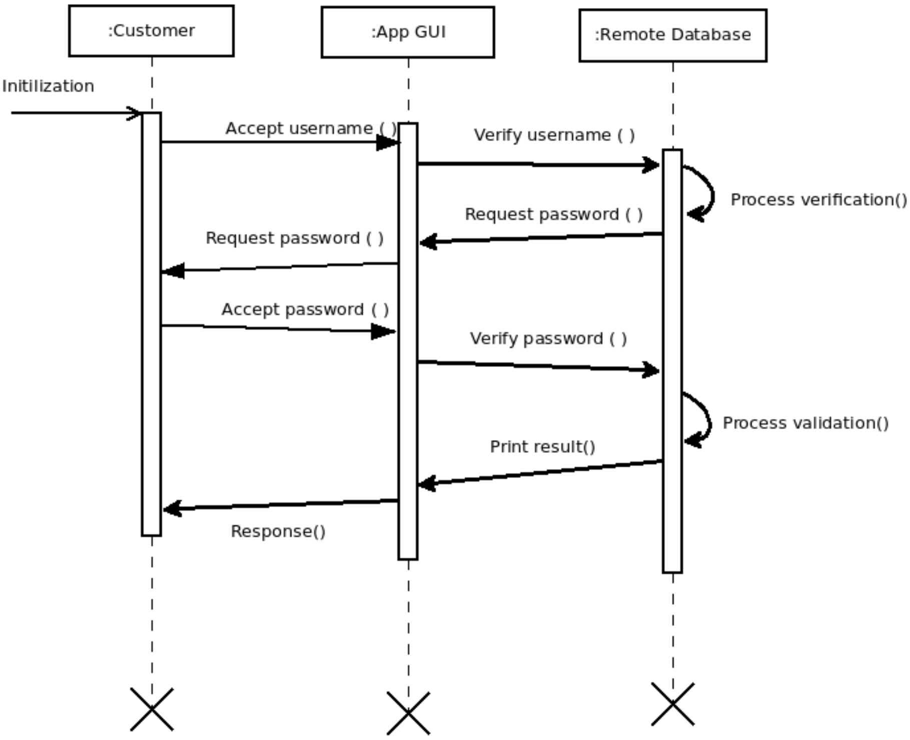


Figure 3.19: Sequence diagram for Login of e-KYC

3.1.4 State Diagram

The following is the e-KYC State Diagram:

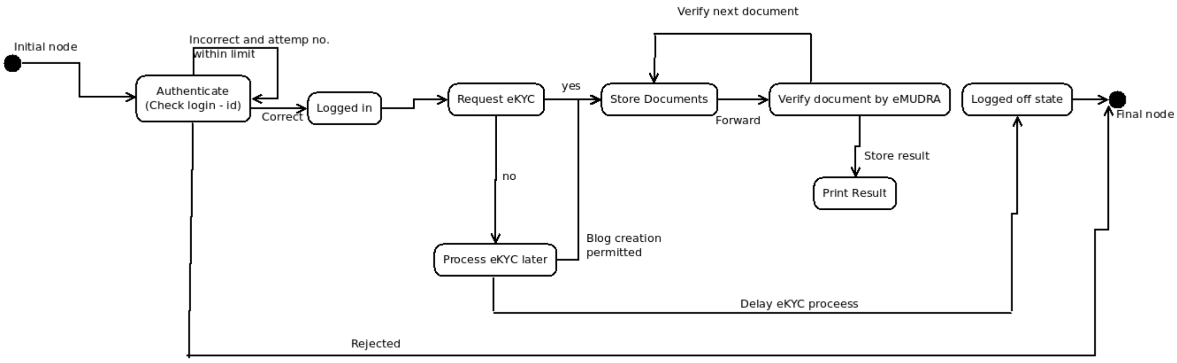


Figure 3.20: e-KYC State Diagram

3.1.5 Collaboration Diagram

Collaboration Diagram
Collaboration Diagram is a type of interaction diagram. It shows the object organization among the system. It describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behavior of a system. Before drawing the collaboration diagram, below things should be defined:
- Objects taking part in the interaction.
- Message flows among the objects.
- The sequence in which the messages are flowing.
- Object organization.
Notation used:
1. Class Roles
Class roles describe how objects behave. Use the UML object symbol to illustrate class roles, but don't list object attributes.

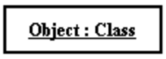


Figure 3.21: Class Role

2. Association Roles
Association roles describe how an association will behave given a particular situation. You can draw association roles using simple lines labeled with stereotypes.

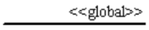


Figure 3.22: Association Role

3. Messages

Unlike sequence diagrams, collaboration diagrams do not have an explicit way to denote time and instead number messages in order of execution. Sequence numbering can become nested using the Dewey decimal system. For example, nested messages under the first message are labeled 1.1, 1.2, 1.3, and so on. The condition for a message is usually placed in square brackets immediately following the sequence number. Use a * after the sequence number to indicate a loop.

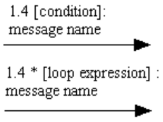


Figure 3.23: Messages

The following is the e-KYC Collaboration Diagram:

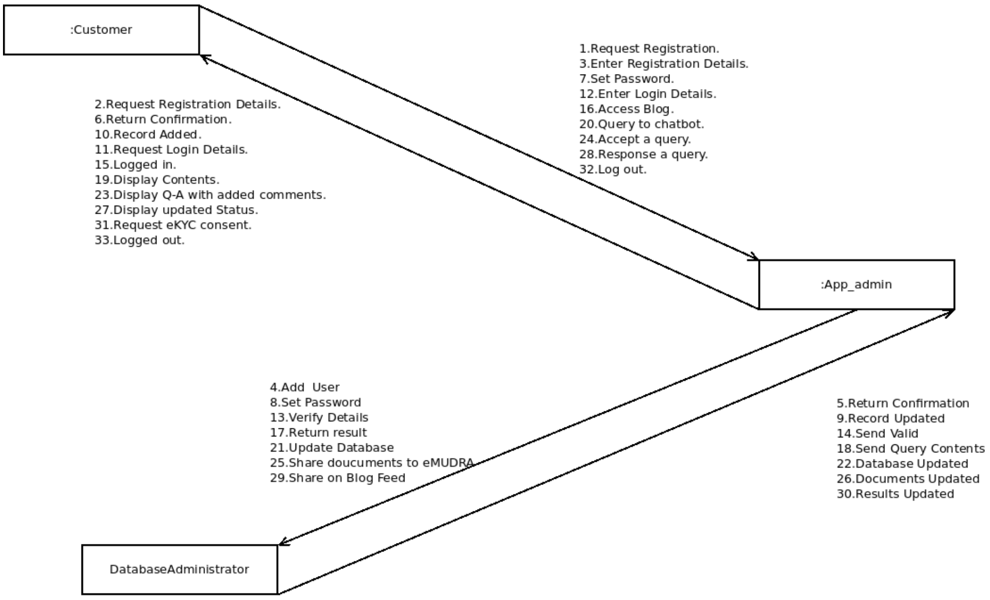


Figure 3.24: e-KYC Collaboration Diagram

3.1.6 Component Diagram

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

A component is something required to execute a stereotype function. Examples of stereotypes in components include executables, documents, database tables, files, and library files.
Components are wired together by using an *assembly connector*to connect the required interface of one component with the provided interface of another component. This illustrates the *service consumer - service provider relationship* between the two components.
An *assembly connector* is a "connector between two components that defines that one component provides the services that another component requires. An assembly connector is a connector that is defined from a required interface or port to a provided interface or port."
When using a component diagram to show the internal structure of a component, the provided and required interfaces of the encompassing component can delegate to the corresponding interfaces of the contained components.
A *delegation connector* is a "connector that links the external contract of a component (as specified by its ports) to the internal realization of that behaviour by the component's parts."
The example above illustrates what a typical Insurance policy administration system might look like. Each of the components depicted in the above diagram may have other component diagrams illustrating its internal structure. The following is the e-KYC Component Diagram:

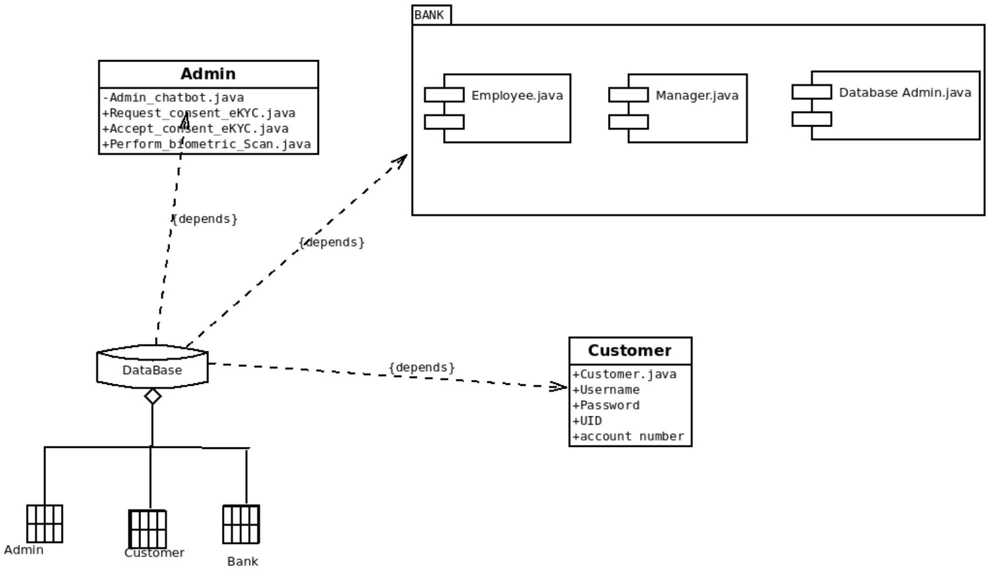


Figure 3.25: e-KYC Component Diagram

3.1.7 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. Purpose: The name Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. UML is mainly designed to focus on software artifacts of a system. But these two diagrams are special diagrams used to focus on software components and hardware components. So most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on hardware topology of a system.

Deployment diagrams are used by the system engineers. The purpose of deployment diagrams can be described as: Visualize hardware topology of a system. Describe the hardware components used to deploy software components. Describe runtime processing nodes. How to draw Deployment Diagram? Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardwares used to deploy the application. Deployment diagrams are useful for system engineers. An efficient deployment diagram is very important because it controls the following

- parameters
- Performance
- Scalability
- Maintainability
- Portability

So before drawing a deployment diagram the following artifacts should be identified:

- Nodes
- Relationships among nodes

The following deployment diagram is a sample to give an idea of the deployment view of order management system. Here we have shown nodes as:

- Monitor
- Modem
- Caching server
- Server

The application is assumed to be a web based application which is deployed in a clustered environment using server 1, server 2 and server 3. The user is connecting to the application using internet. The control is flowing from the caching server to the clustered environment. Where to use Deployment Diagrams?
Deployment diagrams are mainly used by system engineers. These diagrams are used to describe the physical components (hardwares), their distribution and association. To clarify it in details we can visualize deployment diagrams as the hardware components/nodes on which software components reside. Software applications are developed to model complex business processes. Only efficient software applications are not sufficient to meet business requirements. Business requirements can be described as to support increasing number of users, quick response time etc. To meet these types of requirements hardware components should be designed efficiently and in a cost effective way. Now a days software applications are very complex in nature. Software applications can be stand alone, web based, distributed, mainframe based and many more. So it is very important to design the hardware components efficiently.

So the usage of deployment diagrams can be described as follows:

- To model the hardware topology of a system.
- To model embedded system.
- To model hardware details for a client/server system.
- To model hardware details of a distributed application.
- Forward and reverse engineering.

The following is the e-KYC Component Diagram:

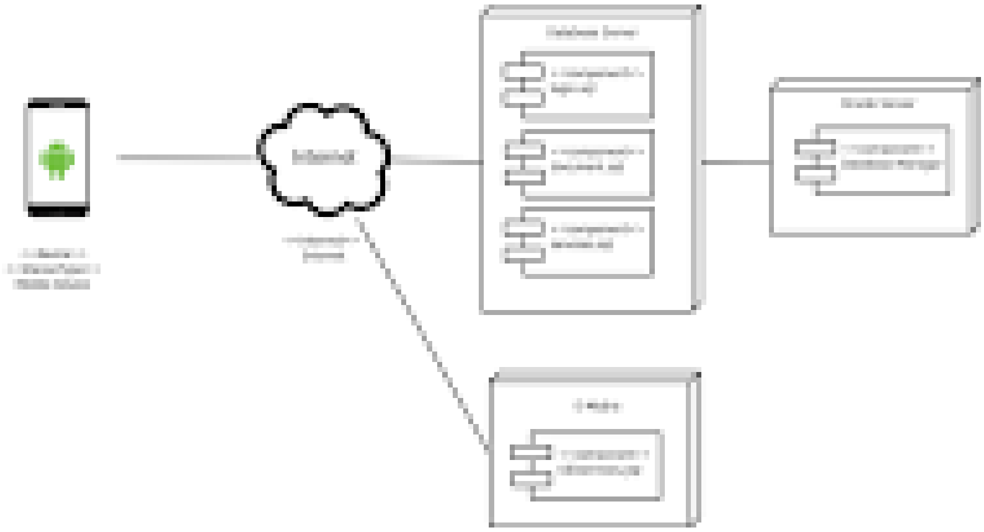
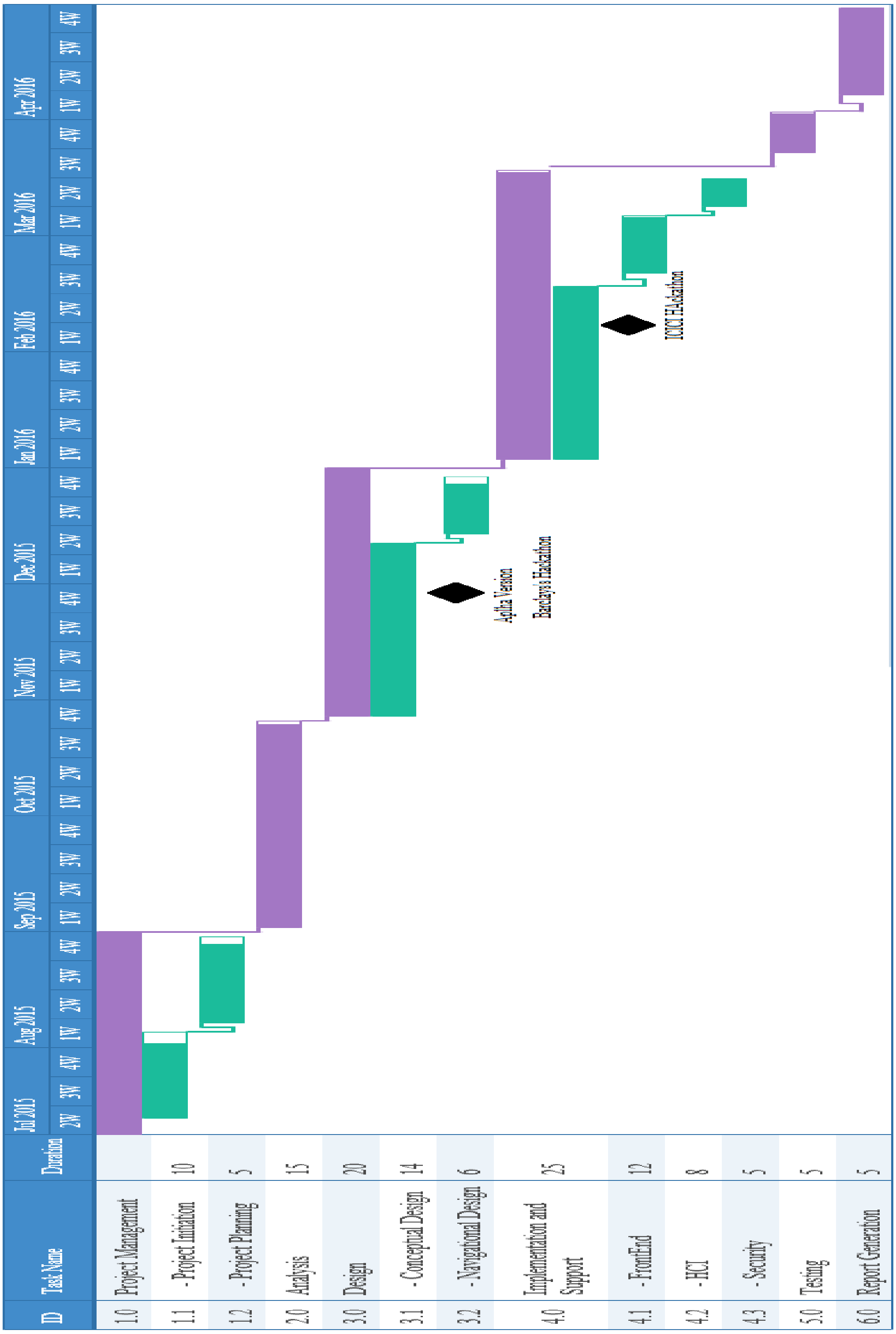


Figure 3.26: e-KYC Component Diagram

3.2 Time-line of the project m-Suvidha



ID	Task Name	Duration
1.0	Project Management	
1.1	- Project Initiation	10
1.2	- Project Planning	5
2.0	Analysis	15
3.0	Design	20
3.1	- Conceptual Design	14
3.2	- Navigational Design	6
4.0	Implementation and Support	25
4.1	- FrontEnd	12
4.2	- HCI	8
4.3	- Security	5
5.0	Testing	5
6.0	Report Generation	5

Figure 3.27: e-KYC Time-line for m-Suvidha

Chapter 4

Security of m-Suvidha

4.1 m-suvidha security

m-Suvidha is in compliance with the coding techniques for the PCIDSS Standards and BS ISO/IEC 27002:2013 m-Suvidha is has implemented the three level security mechanism:

- Device level security
- Communication and transport level security
- Data level security

4.1.1 Device level security

4.1.1.1 Captcha and Login password

Every user in m-suvidha needs a login and password to access any internal window or activity of m-suvidha. This username and password is validated on our server side to filter all unauthorized users to access the application.
There is no access at all for the trespassers or unauthorized users to internal data or usage of m-suvidha.
The password is salt-encrypted so no sql injection is possible. Every password value entered during registration is salt hashed and then sent to server. The password entered during login is first hashed by salt and the the unique hash is sent to server for validation.
In Msuvidha has three frames from which a user can validate itself. The captcha is placed at all the three window frames and its implementation is made sure to be accessed for every entry point to the msuvidha android application.

4.1.1.2 Communication and transport level security/ and also data level security.

To be in compliance especially with the PCIDSS standards the m-suvidha fintech android application uses PGP encryption scheme.
Implementation:
Each data value that enters the m-suvidha android application is encrypted at input extraction state and then passed in android application for internal flow and also in the database in the encrypted format.This ensures that no data ever entered into msuvidha is never accessible in the code to anyone who shall ever attempt to tamper the msuvidha application and crack the code.
The data is also not available to the developers or any person who is using the source or the source data. Thus the internal and the external unauthorized users or personnel is prevented to access the private user data and also tampers it.
The data that is stored is in the database server of the application which is hosted on a cloud platform. Though it is secured yet we implement the extra feature of sending PGP encrypted values and store it in the same format in the database. This shall ensure that even if the database server is hacked into or a network breaching occurs the data is never in the extractable or intelligible form. It is just a bunch of insensible

data that shall be accessed and no frequency or pattern analysis can break the pgp.The password data infact is first pgp encrypted and then also salt encrypted. The two encryption makes it impossible for any network breach to access the login and password list and use it again m-suvidha.
The library of PGP is implemented from the openPGP group which is a standard Bouncy Castle libraries

4.2 BS ISO/IEC 27002:2013 Code of practice for information security controls

4.2.1 Overview

BS ISO/IEC 27002:2013 is the reference handbook for selecting controls for use within an Information Security Management System (ISMS) based on ISO/IEC 27001. It can also be used as a guidance document for any organization wishing to implement commonly accepted information security controls.
Since their conception in the early 1990s, globally recognized standards in Information Security have grown in rigor and recognition. So have information security threats and the best ways to prevent them. BS ISO/IEC 27002:2013 reflect current best practice for information security controls. It provides specific recommendations for possible controls and control objectives, together with comprehensive guidance on how to construct and apply each control.
Using BS ISO/IEC 27002:2013 will ensure your organization's information security measures align with current best practice.

4.2.2 Workflow

BS ISO/IEC 27002:2013 defines a wide range of potential security controls. Each potential control is carefully defined, followed by implementation guidance and other relevant information.
BS ISO/IEC 27002:2013 uses a structured approach, whereby similar or related controls are grouped together into categories with a single control objective. These categories are then assigned to one of fourteen basic clauses, each of which addresses a particular aspect of information security.
Although BS ISO/IEC 27002:2013 is an essential component of building an ISMS based upon BS ISO/IEC 27001:2013, it can be used independently as a source of information security controls following other methodologies or even as a stand-alone guide to best practice information security.

4.2.3 Community who Implements this standard

Anyone who is planning to build, operate, audit or certify an ISMS based on BS ISO/IEC 27001:2013 needs BS ISO/IEC 27002:2013. It provides essential further detail on the controls checklist used in ISO/IEC 27001.
BS ISO/IEC 27002:2013 will also be of interest to anyone with an interest in information security management, or a general interest in information security measures.

4.2.4 Outline for ISO/IEC 27002:2013

4.2.4.1 The standard starts with 5 introductory chapters

- Introduction
- Scope
- Normative references
- Terms and definitions
- Structure of this standard

4.2.4.2 These are followed by 14 main chapters

- Information Security Policies
- Organization of Information Security
- Human Resource Security
- Asset Management
- Access Control
- Cryptography
- Physical and environmental security
- Operation Security- procedures and responsibilities, Protection from malware, Backup, Logging and monitoring, Control of operational software, Technical vulnerability management and Information systems audit coordination
- Communication security - Network security management and Information transfer.
- System acquisition, development and maintenance - Security requirements of information systems, Security in development and support processes and Test data
- Supplier relationships - Information security in supplier relationships and Supplier service delivery management
- Information security incident management - Management of information security incidents and improvements
- Information security aspects of business continuity management - Information security continuity and Redundancies
- Compliance - Compliance with legal and contractual requirements and Information security reviews

4.2.5 Implementation of ISO/IEC 27002

Here are a illustration of typical information security policies and other controls relating to three parts of ISO/IEC 27002. (Note: this is merely an illustration. The list of example controls is incomplete and not universally applicable.)

4.2.5.1 Physical and Environmental security

- Physical access to premises and support infrastructure (communications, power, air conditioning etc.) must be monitored and restricted to prevent, detect and minimize the effects of unauthorized and inappropriate access, tampering, vandalism, criminal damage, theft etc.
- The list of people authorized to access secure areas must be reviewed and approved periodically (at least once a year) by Administration or Physical Security Department, and cross-checked by their departmental managers.
- Suitable video surveillance cameras must be located at all entrances and exits to the premises and other strategic points such as Restricted Areas, recorded and stored for at least one month, and monitored around the clock by trained personnel.
- Access cards permitting time-limited access to general and/or specific areas may be provided to trainees, vendors, consultants, third parties and other personnel who have been identified, authenticated, and authorized to access those areas.
- The date and time of entry and departure of visitors along with the purpose of visits must be recorded in a register maintained and controlled by Site Security or Reception.
- Everyone on site (employees and visitors) must wear and display their valid, issued pass at all times, and must present their pass for inspection on request by a manager, security guard or concerned employee.
- Access control systems must themselves be adequately secured against unauthorized/inappropriate access and other compromises.
- Fire/evacuation drills must be conducted periodically (at least once a year).
- Smoking is forbidden inside the premises other than in designated Smoking Zones.

4.2.5.2 Human Resource security

- All employees must be screened prior to employment, including identity verification using a passport or similar photo ID and at least two satisfactory professional references. Additional checks are required for employees taking up trusted positions.
- All employees must formally accept a binding confidentiality or non-disclosure agreement concerning personal and proprietary information provided to or generated by them in the course of employment.

- Human Resources department must inform Administration, Finance and Operations when an employee is taken on, transferred, resigns, is suspended or released on long-term leave, or their employment is terminated.

- Upon receiving notification from HR that an employee's status has changed, Administration must update their physical access rights and IT Security Administration must update their logical access rights accordingly.

- An employee's manager must ensure that all access cards, keys, IT equipment, storage media and other valuable corporate assets are returned by the employee on or before their last day of employment, as a condition of authorizing their final pay....

4.2.5.3 Access control

- User access to corporate IT systems, networks, applications and information must be controlled in accordance with access requirements specified by the relevant Information Asset Owners, normally according to the user's role.

- Generic or test IDs must not be created or enabled on production systems unless specifically authorized by the relevant Information Asset Owners.

- After a predefined number of unsuccessful logon attempts, security log entries and (where appropriate) security alerts must be generated and user accounts must be locked out as required by the relevant Information Asset Owners.

- Passwords or pass phrases must be lengthy and complex, consisting of a mix of letters, numerals and special characters that would be difficult to guess.

- Passwords or pass phrases must not be written down or stored in readable format.

- Authentication information such as passwords, security logs, security configurations and so forth must be adequately secured against unauthorized or inappropriate access, modification, corruption or loss.

- Privileged access rights typically required to administer, configure, manage, secure and monitor IT systems must be reviewed periodically (at least twice a year) by Information Security and cross-checked by the appropriate departmental managers.

- Users must either log off or password-lock their sessions before leaving them unattended.

- Password-protected screensavers with an inactivity timeout of no more than 10 minutes must be enabled on all workstations/PCs.

- Write access to removable media (USB drives, CD/DVD writers etc.) must be disabled on all desktops unless specifically authorized for legitimate business reasons.

4.3 Payment Card Industry Data Security Standard

The Payment Card Industry Data Security Standard (PCI DSS) is a proprietary information security standard for organizations that handle branded credit cards from the major card schemes including Visa, MasterCard, American Express,Discover, and JCB. Private label cards – those which aren't part of a major card scheme – are not included in the scope of the PCI DSS. The PCI Standard is mandated by the card brands and administered by the Payment Card Industry Security Standards Council. The standard was created to increase controls around cardholder data to reduce credit card fraud. Validation of compliance is performed annually, either by an external Qualified Security Assessor (QSA) that creates a Report on Compliance (ROC) for organizations handling large volumes of transactions, or by Self-Assessment Questionnaire (SAQ) for companies handling smaller volumes.

4.3.1 History

PCI DSS originally began as five different programs: Visa's Cardholder Information Security Program, MasterCard's Site Data Protection, American Express' Data Security Operating Policy, Discover's Information Security and Compliance, and the JCB's Data Security Program. Each company's intentions were roughly similar: to create an additional level of protection for card issuers by ensuring that merchants meet minimum levels of security when they store, process and transmit cardholder data. The Payment Card Industry Security Standards Council (PCI SSC) was formed, and on December 15, 2004, these companies aligned their individual policies and released version 1.0 of the Payment Card Industry Data Security Standard (PCI DSS).
In September 2006, the PCI standard was updated to version 1.1 to provide clarification and minor revisions to version 1.0.
Version 1.2 was released on October 1, 2008. Version 1.1 "sunsetted" on December 31, 2008.
Version 1.2 did not change requirements, only enhanced clarity, improved flexibility, and addressed evolving risks and threats. In August 2009 the PCI SSC announced the move from
version 1.2 to version 1.2.1 for the purpose of making minor corrections designed to create more clarity and consistency among the standards and supporting documents.
Version 2.0 was released in October 2010[1] and is active for merchants and service providers from January 1, 2011 to December 31, 2014.
Version 3.0 was released in November 2013 and is active from January 1, 2014 to December 31, 2017.
Version 3.1 was released in April 2015, and will be retired 3 months after version 3.2 is released.
Version 3.2 is planned for release in March/April 2016.

4.3.2 Requirements

The PCI Data Security Standard specifies twelve requirements for compliance, organized into six logically related groups called "control objectives".

Table 4.1: Control objectives & PCI DSS requirements

Control objectives	PCI DSS requirements
Build and maintain a secure network	1. Install and maintain a firewall configuration to protect cardholder data 2. Do not use vendor-supplied defaults for system passwords and other security parameters
Protect cardholder data	3. Protect stored cardholder data 4. Encrypt transmission of cardholder data across open, public networks
Maintain a vulnerability management program	5. Use and regularly update anti-virus software on all systems commonly affected by malware 6. Develop and maintain secure systems and applications
Implement strong access control measures	7. Restrict access to cardholder data by business need-to-know 8. Assign a unique ID to each person with computer access9. Restrict physical access to cardholder data
Regularly monitor and test networks	10. Track and monitor all access to network resources and cardholder data 11. Regularly test security systems and processes
Maintain an information security policy	12. Maintain a policy that addresses information security

Each version of PCI DSS has divided these twelve requirements into a number of sub-requirements differently, but the twelve high-level requirements have not changed since the inception of the standard.

4.3.3 Compliance versus validation of compliance

Although the PCI DSS must be implemented by all entities that process, store or transmit cardholder data, formal validation of PCI DSS compliance is not mandatory for all entities. Currently both Visa and MasterCard require merchants and service providers to be validated according to the PCI DSS. Visa also offers an alternative program called the Technology Innovation Program (TIP) that allows qualified merchants to discontinue the annual PCI DSS validation assessment. These merchants are eligible if they are taking alternative precautions against counterfeit fraud such as the use of EMV or Point to Point Encryption (P2PE) technology, however they are still required to be PCI DSS compliant. Smaller merchants and service providers are not required to explicitly validate compliance with each of the controls prescribed by the PCI DSS although these organizations must still implement all controls in order to maintain safe-harbour and avoid potential liability in the event of fraud associated with theft of cardholder data.

Issuing banks are not required to go through PCI DSS validation although they still have to secure the sensitive data in a PCI DSS compliant manner. Acquiring banks are required to comply with PCI DSS as well as to have their compliance validated by means of an audit.
In the event of a security breach, any compromised entity which was not PCI DSS compliant at the time of breach will be subject to additional card scheme penalties, such as fines.

4.3.4 Salt overview

In cryptography, a salt is random data that is used as an additional input to a one-way function that ”hashes” a password or passphrase.[1] The primary function of salts is to defend against dictionary attacks versus a list of password hashes and against pre-computed rainbow table attacks.
A new salt is randomly generated for each password. In a typical setting, the salt and the password are concatenated and processed with a cryptographic hash function, and the resulting output (but not the original password) is stored with the salt in a database. Hashing allows for later authentication while protecting the plain-text password in the event that the authentication data store is compromised.
Cryptographic salts are broadly used in many modern computer systems, from Unix system credentials to Internet security.

$$\left(\sum_{n=1}^{6} 95^{n} + (24{,}029 \cdot 7) + (29{,}766 \cdot 8)\right) \mathrm{B} \cdot 4096 \approx 2767.9\,\mathrm{TiB}$$

Figure 4.1: Salt cryptography equation

4.3.5 Captcha

CAPTCHA: Telling Humans and Computers Apart Automatically A CAPTCHA is a program that protects websites against bots by generating and grading tests that humans can pass but current computer programs cannot. For example, humans can read distorted text as the one shown below, but current computer programs can’t:
The term CAPTCHA (for Completely Automated Public Turing Test To Tell Computers and Humans Apart) was coined in 2000 by Luis von Ahn, Manuel Blum, Nicholas Hopper and John Langford of Carnegie Mellon University. Guidelines If your website needs protection from abuse, it is recommended that you use a CAPTCHA. There are many CAPTCHA implementations, some better than others. The following guidelines are strongly recommended for any CAPTCHA code:

- Accessibility. CAPTCHAs must be accessible. CAPTCHAs based solely on reading text — or other visual-perception tasks — prevent visually impaired users from accessing the protected resource. Such CAPTCHAs may make a site

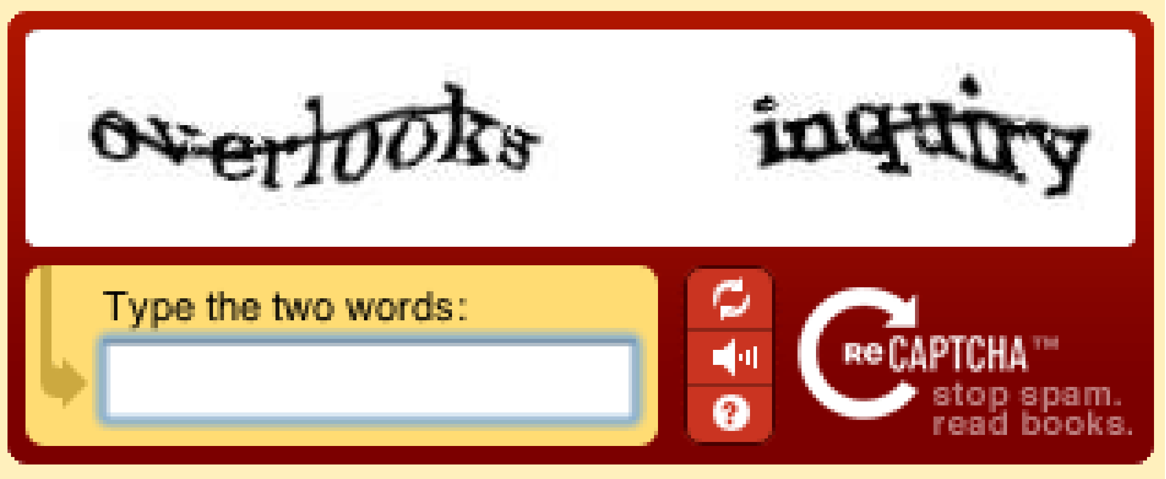


Figure 4.2: Captcha

incompatible with Section 508 in the United States. Any implementation of a CAPTCHA should allow blind users to get around the barrier, for example, by permitting users to opt for an audio or sound CAPTCHA.

- Image Security. CAPTCHA images of text should be distorted randomly before being presented to the user. Many implementations of CAPTCHAs use undistorted text, or text with only minor distortions. These implementations are vulnerable to simple automated attacks.

- Script Security. Building a secure CAPTCHA code is not easy. In addition to making the images unreadable by computers, the system should ensure that there are no easy ways around it at the script level. Common examples of insecurities in this respect include: (1) Systems that pass the answer to the CAPTCHA in plain text as part of the web form. (2) Systems where a solution to the same CAPTCHA can be used multiple times (this makes the CAPTCHA vulnerable to so-called ”replay attacks”). Most CAPTCHA scripts found freely on the Web are vulnerable to these types of attacks.

- Security Even After Wide-Spread Adoption. There are various ”CAPTCHAs” that would be insecure if a significant number of sites started using them. An example of such a puzzle is asking text-based questions, such as a mathematical question (”what is 1+1”). Since a parser could easily be written that would allow bots to bypass this test, such ”CAPTCHAs” rely on the fact that few sites use them, and thus that a bot author has no incentive to program their bot to solve that challenge. True CAPTCHAs should be secure even after a significant number of websites adopt them.

- Should I Make My Own CAPTCHA? In general, making your own CAPTCHA script (e.g., using PHP, Perl or .Net) is a bad idea, as there are many failure modes. We recommend that you use a well-tested implementation such as reCAPTCHA.

4.4 Pretty Good Privacy (PGP)

Pretty Good Privacy (PGP) is a data encryption and decryption computer program that provides cryptographic privacy and authentication for data communication. PGP is often used for signing, encrypting, and decrypting texts, e-mails, files, directories, and whole disk partitions and to increase the security of e-mail communications. It was created by Phil Zimmermann in 1991.

PGP and similar software follow the OpenPGP standard (RFC 4880) for encrypting and decrypting data.

4.4.1 Documentation

The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms, it was developed by the Legion of the Bouncy Castle - with a little help!

The Legion also gratefully acknowledges the contributions made to this package by others. The package is organised so that it contains a light-weight API suitable for use in any environment (including the newly released J2ME) with the additional infrastructure to conform the algorithms to the JCE framework.

Except where otherwise stated, this software is distributed under a license based on the MIT X Consortium license. To view the license, see here. The OpenPGP library also includes a modified BZIP2 library which is licensed under the Apache Software License, Version 1.1.

To view some examples, look at the test programs in the packages:

- org.bouncycastle.crypto.test
- org.bouncycastle.jce.provider.test
- org.bouncycastle.cms.test
- org.bouncycastle.mail.smime.test
- org.bouncycastle.openpgp.test
- org.bouncycastle.tsp.test

There are also some specific example programs for dealing with Attribute Certificates, PKCS12, SMIME and OpenPGP. They can be found in:

- org.bouncycastle.jce.examples
- org.bouncycastle.mail.smime.examples
- org.bouncycastle.openpgp.examples
- org.bouncycastle.x509.examples

Finally there are also code examples from Beginning Cryptography with Java which demonstrate both the use of the JCE/JCA and also some of the Bouncy Castle APIs such as for certificate generation, CMS and S/MIME. Note: the book was written to cover J2SE 5.0, while many of the examples will work with earlier JDKs, some will not

compile if you are not using J2SE 5.0 or later.
Unfortunately some examples in the book are now out of date (for 1.46 and above). You can also find some more uptodate documentation and examples at the guide project. To verify the packages, run the following Java programs with the appropriate classpath:

- java org.bouncycastle.crypto.test.RegressionTest
- java org.bouncycastle.jce.provider.test.RegressionTest

4.4.2 Code snippets

Code snippets implemented for encryption and decryption in m-suvidha

```
    public class Tester {

private static final String PASSPHRASE = ”test”;

private static final String DE_INPUT = ”src/test/x.pgp”;
private static final String DE_OUTPUT = ”src/test/x.txt”;
private static final String DE_KEY_FILE = ”src/test/secring.skr”;

private static final String E_INPUT = ”src/test/x.txt”;
private static final String E_OUTPUT = ”src/test/x.pgp”;
private static final String E_KEY_FILE = ”src/test/pubring.pkr”;

public static void testDecrypt() throws Exception {
PGPFileProcessor p = new PGPFileProcessor();
p.setInputFileName(DE_INPUT);
p.setOutputFileName(DE_OUTPUT);
p.setPassphrase(PASSPHRASE);
p.setSecretKeyFileName(DE_KEY_FILE);
System.out.println(p.decrypt());
}

public static void testEncrypt() throws Exception {
PGPFileProcessor p = new PGPFileProcessor();
p.setInputFileName(E_INPUT);
p.setOutputFileName(E_OUTPUT);
p.setPassphrase(PASSPHRASE);
p.setPublicKeyFileName(E_KEY_FILE);
System.out.println(p.encrypt());
}
}
```

Chapter 5

Artificial Intelligence: Chatterbot

5.1 Artificial Intelligence

Artificial is not just an important topic, but by far the most important topic of m-suvidha's element of artificial intelligence, the chat bot.
”We are on the edge of change comparable to the rise of human life on Earth ”
-VERNOR VINGER
Technology has progressed a lot, but humans, not so much.
The following are the basic questions about artificial intelligence.

WHAT IS ‘ARTIFICIAL INTELLIGENCE’?
It is the science and engineering of making intelligent machine Especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence. Scientists are saying,” A.I. is near ” which exactly means that the time is near when a programed artificial intelligence will have more smartness and will start to think like a human being. Some important questions that come to our mind are :
In the future, will A.I. control us or will it be like our friend? Well these questions arise alot after having a deep study on A.I. In the explanation above we mentioned Intelligence. But, do we know what really intelligence is Intelligence is the computational part or the ability to achieve goals in one's world.

WHEN DID IT ALL STARTED?
After world war II, a number of people started to work independently on intelligent machines. The first man who researched about intelligent machines was ‘ALAN TURING’. He gave a lecture on it in 1947, and spread the word to the world and he may have been the first to decide that A.I. can be best researched by programing computers than by building machines. And by 1950 scientists all around the globe started to focus on programing computers

TURING TEST
ALAN, also said about the conditions for considering a machine to be intelligent. He argued that if a machine could successfully pretend to be a human and have all the knowledge as a human, then you certainly consider it Intelligent. And this requires a test called TURNING TEST. ‘Turing test’ is organised every year in many big science centres like MIT media lab, Harvard University and many more. Sometimes without passing turing test, a machine can be considered intelligent.

Is A.I. all about simulating human intelligence ?
And here we go. Sometimes. But not always or even usually we can learn something about how you make machines to solve problems by observing people or just by observing their own method. On the other hand, most work in A.I., involves studing the problems that the world creates, rather than people or animals.

HUMAN LEVEL INTELLIGENCE
The ultimate effort is to make computer programs that can solve problems and achieve goals in the real world. However most A.I. researchers believe that new fundamental aids are required and therefore it can not be predicted when human level intelligence

will be achieved. Some people say that A.I. is a bad idea, as it is impossible to create A.I. according to them. The philosopher 'Jhon Searle' said that

"the idea of a non biological machine being intelligent is incoherent."

And the philosopher 'Hubert Dreyfus' said
A.I. is impossible.

On the other hand, most of interested scientist invested in A.I. and one of them is 'ELON MUSK'.'Elon musk' is the future of sci-tech, he is the real Tony Stark of this era. Elon Musk has donated millions for the future of life institute, and now his organization is putting that money to use by funding research to keep artificial intelligence "rebust and benificial". Musk, The CEO of Tesla Space X (we will discuss about it in future posts) has stated that its a concern of his. He brought up terminator in the past while discussing his concerns about the evolution of A.I. and stated that A.I. has the potential to be "MORE DANGEROUS THAN NUKES" Musk says 'A.I is our biggest existential threat' The scientist Stephen Hawkings and Elon Musk have signed an open letter to ensure that A.I. research will benifit Humanity.

HOW DOES A PROGRAM (A.I.) LEARN?
The simplest method is learning by trail and error. Let me explain you by an example. A program for solving mate in one chess, will play randomly until it is found that check mate is achieved. After that the AI remembers the successful move and next time the computer is given the same problem, it is able to produce the answer immediately. Further, the program will start thinking and understanding the way we are responding, it will try to analyze our strategy and what we do to win the game. Slowly the program will also produce its own technique which would be better and then it will win the chase for sure because now it started to THINK!

ARTIFICIAL INTELLIGENCE IN OUR DAILY LIFE
WHERE IS A.I.??
A.I. is a broad topic, it ranges from your phones to the self driving cars (we will discuss about it later). We can say that it is a computer program or all those stuffs which you use daily on your smart phones or tablets.
Google Now, Cortana, Siri, Hound etc are some of the programmes that learns from you and tries to help you out with your daily life. The program learns everyday from us, you will notice it while using your smart devices if you pay attention.
In Future, A.I. might change the world dramatically. Actually A.I. reffers to all those things that are confusing. We use Artificial Intelligence all the time in our daily lives, but we often don't realize

5.2 Chat bot: The Chatterbot

A chatterbot (also known as a talkbot, chatbot, Bot, chatterbox, Artificial Conversational Entity) is a computer program which conducts a conversation via auditory or textual methods. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, thereby passing the Turing test. Chatterbots are typically used in dialog systems for various practical purposes including customer service or information acquisition. Some chatterbots use sophisticated natural language processing systems, but many simpler systems scan for keywords within the input, then pull a reply with the most matching keywords, or the most similar wording pattern, from a database.
The term "ChatterBot" was originally coined by Michael Mauldin (creator of the first Verbot, Julia) in 1994 to describe these conversational programs.

5.3 Chat bot in m-suvidha.

An intelligent chat bot will be used to give information or answers to any question asked by user related to bank. Our Intelligent system will first take input from bank customer. This input will be taken as text in written format. According to input, intelligent system will processes the query and give response to user. An artificial intelligence is most important and helpful part of our project. Intelligent system is automation of activities associated with human thinking, decision making, and problem solving process. This system will be available on android. Our system will represent the design and development of an intelligent chat bot. It will present a technology demonstrator to verify a proposed framework required to support such a bot (an android service). While a black box approach is used, by controlling the communication structure, to and from the android-service, the android-service allows all types of clients to communicate to the server from any platform. The service provided will be accessible through a generated interface which allows for seamless Java processing; whereby the extensibility improves the lifespan of such a service. By introducing an artificial brain, the android-based bot generates customized user responses, aligned to the desired character. Questions asked to the bot, which will not be understood, are further processed using a third-party manual system, and the response will be archived, improving the artificial brain capabilities for future generation of responses

5.4 Detailed Implementations and Technologies involved

Basically a chat bot is a computer program that when you provide it with some inputs in Natural Language (English, French ...) responds with something meaningful in that same language. Which means that the strength of a chat bot could be directly measured by the quality of the output selected by the Bot in response to the user. By the previous description, we could deduce that a very basic chat bot can be written in a few lines of code in a given specific programming language.

5.5 Code

```
import java.io.*;
import java.util.*;


    public class Chatterbot1 {


    static String[] Response = {
”I HEARD YOU!”,
”SO,YOU ARE TALKING TO ME.”,
”CONTINUE,I’M LISTENING.”,
”VERY INTERESTING CONVERSATION.”,
”TELL ME MORE...”
};

/**
* @param args
*/
public static void main(String[] args) throws Exception {
// TODO Auto-generated method stub
while(true) {
System.out.print(”¿”);
BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
String sInput = in.readLine();
Random generator = new Random();
int nSelection = generator.nextInt(Response.length);
String sResponse = Response[nSelection];
if(sInput.equalsIgnoreCase(”BYE”)) {
System.out.println(”IT WAS NICE TALKING TO YOU USER, SEE YOU NEXT
TIME!”);
break;
} else {
System.out.println(sResponse);
}
}
}
}
```

As you can see, it doesn’t take a lot of code to write a very basic program that can interact with a user but it would probably be very difficult to write a program that would really be capable of truly interpreting what the user is actually saying and after that would also generate an appropriate response to it. These have been a long term goal since the beginning and even before the very first computers were created. In 1951,the British mathematician Alan Turing has came up with the question Can machines think and he has also propose a test which is now known as the Turing Test.

In this test, a computer program and also a real person is set to speak to a third person (the judge) and he has to decide which of them is the real person. Nowadays, there is a competition that was named the Loebner Prize and in this competition bots that has successfully fool most of the judge for at list 5 minutes would win a prize of 100.000. So far no computer program was able to pass this test successfully. One of the major reasons for this is that computer programs written to compute in such contest have naturally the tendency of committing a lot of typo (they are often out of the context of the conversation). Which means that generally, it isn't that difficult for a judge to decide whether he is speaking to a "computer program" or a real person. Also, the direct ancestor of all those program that tries to mimic a conversation between real human beings is Eliza, the first version of this program was written in 1966 by Joseph Weizenbaum a professor of MIT.

Chatbots in general are considered to belong to the weak AI field (weak artificial intelligence) as opposed to strong a.i who's goal is to create programs that are as intelligent as humans or more intelligent. But it doesn't mean that chatbots do not have any true potential. Being able to create a program that could communicate the same way humans do would be a great advance for the AI field. Chatbot is this part of artificial intelligence which is more accessible to hobbyist (it only take some average programming skill to be a chatbot programmer). So, programmers out there who wanted to create true AI or some kind of artificial intelligence, writing intelligent chatbots is a great place to start!

5.6 Problems in the above version

Well, there is a lot of them. First of all, we can clearly see that the program isn't really trying to understand what the user is saying but instead he is just selecting a random response from his database each time the user type some sentence on the keyboard. And also, we could notice that the program repeat himself very often. One of the reason for this is because of the size of the database which is very small (5 sentences). The second thing that would explain the repetitions is that we haven't implemented any mechanism that would control this unwanted behavior. How do we move from a program that just select responses randomly to whatever input that the user might enter on the keyboard to a program that shows some more understanding of the inputs?

The answer to that question is quiet simple; we simply need to use keywords. A keyword is just a sentence (not necessarily a complete one) or even a word that the program might recognize from the user's input which then makes it possible for the program to react to it (ex: by printing a sentence on the screen). For the next program, we will write a knowledge base or database, it will be composed of keywords and some responses associated to each keyword.

so, now we know what to do to improve "our first chatterbot" and make it more intelligent. Let's proceed on writing "our second bot", we will call it chatterbot2.

code

Now, the program can understand some sentences like "what is your name", "are you intelligent" etc And also he can choose an appropriate response from his list of responses for this given sentence and just display it on the screen. Unlike the previous version of the program (chatterbot1) Chatterbot2 is capable of choosing a suitable response to the given user input without choosing random responses that doesn't take into account what actually the user trying to say.

We've also added a couple of new techniques to theses new program: when the program is unable to find a matching keyword the current user input, it simply answers by saying that it doesn't understand which is quiet human like.

5.7 Improvement in previous chatterbot to make it better

There are quiet a few things that we can improve, the first one is that since the chatterbot tends to be very repetitive, we might create a mechanism to control these repetitions. We could simply store the previous response of that Chatbot within a string sPrevResponse and make some checkings when selecting the next bot response to see if it's not equal to the previous response. If it is the case, we then select a new response from the available responses.

The other thing that we could improve would be the way that the chatbot handles the users inputs, currently if you enter an input that is in lower case the Chatbot would not understand anything about it even if there would be a match inside the bot's database for that input. Also if the input contains extra spaces or punctuation characters (!;,.) this also would prevent the Chatbot from understanding the input. That's the reason why we will try to introduce some new mechanism to preprocess the user's inputs before it can be search into the Chatbot database. We could have a function to put the users inputs in upper case since the keywords inside the database are in uppercase and another procedure to just remove all of the punctuations and extra spaces that could be found within users input.

5.8 Problems with current version

Clearly there are still many limitations with this version of the program. The most obvious one would be that the program use "exact sentence matching" to find a response to the user's input. This means that if you would go and ask him "what is your name again", the program will simply not understand what you are trying to say to him and this is because it was unable to find a match for this input. And this definitely would sound a little bit surprising considering the fact that the program can understand the sentence "what is your name".

5.9 Fuzzy String Search and Levenshtein distance

There are at list two ways to solve this problem, the most obvious one is to use a slightly more flexible way for matching keywords in the database against the user's input. All we have to do to make this possible is to simply aloud keywords to be found within the inputs so that we will no longer have the previous limitation. The other possibility is much more complex, it use's the concept of Fuzzy String Search. To apply this method, it could be useful at first to break the inputs and the current keyword in separate words, after that we could create two different vectors, the first one could be use to store the words for the input and the other one would store the words for the current keyword. Once we have done this we could use the Levenshtein distance for measuring the distance between the two word vectors. (Notice that in order for this method to be effective we would also need an extra keyword that would represent the subject of the current keyword).

So, there you have it, two different methods for improving the chatterbot. Actually we could combine both methods and just selecting which one to use on each situation.

Finally, there are still another problem that you may have noticed with the previous chatterbot, you could repeat the same sentence over and over and the program wouldn't have any reaction to this. We need also to correct this problem.

In particular, the function for searching for keywords inside the database is now a little bit more flexible

5.10 Improvement

Here are some ideas

- since the code for the chatterbots have started to grow, it would be a good thing to encapsulate the implementation of the next chatterbot by using a class.
- also the database is still much too small to be capable of handling a real conversation with users, so we will need to add some more entries in it.
- it may happen sometimes that the user will press the enter key without entering anything on the keyboard, we need to handle this situation as well.
- the user might also try to trick the chatterbot by repeating his previous sentence with some slight modification, we need to count this as a repetition from the user.
- and finally, pretty soon you will also notice that we might need a way for ranking keywords when we have multiple choices of keywords for a given input, we need a way for choosing the best one among them.

5.11 Functions of Chat bot

select_response(): this function selects a response from a list of responses, there is a new helper function that was added to the program shuffle, this new function shuffles a list of strings randomly after seed_random_generator() was called.

save_prev_input(): this function simply saves the current user input into a variable (m_sPrevInput) before getting some new inputs from the user.

void save_prev_response(): the function save_prev_response() saves the current response of the chatterbot before the bot have started to search responses for the current input, the current responsesis save in the varaible (m_sPrevResponse).

void save_prev_event(): this function simply saves the current event (m_sEvent) into the variable (m_sPrevEvent). An event can be when the program has detected a null input from the user also, when the user repeats himself or even when the chatterbot makes repetitions has well etc.

void set_event(std::string str): sets the current event (m_sEvent)

void save_input(): makes a backup of the current input (m_sIntput) into the variable m_sInputBackup.

void set_input(std::string str): sets the current input (m_sInput)

void restore_input(): restores the value of the current input (m_sInput) that has been saved previously into the variable m_sInputBackup.

void print_response(): prints the response that has been selected by the chat robot on the screen.

void preprocess_input(): this function does some preprocessing on the input like removing punctuations, redundant spaces charactes and also it converts the input to uppercase.

bool bot_repeat(): verifies if the chatterbot has started to repeat himself.

bool user_repeat(): Verifies if the user has repeated his self.

bool bot_understand(): Verifies that the bot understand the current user input (m_sInput).

bool null_input(): Verifies if the current user input (m_sInput) is null.

bool null_input_repetition(): Verifies if the user has repeated some null inputs.

bool user_want_to_quit(): Check to see if the user wants to quit the current session with the chatterbot.

bool same_event(): Verifies if the current event (m_sEvent) is the same as the previous one (m_sPrevEvent).

bool no_response(): Checks to see if the program has no response for the current input.

bool same_input(): Verifies if the current input (m_sInput) is the same as the previous one (m_sPrevInput).

bool similar_input(): Checks to see if the current and previous input are similar, two inputs are considered similar if one of them is the substring of the other one (e.g.: how are you and how are you doing would be considered similar because how are you is a substring of how are you doing.

void get_input(): Gets inputs from the user.

void respond(): handles all responses of the chat robot whether it is for events or simply the current user input. So, basically, these function controls the behaviour of the program.

find_match(): Finds responses for the current input.

void handle_repetition(): Handles repetitions made by the program.

handle_user_repetition(): Handles repetitions made by the user.

void handle_event(std::string str): This function handles events in general.

5.12 Keyword Matching

Now, in these section, we a re going to introduce the concept of 'keyword ranking' into the Chatterbot. Keyword ranking is a way for the program to select the best keywords in his database when there are more than one keyword that match the users inputs. Ex: if we have the current user input: What is your name again, by looking into his database, the Chatbot would have a list of two keywords that match this input: 'WHAT' and 'WHAT IS YOUR NAME'. Which one is the best? Well, the answer is quiet simple, it is obviously: 'What is your name' simply because it is the longest keyword.

5.13 Equivalent keywords

Within the Chatterbots the record for the database aloud us to use only one keyword for each set of responses but sometimes it could be Useful to have more than one keyword associated to each set of responses. Specially when these keywords have the same meaning. E.g.: What is your name and Can you please tell me your name have both had the same meaning? So there would be no need to use different records for these keywords instead we can just modify the record structure so that it aloud us to have more than one keyword per records.

5.14 Keyword transposition and template response

One of the well known mechanisms of chatterbots is the capacity to reformulate the user's input by doing some basic verb conjugation. Example, if the user enters: YOU ARE A MACHINE, the chatterbot might respond: So, you think that I'm a machine. How did we arrive at this transformation? We may have done it by using two steps:

We make sure that the chatterbot have a list of response templates that is linked to the corresponding keywords. Responses templates are a sort of skeleton to build new responses for the chatterbot. usually we used wildcards in the responses to indicate that it is a template. On the previous example, we have used the template: (so, you think that*) to construct our response. During the reassembly process, we simply replace the wildcard by some part of the original input. In that same example, we have used: You are a machine, which is actually the complete original input from the user. After replacing the wildcard by the user's input, we have the following sentence: So, you think that you are a machine but we can not use these sentence as it is, before that we need to make some pronoun reversal in it.

The usual transpositions that we use mostly are the replacement of pronoun of the first person to pronoun of the second person, e.g.: you -¿ me, I'm -¿ you are etc. In the previous example by replacing "YOU ARE" by "I'M" in the users input, After applying these changes, the original sentence becomes: I'm a machine. Now we can replace the wildcard from the template by these new sentence which give us our final response for the Chatbot: So, you think that I'm a machine. Notice that it's not a good thing to use transposition too much during a conversation, the mechanism would become too obvious and it could create some repetition.

5.15 Code

```
import java.io.;
import java.nio.Buffer;
import java.util.;
public class ChatBotv1 {
private static String sInput = new String(””);
private static String sResponse = new String(””);
private static String sPrevInput = new String(””);
private static String sPrevResponse = new String(””);
private static String sEvent = new String(””);
private static String sPrevEvent = new String(””);
private static String sInputBackup = new String(””);
private static String sSubject = new String(””);
private static String sKeyWord = new String(””);
private static boolean bQuitProgram = false;
final static int maxInput = 1;
final static int maxResp = 6;
final static String delim = ”?!.;,”;
static String KnowledgeBase[][][] = {
//INSERT THE KB HERE AS UPADATED
```

PROPERLY CATEGORIZE THE KB AS PER PROXIMITY OF QUESTIONS..
SPLIT A SINGLE STATEMENT AS BELOW
eg: {"HOW TO "
+ "[SPACE] ENTER A NUMBER "
+ "ON A NEW LINE."}
WE HAVE NOT ADDED KB DUE TO ITS EXTENSIVE NATURE

```
private static Vector¡String¿ respList = new Vector¡String¿(maxResp);
public static void get_input() throws Exception
{
System.out.print("¿");
// saves the previous input
save_prev_input();
BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
sInput = in.readLine();
preprocess_input();
}
public static void respond()
{
save_prev_response();
set_event("BOT UNDERSTAND**");
if(null_input())
{
handle_event("NULL INPUT**");
}
else if(null_input_repetition())
{
handle_event("NULL INPUT REPETITION**");
}
else if(user_repeat())
{
handle_user_repetition();
}
else
{
find_match();
}

    if(user_want_to_quit())
{
bQuitProgram = true;
}

if(!bot_understand())
{
handle_event("BOT DON'T UNDERSTAND**");
}
```

```
if(respList.size() ¿ 0)
{
select_response();

if(bot_repeat())
{
handle_repetition();
}
print_response();
}
}

    public static boolean quit() {
return bQuitProgram;
}

    // make a search for the user’s input
// inside the database of the program
public static void find_match()
{
respList.clear();
// introduce thse new ”string variable” to help
// support the implementation of keyword ranking
// during the matching process
String bestKeyWord = ””;
Vector¡Integer¿ index_vector = new Vector¡Integer¿(maxResp);

    for(int i = 0; i ¡ KnowledgeBase.length; ++i)
{
String[] keyWordList = KnowledgeBase[i][0];

    for(int j = 0; j ¡ keyWordList.length; ++j)
{
String keyWord = keyWordList[j];
// we inset a space character
// before and after the keyword to
// improve the matching process
keyWord = insert_space(keyWord);

// there has been some improvements made in
// here in order to make the matching process
// a littlebit more flexible
if( sInput.indexOf(keyWord) != -1 )
{
//’keyword ranking’ feature implemented in this section
if(keyWord.length() ¿ bestKeyWord.length())
{
bestKeyWord = keyWord;
index_vector.clear();
index_vector.add(i);
}
else if(keyWord.length() == bestKeyWord.length())
{
index_vector.add(i);
}
}
}
}
if(index_vector.size() ¿ 0)
{
sKeyWord = bestKeyWord;
Collections.shuffle(index_vector);
int respIndex = index_vector.elementAt(0);
int respSize = KnowledgeBase[respIndex][1].length;
for(int j = 0; j ¡ respSize; ++j)
{
respList.add(KnowledgeBase[respIndex][1][j]);
}
}
}

    void preprocess_response()
{
if(sResponse.indexOf(”*”) != -1)
{
// extracting from input
find_subject();
// conjugating subject
sSubject = transpose(sSubject);

sResponse = sResponse.replaceFirst(”*”, sSubject);
}
}

void find_subject()
{
sSubject = ””; // resets subject variable
StringBuffer buffer = new StringBuffer(sInput);
buffer.deleteCharAt(0);
sInput = buffer.toString();
int pos = sInput.indexOf(sKeyWord);
if(pos != -1)
{
```

```
sSubject = sInput.substring(pos + sKeyWord.length() - 1,sInput.length());
}
}

    // implementing the 'sentence transposition' feature
public static String transpose( String str )
{
boolean bTransposed = false;
for(int i = 0; i ¡ transposList.length; ++i)
{
String first = transposList[i][0];
insert_space(first);
String second = transposList[i][1];
insert_space(second);

String backup = str;
str = str.replaceFirst(first, second);
if(str != backup)
{
bTransposed = true;
}
}

if( !bTransposed )
{
for( int i = 0; i ¡ transposList.length; ++i )
{
String first = transposList[i][0];
insert_space(first);
String second = transposList[i][1];
insert_space(second);
str = str.replaceFirst(first, second);
}
}
return str;
}

public static void handle_repetition()
{
if(respList.size() ¿ 0)
{
respList.removeElementAt(0);
}
if(no_response())
{
save_input();
set_input(sEvent);
```

```
find_match();
restore_input();
}
select_response();
}

public static void handle_user_repetition()
{
if(same_input())
{
handle_event(”REPETITION T1**”);
}
else if(similar_input())
{
handle_event(”REPETITION T2**”);
}
}

    public static void handle_event(String str)
{
save_prev_event();
set_event(str);

save_input();
str = insert_space(str);

set_input(str);

if(!same_event())
{
find_match();
}

restore_input();
}

public static void signon()
{
handle_event(”SIGNON**”);
select_response();
print_response();
}

public static void select_response() {
Collections.shuffle(respList);
sResponse = respList.elementAt(0);
```

```
}

public static void save_prev_input() {
sPrevInput = sInput;
}

public static void save_prev_response() {
sPrevResponse = sResponse;
}

    public static void save_prev_event() {
sPrevEvent = sEvent;
}

public static void set_event(String str) {
sEvent = str;
}

public static void save_input() {
sInputBackup = sInput;
}

public static void set_input(String str) {
sInput = str;
}

public static void restore_input() {
sInput = sInputBackup;
}

public static void print_response() {
if(sResponse.length() ¿ 0) {
System.out.println(sResponse);
}
}

public static void preprocess_input() {
sInput = cleanString(sInput);
sInput = sInput.toUpperCase();
sInput = insert_space(sInput);
}

public static boolean bot_repeat() {
return (sPrevResponse.length() ¿ 0 &&
sResponse == sPrevResponse);
}
```

```
public static boolean user_repeat() {
return (sPrevInput.length() ¿ 0 &&
((sInput == sPrevInput) ——
(sInput.indexOf(sPrevInput) != -1) ——
(sPrevInput.indexOf(sInput) != -1)));
}


public static boolean bot_understand() {
return respList.size() ¿ 0;
}

public static boolean null_input() {
return (sInput.length() == 0 && sPrevInput.length() != 0);
}

public static boolean null_input_repetition() {
return (sInput.length() == 0 && sPrevInput.length() == 0);
}

public static boolean user_want_to_quit() {
return sInput.indexOf(”BYE”) != -1;
}

public static boolean same_event() {
return (sEvent.length() ¿ 0 && sEvent == sPrevEvent);
}

public static boolean no_response() {
return respList.size() == 0;
}

public static boolean same_input() {
return (sInput.length() ¿ 0 && sInput == sPrevInput);
}

public static boolean similar_input() {
return (sInput.length() ¿ 0 &&
(sInput.indexOf(sPrevInput) != -1 ——
sPrevInput.indexOf(sInput) != -1));
}

static boolean isPunc(char ch) {
return delim.indexOf(ch) != -1;
}

// removes punctuation and redundant
// spaces from the user’s input
static String cleanString(String str) {
StringBuffer temp = new StringBuffer(str.length());
char prevChar = 0;
for(int i = 0; i ¡ str.length(); ++i) {
if((str.charAt(i) == ’ ’ && prevChar == ’ ’ ) —— !isPunc(str.charAt(i))) {
temp.append(str.charAt(i));
prevChar = str.charAt(i);
}
else if(prevChar != ’ ’ && isPunc(str.charAt(i)))

temp.append(’ ’); } } return temp.toString();
}

static String insert_space(String str)
{
StringBuffer temp = new StringBuffer(str);
temp.insert(0, ’ ’);
temp.insert(temp.length(), ’ ’);
return temp.toString();
}
public static void main(String[] args) throws Exception {
try {
signon();
while(!quit()) {
get_input();


    respond();
}
}
catch(Exception e) {
e.printStackTrace();
}
}
    }
```

Chapter 6

Implementation

6.1 m-Suvidha implementation highlights

m-Suvidha has its underlying base empowered by Android. The android framework supports Java implementations and various java libraries. Enhancing this power of android

- m-Suvidha has obtained its powerful security measures from the java cryptography library.
- m-Suvidha has obtained highly optimized performance from android java interpreter.
- The sandbox gives an individual life cycle and environment for m-suvidha.
- m-Suvidha has seamless flow of wire frames
- m-Suvidha has dynamic UI/UX design from android platform.
- m-Suvidha has a very flexible support to incorporate multi-lingual capability. It is currently supporting English,Hindi,Marathi and Gujarati.
- m-Suvidha has strongly exploited the API integration to integrate the ICICI API's to extract data and communicate with bank interfaces and gateways.
- m-Suvidha has captured the data base connectivity features and strong integration of php and sql databases. It communicates with entities hosted on cloud platform.
- m-Suvidha provides an adaptive app framework that allows you to provide unique resources for different device configurations.
- m-Suvidha queries the availability of device features at runtime if any app features require specific hardware such as a camera to capture account holders image.
- m-suvidha has enhanced and stimulated its own intelligence by blending network security intelligence algorithm to verify the signed documents and verify the Certifying authority dynamically.
- Like above m-Suvidha has also enhanced and stimulated its own intelligence which is backed up its own knowledge base and complex algorithm for Artificial intelligence in chatbot system for chat support.

6.2 Android basics

6.2.1 Introduction to Android

Android provides a rich application framework that allows you to build innovative apps and games for mobile devices in a Java language environment.

6.2.2 Apps provide multiple entry points

Android apps are built as a combination of distinct components that can be invoked individually. For instance, an individual activity provides a single screen for a user interface, and a service independently performs work in the background. From one component you can start another component using an intent. You can even start a component in a different app, such as an activity in a maps app to show an address. This model provides multiple entry points for a single app and allows any app to behave as a user's ”default” for an action that other apps may invoke.

6.2.3 Apps adapt to different devices

Android provides an adaptive app framework that allows you to provide unique resources for different device configurations. For example, you can create different XML layout files for different screen sizes and the system determines which layout to apply based on the current device's screen size. You can query the availability of device features at runtime if any app features require specific hardware such as a camera. If necessary, you can also declare features your app requires so app markets such as Google Play Store do not allow installation on devices that do not support that feature.

6.2.4 Application Fundamentals

Android apps are written in the Java programming language. The Android SDK tools compile your code—along with any data and resource files—into an APK: an Android package, which is an archive file with an .apk suffix. One APK file contains all the contents of an Android app and is the file that Android-powered devices use to install the app. Once installed on a device, each Android app lives in its own security sandbox:

- The Android operating system is a multi-user Linux system in which each app is a different user.

- By default, the system assigns each app a unique Linux user ID (the ID is used only by the system and is unknown to the app). The system sets permissions for all the files in an app so that only the user ID assigned to that app can access them.

- Each process has its own virtual machine (VM), so an app's code runs in isolation from other apps.

- By default, every app runs in its own Linux process. Android starts the process when any of the app's components need to be executed, then shuts down the process when it's no longer needed or when the system must recover memory for other apps.

In this way, the Android system implements the principle of least privilege. That is, each app, by default, has access only to the components that it requires to do its work and no more. This creates a very secure environment in which an app cannot access parts of the system for which it is not given permission.

6.2.5 App Components

App components are the essential building blocks of an Android app. Each component is a different point through which the system can enter your app. Not all components are actual entry points for the user and some depend on each other, but each one exists as its own entity and plays a specific role—each one is a unique building block that helps define your app's overall behavior. There are four different types of app components. Each type serves a distinct purpose and has a distinct lifecycle that defines how the component is created and destroyed. Here are the four types of app components:

6.2.5.1 Activities

An activity represents a single screen with a user interface. For example, an email app might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails. Although the activities work together to form a cohesive user experience in the email app, each one is independent of the others. As such, a different app can start any one of these activities (if the email app allows it). For example, a camera app can start the activity in the email app that composes new mail, in order for the user to share a picture. An activity is implemented as a subclass of Activity and you can learn more about it in the Activities developer guide.

6.2.5.2 Services

A service is a component that runs in the background to perform long-running operations or to perform work for remote processes. A service does not provide a user interface. For example, a service might play music in the background while the user is in a different app, or it might fetch data over the network without blocking user interaction with an activity. Another component, such as an activity, can start the service and let it run or bind to it in order to interact with it. A service is implemented as a subclass of Service and you can learn more about it in the Services developer guide.

6.2.5.3 Content providers

A content provider manages a shared set of app data. You can store the data in the file system, an SQLite database, on the web, or any other persistent storage location your app can access. Through the content provider, other apps can query or even modify the data (if the content provider allows it). For example, the Android system provides a content provider that manages the user's contact information. As such, any app with the proper permissions can query part of the content provider (such as ContactsContract.Data) to read and write information about a particular person.

Content providers are also useful for reading and writing data that is private to your app and not shared. For example, the Note Pad sample app uses a content provider to save notes. A content provider is implemented as a subclass of ContentProvider and must implement a standard set of APIs that enable other apps to perform transactions. For more information, see the Content Providers developer guide.

6.2.5.4 Broadcast receivers

A broadcast receiver is a component that responds to system-wide broadcast announcements. Many broadcasts originate from the system—for example, a broadcast announcing that the screen has turned off, the battery is low, or a picture was captured. Apps can also initiate broadcasts—for example, to let other apps know that some data has been downloaded to the device and is available for them to use. Although broadcast receivers don't display a user interface, they may create a status bar notification to alert the user when a broadcast event occurs. More commonly, though, a broadcast receiver is just a ”gateway” to other components and is intended to do a very minimal amount of work. For instance, it might initiate a service to perform some work based on the event. A broadcast receiver is implemented as a subclass of BroadcastReceiver and each broadcast is delivered as an Intent object. For more information, see the BroadcastReceiver class. A unique aspect of the Android system design is that any app can start another app's component

6.3 m-Suvidha implementation wire-flow

6.3.1 wire-flow sequencing

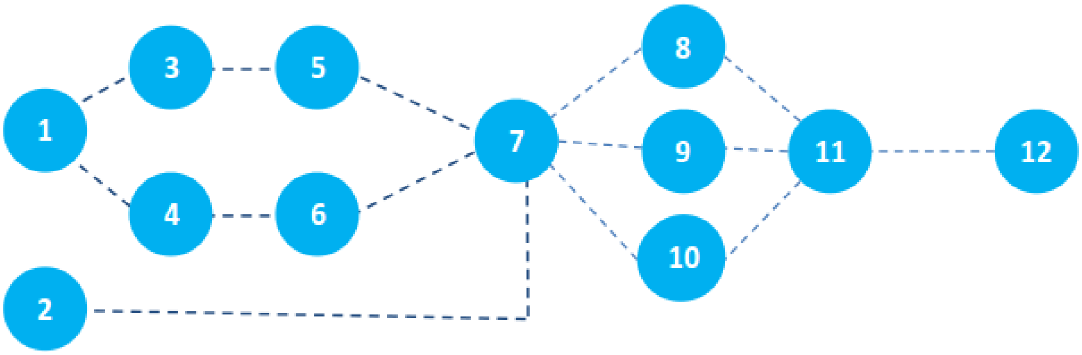


Figure 6.1: Wireflow

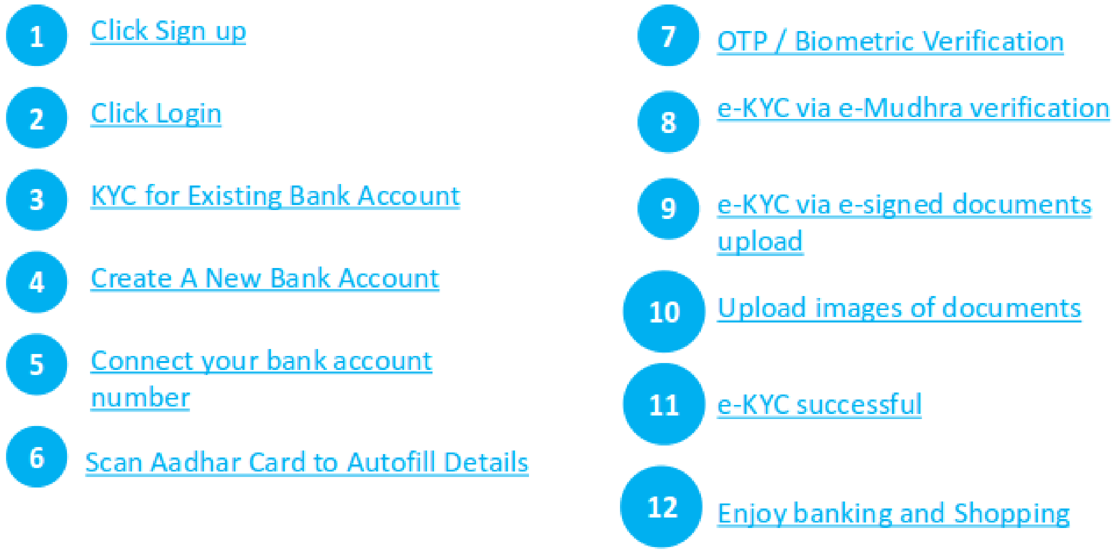

1 Click Sign up
2 Click Login
3 KYC for Existing Bank Account
4 Create A New Bank Account
5 Connect your bank account number
6 Scan Aadhar Card to Autofill Details
7 OTP / Biometric Verification
8 e-KYC via e-Mudhra verification
9 e-KYC via e-signed documents upload
10 Upload images of documents
11 e-KYC successful
12 Enjoy banking and Shopping

Figure 6.2: Wireflow Description

6.4 Sign-up

6.4.1 Sign-up details

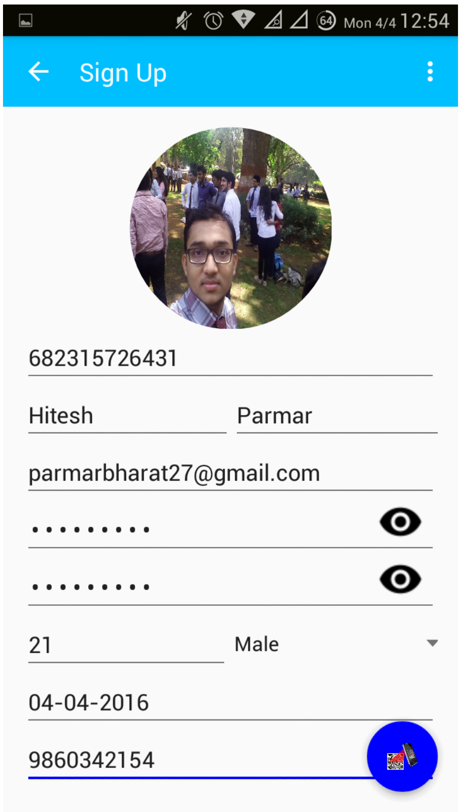
The sign-up is functionality we use to on-board the customer. o basically any user to the m-suvidha who opens the app and is not previously registered is lead to this frame of m-suvidha whereby the user can on-board himself.
There are two options given to the user in this window:

1. Already have a bank account.
2. Open a Bank Account.

The option 1 of Already have a bank account is for the user who is registered with bank and may have either single and multiple bank accounts. This option will enable her/him to register with the existing accounts as the app fetches the existing account data.

The option 2 of open a bank account is for the user who doesn't have an account with the bank or is not involved with the bank in any way.This option will enable the user to setup his new bank profile. Refer Figure 6.3

6.4.2 Screenshot of Sign-up



682315726431

Hitesh Parmar

parmarbharat27@gmail.com

•••••••••

•••••••••

21 Male

04-04-2016

9860342154

Figure 6.3: Sign-up screenshot

6.4.3 Code snippet of Sign-up

```
public class StartJunction extends AppCompatActivity {
Button newAccount ;
Button bankKYC;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_start_junction);
newAccount = (Button) findViewById(R.id.newAccount);
bankKYC = (Button) findViewById(R.id.bankKYC);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setDisplayShowHomeEnabled(true);

bankKYC.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Intent i = new Intent(StartJunction.this, HaveAccount.class);
startActivity(i);
}
});
newAccount.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Intent i = new Intent(StartJunction.this, MainActivity.class);
startActivity(i);
} });
}
```

6.5 Login

6.5.1 Login details

The Login functionality is to allow a registered m-Suvidha user to log into the m-Suvidha account to access the functionality of the app. The following is used to authenticate the user:

1. EmailId
2. Password
3. Aadhar Number

On entering the above details the user can request an OTP to authenticate herself/himself. The OTP is a unique functionality that is important to m-Suvidha as it is integrated with the UIDAI(Unique Identification Authority of India server. The UIDAI server generates an OTP and sends it as an SMS on the registered mobile number corresponding to the aadhar number entered above. As per the RBI guidelines for e-KYC, OTP is one of the two legitimate ways to authenticate a person.
The OTP received as an SMS is supposed to be entered in the OTP field which is then sent to the server whereby successfully authenticating the user. Refer Figure 6.4

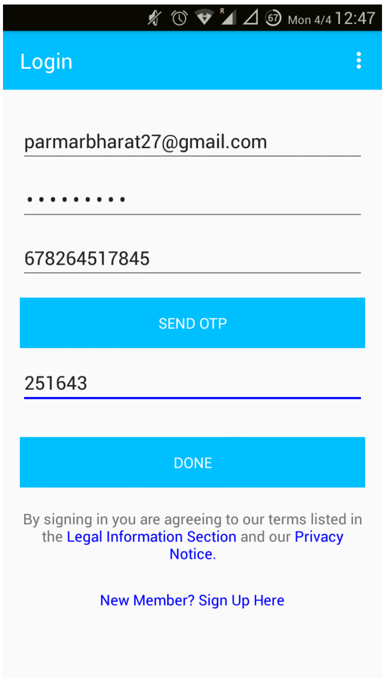
6.5.2 Screen-shot of Login

[Figure 6.4]

6.5.3 Code snippet of Login

```
public class LoginActivity extends AppCompatActivity {
Button sendOTP, done;
Context c;

protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_login_activity);
Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);
```


Mon 4/4 12:47

Login

parmarbharat27@gmail.com

•••••••••

678264517845

SEND OTP

251643

DONE

By signing in you are agreeing to our terms listed in the Legal Information Section and our Privacy Notice.

New Member? Sign Up Here

Figure 6.4: Login screenshot

```
// Progress dialog
pDialog = new ProgressDialog(this);
pDialog.setCancelable(false);

// SQLite database handler
db = new SQLiteHandler(getApplicationContext());

// Session manager
session = new SessionManager(getApplicationContext());

// Check if user is already logged in or not
if (session.isLoggedIn()) {
// User is already logged in. Take him to main activity
Intent intent = new Intent(LoginActivity.this, Navigation.class);
startActivity(intent);
finish();
}
// Login button Click Event
done.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) {
String email = inputEmail.getText().toString().trim();
String password = inputPassword.getText().toString().trim();
if (!email.isEmpty() && !password.isEmpty()) {
checkLogin(email, password);
} else {
```

```
Toast.makeText(getApplicationContext(),
”Please enter the credentials!”, Toast.LENGTH_LONG)
.show();
}
}
});
```

6.6 KYC for Existing Bank Account

6.6.1 KYC for Existing Bank Account

Many customers have an account with the bank but the KYC is not done for that account. Thus, m-Suvidha allows the users with an existing bank account to perform KYC without having to re-enter the details. It asks the user to enter the bank account number along with the aadhar id. The users can then perform the KYC after they login. Refer Figure 6.5

6.6.2 Screenshot of KYC for Existing Bank Account

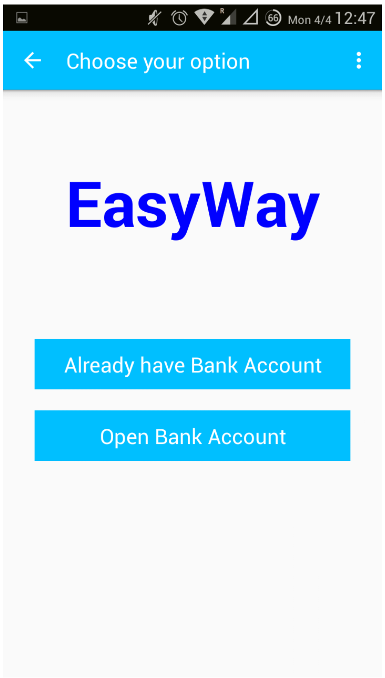


Figure 6.5: KYC for Existing Bank Account screenshot

6.6.3 Code snippet of KYC for Existing Bank Account

```
¡?xml version=”1.0” encoding=”utf-8”?¿
¡RelativeLayout xmlns:android=”http://schemas.android.com/apk/res/android”
....XML ENCODING
¿

¡ScrollView
android:layout_width=”match_parent”
android:layout_height=”match_parent”¿

¡LinearLayout
android:layout_width=”fill_parent”
android:layout_height=”fill_parent”
android:orientation=”vertical”¿

¡EditText
android:layout_width=”fill_parent”
android:layout_height=”wrap_content”
android:layout_marginTop=”10dp”
android:singleLine=”true”
android:inputType=”textEmailAddress”
android:id=”@+id/have_email_id”
android:hint=”@string/have_email_id”/¿
¡/LinearLayout¿

....AND OTHER SIMILAR XML

¡Button
android:layout_width=”fill_parent”
android:layout_height=”wrap_content”
android:id=”@+id/btn_proceed”
android:background=”@drawable/ripple”
android:layout_marginTop=”20dp”
android:textColor=”#FFFFFF”
android:text=”@string/proceed”/¿

¡/LinearLayout¿
¡/ScrollView¿
¡/RelativeLayout¿
```

6.7 Create A New Bank Account

6.7.1 Create A New Bank Account details

This is for the users who do not have a bank account. They are the new customers who need to register themselves using the following fields:

1.Aadhar Number
2.FirstName and LastName
3.EmailId
4.Password
5.Age
6.Gender
7.Date Of Birth
8.Mobile Number
9.Address

After the registration, the users can login to m-Suvidha using the registered details.

6.7.2 Code snippet of Create A New Bank Account

```
done.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
checkDone();
if (allgood) {
//for image
uploadImage();
//————image end
registerUser(uid1, name1, name2, email_id1, password1, age1, gender1, dob1, phonenum1, address1);
Intent i = new Intent(MainActivity.this, Navigation.class);

Notification myNotication;
NotificationManager manager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
;

Intent intent = new Intent(”com.rj.notitfications.SECACTIVITY”);

PendingIntent pendingIntent = PendingIntent.getActivity(MainActivity.this, 1, intent, 0);

Notification.Builder builder = new Notification.Builder(MainActivity.this);

builder.setAutoCancel(true);
builder.setTicker(”Account Creation Successful”);
builder.setContentTitle("EasyWay-Congratulations");
builder.setContentText("Your account is created successfully");
builder.setSmallIcon(android.R.drawable.stat_notify_more);
builder.setContentIntent(pendingIntent);
builder.setOngoing(true);
builder.setSubText("Click to continue"); //API level 16
builder.setNumber(1);
builder.build();

myNotication = builder.getNotification();
manager.notify(11, myNotication);

startActivity(i);
finish();
} else {
Toast.makeText(c, "Enter data in all fields", Toast.LENGTH_LONG).show();
}

}
});
```

6.8 Scan Aadhar Card to Autofill Details

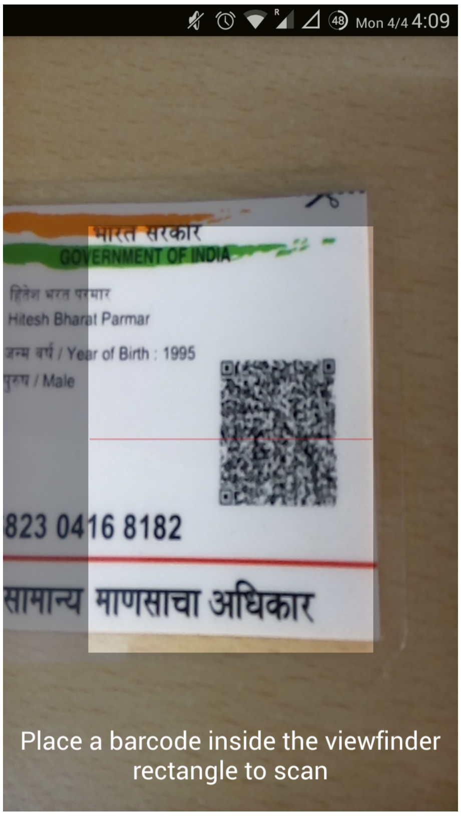
6.8.1 Scan Aadhar Card to Autofill Details

m-Suvidha has a QR code scanner. This saves the users of the effort to enter the details. The users can scan the QR code of the aadhar card that will fill in the following fields automatically:

1. Aadhar number
2. FirstName and LastName
3. Gender
4. Date Of Birth
5. Address

Refer Figure **??**

6.8.2 Screenshot of Scan Aadhar Card to Autofill Details



Place a barcode inside the viewfinder rectangle to scan

Figure 6.6: Scan Aadhar Card to Autofill screenshot

6.8.3 Code snippet of Scan Aadhar Card to Autofill Details

```
if (requestCode == RESULT_SCAN_QR)
if (resultCode == RESULT_OK)
String contents = data.getStringExtra(”SCAN_RESULT”);
try

card = new AadhaarXMLParser().parse(contents);
Toast.makeText(this, ”Card Parsed ”, Toast.LENGTH_SHORT).show();
firstname.setText(card.name.split(” ”)[0]);
lastname.setText(card.name.substring(card.name.lastIndexOf(” ”)));
uid.setText(card.uid);
address.setText(card.house + ” ” + card.dist + ”, ” + card.state + ”, ” + card.pincode);
//for setting age
Date d = new Date();
String s = d.toString();
String[] a = s.split(” ”);
int approxage = Integer.parseInt(a[5]) - Integer.parseInt(card.yob);
age.setText(String.valueOf(approxage));
catch (Exception e)
e.printStackTrace();
else if (resultCode == RESULT_CANCELED)
```

6.9 OTP / Biometric Verification

6.9.1 OTP / Biometric Verification details

According to the RBI guidelines, it is advised that while using e-KYC service of UIDAI, the individual user has to authorize the UIDAI, by explicit consent, to release her or his identity/address through biometric authentication and/or One Time Pin (OTP) verification to the bank branches/business correspondents (BCs). The UIDAI then transfers the data of the individual comprising name, age, gender, and photograph of the individual, electronically to the bank/BCs, which may be accepted as valid process for KYC verification.
m-Suvidha provides a One Time Pin provision with the mobile number of the user that is registered with UIDAI. The user can request an OTP on the registered mobile number and verify his/her identity with the UIDAI.

6.9.2 Code snippet of OTP / Biometric Verification

```
public static class RequestOTPTask extends AsyncTask¡String, Void, String[]¿ {

@Override
protected String[] doInBackground(String... params) {
Log.d(”Request OTP Async”, ”YEahh”);
//get the various required fields
String son1 = params[1];
Boolean otp_sent = false;
DefaultHttpClient client = new DefaultHttpClient();
HostnameVerifier hostnameVerifier = org.apache.http.conn.ssl.SSLSocketFactory.
ALLOW_ALL_HOSTNAME_VERIFIER;
String result[] = new String[1];
//adding ssl capabilities
SchemeRegistry registry = new SchemeRegistry();
SSLSocketFactory socketFactory = SSLSocketFactory.getSocketFactory();
socketFactory.setHostnameVerifier((X509HostnameVerifier) hostnameVerifier);
registry.register(new Scheme(”https”, socketFactory, 443));
SingleClientConnManager mgr = new SingleClientConnManager(client.getParams(),
registry);
DefaultHttpClient httpClient = new DefaultHttpClient(mgr, client.getParams());
HttpConnectionParams.setConnectionTimeout(httpClient.getParams(), 10000); //Time-
out Limi
HttpResponse response;
try {
HttpPost post = new HttpPost(”https://ac.khoslalabs.com/hackgate/hackathon/otp”);
StringEntity se = new StringEntity(son1);
se.setContentType(new BasicHeader(HTTP.CONTENT_TYPE, ”application/json”));
post.setEntity(se);
response = httpClient.execute(post);
/*Checking response */
if (response != null) {
Log.d(”RESPONSE”, ”FOUND”);
ResponseHandler¡String¿ handler = new BasicResponseHandler();
String body = handler.handleResponse(response);
Log.d(”stream”, body);
if (body.indexOf(”success:̈true”) ¿ 0) {
otp_sent = true;
otpResponseState = true;
result[0] = ”OTP Sent!”;
}
if (body.indexOf(”success:̈false”) ¿ 0) {
result[0] = ”OTP not received. Temp OTP used. Scan Aadhaar !”;
SmsManager smsManager = SmsManager.getDefault();
TelephonyManager tMgr = (TelephonyManager)
c.getSystemService(Context.TELEPHONY_SERVICE);
//String mPhoneNumber = tMgr.getLine1Number(); - for getting sim number - not good to use
smsManager.sendTextMessage(”7387144678”, null, ”Your OTP is 123456, This password is valid only for 300 seconds from now.”, null, null);
}
}
} catch (Exception e) {
e.printStackTrace();
}
if (otp_sent == true) {
Log.d(”AsyncTask”, ”OTP succesfully sent”);
return result;
}
result[0] = ”Check Connectivity.”;
return result;
}
}
```

6.10 e-KYC via e-Mudhra verification

6.10.1 e-KYC via e-Mudhra verification details

eMudhra eKYC services is a quick and convenient mechanism for organizations to get the customers' KYC details verified electronically based on customers' Aadhaar details. Using this KYC API, agencies can conduct electronic identity verification using OTP/biometrics and obtain a digitally signed (by UIDAI) electronic identity document. This makes the entire process extremely simple for customers and agencies and cost effective.

eMudhra eKYC Services highlights:

1. eMudhra is a licensed KYC User Agency (KUA) of UIDAI

2. Digital authentication of customers KYC can be done by submission of their OTP/biometric information

3. eMudhra KUA can fetch customers KYC details electronically from aadhar repository of UIDAI

4. By this process customers authenticity can be electronically verified within few seconds

e-KYC Process:

1. Customer comes to any organization for eKYC service.

2. Organisation initiates the KYC verification process through its application using eKYC service.

3. Customer's Aadhaar data is sent to Aadhaar repository through eKYC Service provider (eMudhra).

4. Aadhaar repository will return the KYC information to ASP through eMudhra.

Using the e-Sign API provided by e-Mudhra, the Application Service Provider (ASP) sends the request xml for electronic signature with the inputs Aadhaar Number, Authentication parameter (OTP/Biometric) and Document Hash and obtains the response xml from eMudhra eSign Service which has Consolidated PKCS #7 Response, User x.509 certificate and Signature Data.
In this case, m-Suvidha is an Application Service Provider(ASP). However, since m-Suvidha is not an authorized entity, it cannot successfully integrate the API.

6.11 e-KYC via e-signed documents upload

6.11.1 e-KYC via e-signed documents upload details

Another way to perform e-KYC is to let the user upload e-signed documents. The user can upload the e-signed documents on m-Suvidha and have their e-KYC done instantly. There are agencies like DigiLocker that enable the user to obtain electronically signed documents. These e-signed documents can be uploaded and the address of the user can be verified.

6.11.2 Code snippet of e-KYC via e-signed documents upload

```
private void selectImage1() {
final CharSequence[] items = {"Take Photo", "Choose from Library",
"Cancel"};

AlertDialog.Builder builder = new AlertDialog.Builder(DigitalCertificates.this);
builder.setTitle("Add Photo!");
builder.setItems(items, new DialogInterface.OnClickListener() {
```

```
@Override
public void onClick(DialogInterface dialog, int item) {
if (items[item].equals("Take Photo")) {
Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE1);
}

} else if (items[item].equals("Choose from Library")) {
Intent intent = new Intent();
intent.setType("application/pdf");
intent.setAction(Intent.ACTION_GET_CONTENT);
startActivityForResult(Intent.createChooser(intent, "Select Picture"),
PICK_IMAGE_REQUEST1);

} else if (items[item].equals("Cancel")) {
dialog.dismiss();
}
}
});
builder.show();
}
```

6.12 Upload images of documents

6.12.1 Upload images of documents details

The following type documents are needed to perform KYC:

1. Identity proof
    - Photo PAN Card
    - In case of Non Photo PAN Card in addition to copy of PAN Card any one of the following : Driving License /Passport copy / Voter ID /Bank Photo Pass Book.
2. Proof of Address (any one of the following):
    - Latest Telephone Bill: Landline/Mobile (not more than 3 months prior to the date of application).
    - Latest Electricity Bill (not more than 3 months prior to the date of application).
    - Passport copy
    - Latest Bank Passbook/Bank Account Statement (not more than 3 months prior to the date of application) .
    - Latest Demat Account statement (not more than 3 months prior to the date of application).

- Voter ID.
- Driving License.
- Ration Card.
- Rent Agreement.

3. Recent photograph

m-Suvidha allows the users to upload the scanned images of the documents easily. The Aadhar number is used as a proof of identity accompanied with the OTP verification. The users are given an assurance that their personal details will not be compromised. The uploaded documents are stored securely on the server and help the application perform the electronic KYC. Refer Figure 6.7

6.12.2 Screenshot of Upload images of documents

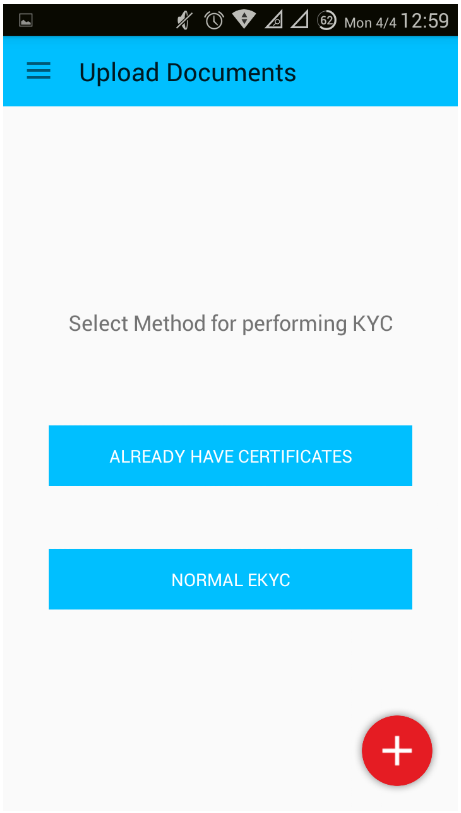


Figure 6.7: Upload images of documents screenshot

6.12.3 Code snippet of Upload images of documents

```
public class UploadImages extends AppCompatActivity {
RadioButton pan, aadhar, other_id, other_address;
RadioGroup radioGroup;
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.upload_images);
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setDisplayShowHomeEnabled(true);

View.OnClickListener pan_radio_listener = new View.OnClickListener(){
public void onClick(View v) {
//Your Implementations...
Intent intent = new Intent(UploadImages.this, PanCard.class);
startActivity(intent);
}
};
View.OnClickListener aadhar_radio_listener = new View.OnClickListener(){
public void onClick(View v) {
//Your Implementations...
Intent intent = new Intent(UploadImages.this, AadharCard.class);
startActivity(intent);
}
};
View.OnClickListener other_id_radio_listener = new View.OnClickListener()
public void onClick(View v) {
//Your Implementations...
Intent intent = new Intent(UploadImages.this, OtherId.class);
startActivity(intent);
}
};
View.OnClickListener other_address_radio_listener = new View.OnClickListener()
public void onClick(View v) {
//Your Implementations...
Intent intent = new Intent(UploadImages.this, OtherAddress.class);
startActivity(intent);
}
};
pan.setOnClickListener(pan_radio_listener);
aadhar.setOnClickListener(aadhar_radio_listener);
other_id.setOnClickListener(other_id_radio_listener);
other_address.setOnClickListener(other_address_radio_listener);
radioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener()
{
@Override
public void onCheckedChanged(RadioGroup group, int checkedId) {
}
});
}
public boolean onOptionsItemSelected(MenuItem item){
switch (item.getItemId()) {
```

```
case android.R.id.home:
this.finish();
return true;
}
return super.onOptionsItemSelected(item);
}
}
```

6.13 e-KYC successful

6.13.1 e-KYC successful details

After the user successfully uploads the documents, the user will be notified that the e-KYC is successful. The user can then use the Savings account into the wallet for banking and Shopping. The newly created savings bank account is KYC enabled. Thus, the user is smoothly on-boarded and can enjoy additional benefits. The user will be assigned a unique bank account number that will be notified to the user within the application.

6.14 Enjoy banking and Shopping

6.14.1 Enjoy banking and Shopping details

m-Suvidha allows the users to enjoy banking and shopping through the KYC enabled wallet. The users enjoy heightened benefits such as increased transaction limit from Rs. 10K to Rs. 100K. m-suvidha provides a really good on-boarding experience and ensures that the users don't have to spend time and efforts as it would be in the past. the users can enjoy shopping through their wallet and also perform cash transfer transactions through m-Suvidha.

6.15 Multi-Lingual support

6.15.1 Multi-Lingual support details

Inclusion is very important when it comes to Banking services. Inclusion of people of different cultures, different geographies, speaking different languages is very critical to the success of any banking service. m-Suvidha provides a Multi-lingual support that aids cultural inclusion. It aims at making the on-boarding experience better for the users by providing the service in their native languages.

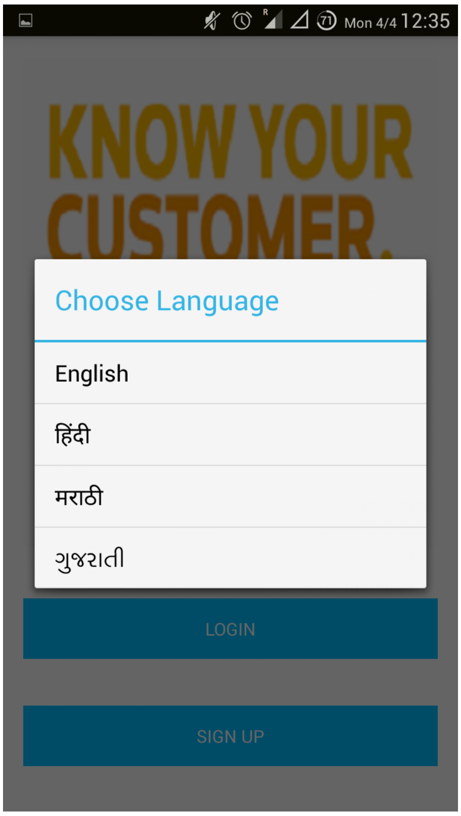
It supports the following languages:

1. English
2. Hindi
3. Marathi

4. Gujarati

Refer Figure 6.8

6.15.2 Screenshot of Multi-Lingual support



Choose Language

English

हिंदी

मराठी

ગુજરાતી

Figure 6.8: Multi-Lingual support sreenshot

6.15.3 Code snippet of Multi-Lingual support

```
final CharSequence[] items = {getString(R.string.english),getString(R.string.hindi
),getString(R.string.marathi),getString(R.string.gujrati)};

builder.setTitle(”Choose Language”);
builder.setItems(items, new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int item) {
Toast.makeText(getApplicationContext(), items[item], Toast.LENGTH_SHORT).show();
switch(item){
case 0:setLocale(”en”);
break;
case 1:setLocale(”hi”);
break;
case 2:setLocale(”mr”);
break;
case 3:setLocale(”gu”);
break;
default:setLocale(”en”);
break;}
}
});
AlertDialog alert = builder.create();
alert.show();
```

6.16 Guidelines

6.16.1 Guidelines details

Guidelines provide a list of Frequently Asked Questions (FAQs) along with their answers to help the user navigate easily through the application. It aims at solving the queries of the users with lucid explanations.

The following are a few Frequently Asked Questions(FAQs) listed in m-Suvidha:

1. What is KYC and why is it required?
2. What are the KYC requirements for opening a bank account?
3. What are the documents to be given as ’Proof Of Identity’ and ’Proof Of Address’?

Refer Figure **??**

6.16.2 Screenshot of Guidelines

6.16.3 Code snippet of Guidelines

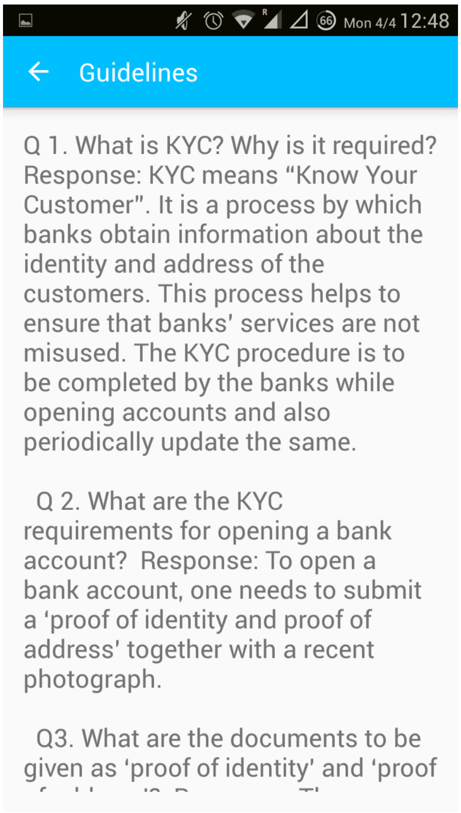
```
public class Guidelines extends AppCompatActivity {

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity
_guidelines);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setDisplayShowHomeEnabled(true);


public boolean onOptionsItemSelected(MenuItem item){

switch (item.getItemId()) {
case android.R.id.home:
```



Q 1. What is KYC? Why is it required? Response: KYC means "Know Your Customer". It is a process by which banks obtain information about the identity and address of the customers. This process helps to ensure that banks' services are not misused. The KYC procedure is to be completed by the banks while opening accounts and also periodically update the same.

Q 2. What are the KYC requirements for opening a bank account? Response: To open a bank account, one needs to submit a 'proof of identity and proof of address' together with a recent photograph.

Q3. What are the documents to be given as 'proof of identity' and 'proof

Figure 6.9: Guidelines sreenshot

```
this.finish();
return true;

return super.onOptionsItemSelected(item); }

}
```

6.17 Chatbot

6.17.1 Chatbot details

m-Suvidha has a live chatbot that will help the users with their queries. The users can initiate a conversation with the bot at anytime. The bot is sufficiently trained to answer the related queries and is efficient. It is one of the unique features of m-Suvidha.

6.18 Share application facility

6.18.1 Share application facility details

We live in the Social Networking era. Sharing digital content has become an integral part of our lives. m-Suvidha provides a facility to the users to share the application with their friends and networks easily. This allows the users to exchange the .apk file easily. It allows an exchange through a variety of media. Refer figure 6.10

A few are listed below:

1. Bluetooth
2. Mail
3. SMS
4. Facebook
5. Messenger
6. Whatsapp
7. Software Data cable
8. PushBullet

6.18.2 Screenshot of share application facility

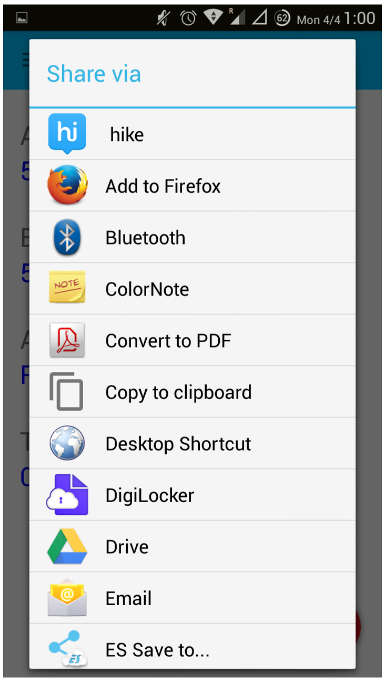


Figure 6.10: share application facility screenshot

6.18.3 Code snippet of share application facility

```
private void shareIt()
Intent sharingIntent = new Intent(android.content.Intent.ACTION_SEND);
sharingIntent.setType(”text/plain”);
String shareBody = ”Here is our APP link,www.EasyWayFinTech.com Download and
```

```
get OnBoarded”;
sharingIntent.putExtra(android.content.Intent.EXTRA_SUBJECT, ”Play Store Link -
EASY WAY”);
sharingIntent.putExtra(android.content.Intent.EXTRA_TEXT, shareBody);
startActivity(Intent.createChooser(sharingIntent, ”Share via”));
```

6.19 Transaction History - Mini Statement

6.19.1 Transaction History - Mini Statement Details

Transaction history implies a record of all the transactions done by the users in the past. m-Suvidha allows the users to see their transaction history over a period of time. They receive a mini-statement of their transactions during the stipulated time. This has been implemented by using the Application Programming Interface(API) by ICICI Bank.It specifies the details along the following fields:

1. Account Number
2. Transaction Amount
3. Transaction Type
4. Closing Balance
5. Transaction Date

6.20 Balance inquiry

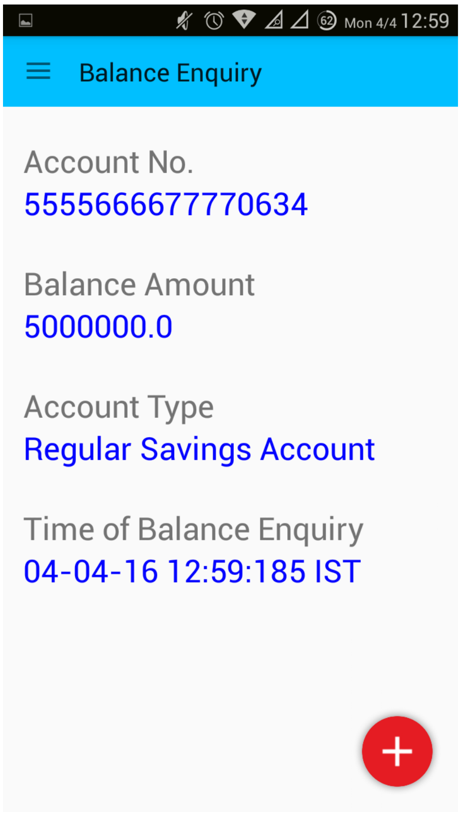
6.20.1 Balance inquiry details

m-Suvidha allows the user to check the balance at any point in time. The user will be provided with the current balance in the account. This helps the user to keep a check on the transactions and track the balance successfully.

6.20.2 Screenshot of Balance inquiry

```
class RetrieveFeedTask extends AsyncTask¡Void, Void, String¿ {
private Exception exception;
protected void onPreExecute() {
}
protected String doInBackground(Void... urls) {
// String email = emailText.getText().toString();
// Do some validation here

try {
//URL url = new URL(API_URL + ”email=” + email + ”&apiKey=” + API_KEY);
URL url = new URL(”http://retailbanking.mybluemix.net/banking/icicibank/
```



Account No.
5555666677770634

Balance Amount
5000000.0

Account Type
Regular Savings Account

Time of Balance Enquiry
04-04-16 12:59:185 IST

Figure 6.11: Balance inquiry screenshot

```
balanceenquiry?client_id=parmarbharat27@gmail.com&token=10195d088a84&
accountno=5555666677770634”);
HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
try {
BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(urlConnection.getInputStream()));
StringBuilder stringBuilder = new StringBuilder();
String line;
while ((line = bufferedReader.readLine()) != null) {
stringBuilder.append(line).append(””);
}
bufferedReader.close();
return stringBuilder.toString();
}
finally{
urlConnection.disconnect();
}
}
catch(Exception e) {
Log.e(”ERROR”, e.getMessage(), e);
return null;
}
}
```

```
protected void onPostExecute(String response) {
if(response == null) {
response = "THERE WAS AN ERROR";
}
progressBar.setVisibility(View.GONE);
Log.i("INFO", response);
// tv.setText(response);
try {
//JSONObject object = (JSONObject) new JSONTokener(response).nextValue();
JSONArray jsonArr = new JSONArray(response);
JSONObject object = jsonArr.getJSONObject(1);
Double balance = object.getDouble("balance");
String accountno = object.getString("accountno");
String accounttype = object.getString("accounttype");
String balancetime = object.getString("balancetime");
//JSONArray photos = object.getJSONArray("photos");
account_no.setText(accountno);
balance_amount.setText(String.valueOf(balance));
account_type.setText(accounttype);
balance_time.setText(String.valueOf(balancetime));
} catch (JSONException e) {
e.printStackTrace();
}
}
```

6.21 Logout

6.21.1 Logout details

The login creates a session for the user. This session terminates with the logout. the user has to login again to start a new session. The user cannot avail the facilities after she/he has logged out of the system.

6.21.2 Screenshot of Logout

6.21.3 Code snippet of Logout

```
private void logoutUser() {
session.setLogin(false);
db.deleteUsers();
// Launching the login activity
Intent intent = new Intent(Navigation.this, LoginActivity.class);
startActivity(intent);
finish();
}
public void onBackPressed() {
new AlertDialog.Builder(this)
.setIcon(android.R.drawable.alert_dark_frame)
.setTitle(getString(R.string.closing_application))
.setMessage(getString(R.string.logout_and_exit))
.setPositiveButton(getString(R.string.yes), new DialogInterface.OnClickListener()
{
@Override
public void onClick(DialogInterface dialog, int which) {
session.setLogin(false);
db.deleteUsers();
finish();
}

})
.setNegativeButton(getString(R.string.no), null)
.show();
}
```

6.22 Captcha

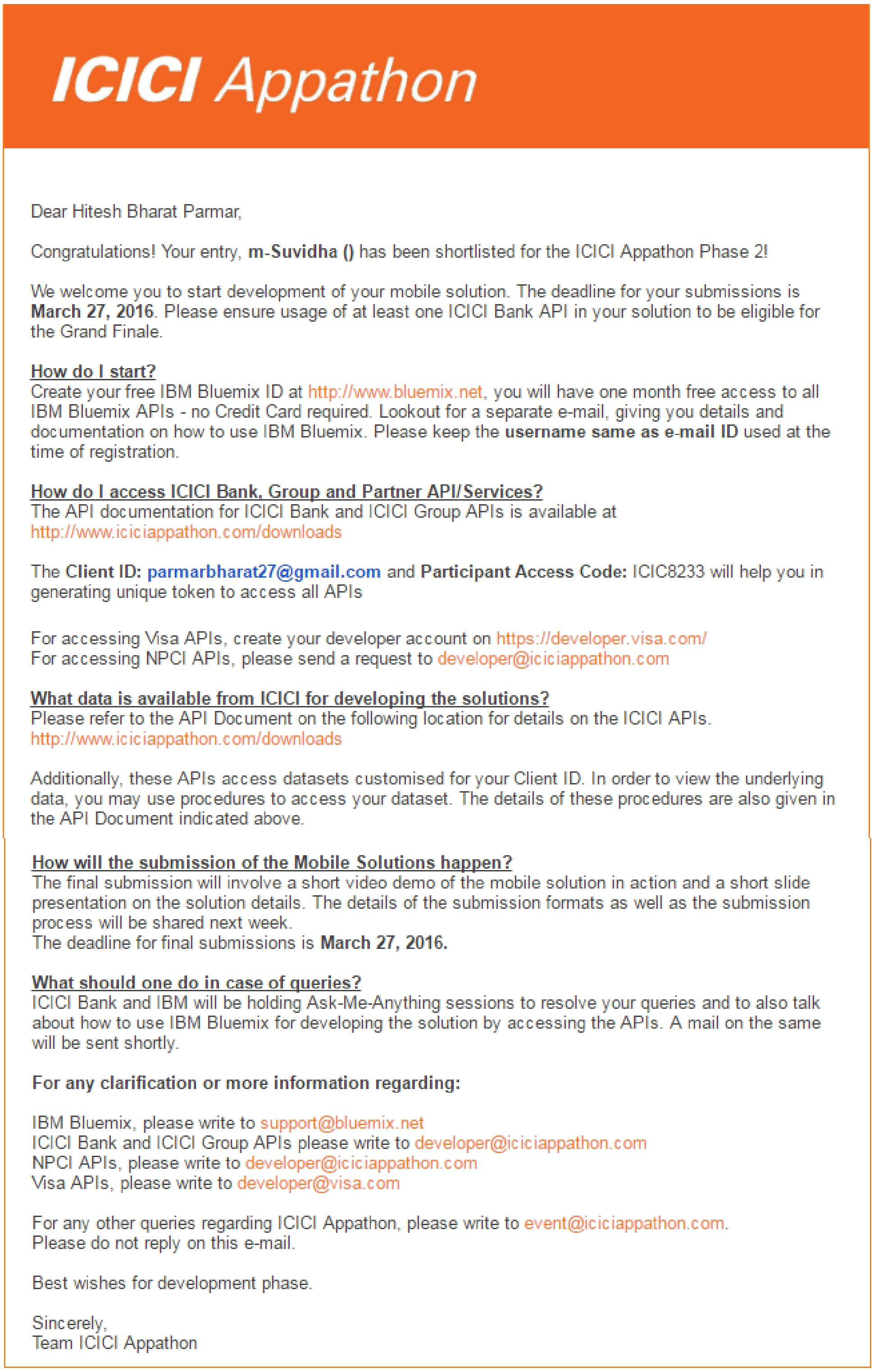
6.22.1 Captcha details

A CAPTCHA is a program that protects websites against bots by generating and grading tests that humans can pass but current computer programs cannot. For example, humans can read distorted text but current computer programs can't. m-Suvidha uses Captcha as another layer of security. It protects the application from intentional external attacks.

Chapter 7

Accomplishments

7.1 ICICI Appathon



ICICI Appathon

Dear Hitesh Bharat Parmar,

Congratulations! Your entry, **m-Suvidha ()** has been shortlisted for the ICICI Appathon Phase 2!

We welcome you to start development of your mobile solution. The deadline for your submissions is **March 27, 2016**. Please ensure usage of at least one ICICI Bank API in your solution to be eligible for the Grand Finale.

How do I start?
Create your free IBM Bluemix ID at http://www.bluemix.net, you will have one month free access to all IBM Bluemix APIs - no Credit Card required. Lookout for a separate e-mail, giving you details and documentation on how to use IBM Bluemix. Please keep the **username same as e-mail ID** used at the time of registration.

How do I access ICICI Bank, Group and Partner API/Services?
The API documentation for ICICI Bank and ICICI Group APIs is available at
http://www.iciciappathon.com/downloads

The **Client ID: parmarbharat27@gmail.com** and **Participant Access Code:** ICIC8233 will help you in generating unique token to access all APIs

For accessing Visa APIs, create your developer account on https://developer.visa.com/
For accessing NPCI APIs, please send a request to developer@iciciappathon.com

What data is available from ICICI for developing the solutions?
Please refer to the API Document on the following location for details on the ICICI APIs.
http://www.iciciappathon.com/downloads

Additionally, these APIs access datasets customised for your Client ID. In order to view the underlying data, you may use procedures to access your dataset. The details of these procedures are also given in the API Document indicated above.

How will the submission of the Mobile Solutions happen?
The final submission will involve a short video demo of the mobile solution in action and a short slide presentation on the solution details. The details of the submission formats as well as the submission process will be shared next week.
The deadline for final submissions is **March 27, 2016.**

What should one do in case of queries?
ICICI Bank and IBM will be holding Ask-Me-Anything sessions to resolve your queries and to also talk about how to use IBM Bluemix for developing the solution by accessing the APIs. A mail on the same will be sent shortly.

For any clarification or more information regarding:

IBM Bluemix, please write to support@bluemix.net
ICICI Bank and ICICI Group APIs please write to developer@iciciappathon.com
NPCI APIs, please write to developer@iciciappathon.com
Visa APIs, please write to developer@visa.com

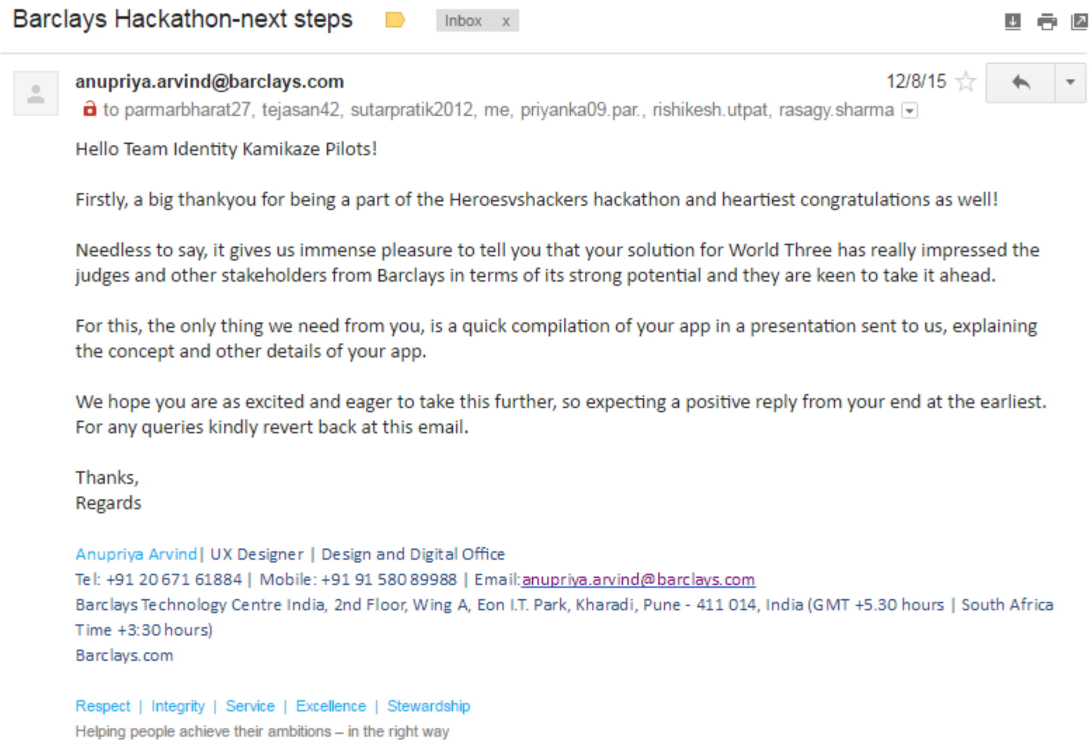
For any other queries regarding ICICI Appathon, please write to event@iciciappathon.com.
Please do not reply on this e-mail.

Best wishes for development phase.

Sincerely,
Team ICICI Appathon

Figure 7.1: ICICI Appathon

7.2 Hackathon and third party start-up at Barclays



anupriya.arvind@barclays.com 12/8/15

to parmarbharat27, tejasan42, sutarpratik2012, me, priyanka09.par., rishikesh.utpat, rasagy.sharma

Hello Team Identity Kamikaze Pilots!

Firstly, a big thankyou for being a part of the Heroesvshackers hackathon and heartiest congratulations as well!

Needless to say, it gives us immense pleasure to tell you that your solution for World Three has really impressed the judges and other stakeholders from Barclays in terms of its strong potential and they are keen to take it ahead.

For this, the only thing we need from you, is a quick compilation of your app in a presentation sent to us, explaining the concept and other details of your app.

We hope you are as excited and eager to take this further, so expecting a positive reply from your end at the earliest. For any queries kindly revert back at this email.

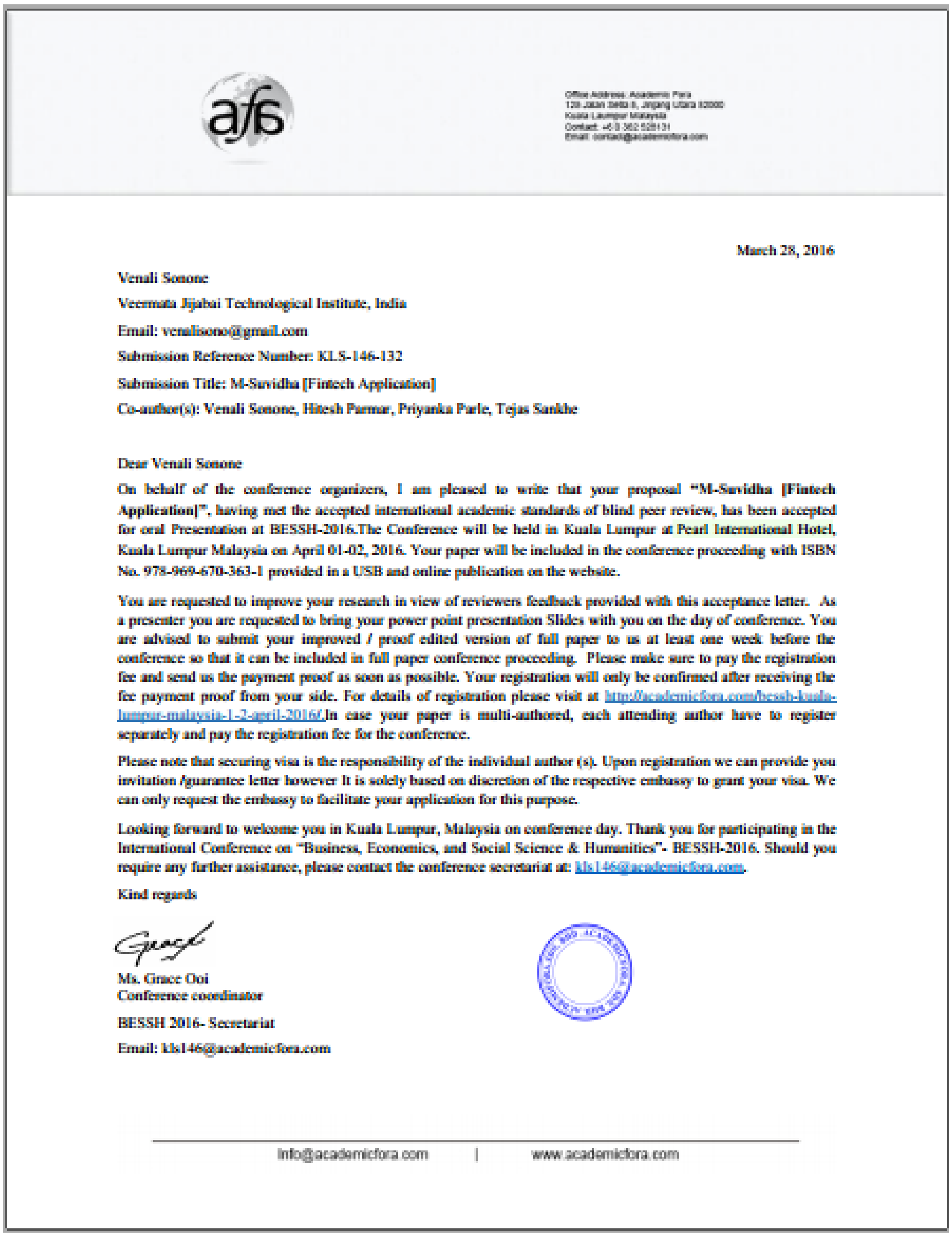
Thanks,
Regards

Anupriya Arvind| UX Designer | Design and Digital Office
Tel: +91 20 671 61884 | Mobile: +91 91 580 89988 | Email:anupriya.arvind@barclays.com
Barclays Technology Centre India, 2nd Floor, Wing A, Eon I.T. Park, Kharadi, Pune - 411 014, India (GMT +5.30 hours | South Africa Time +3:30 hours)
Barclays.com

Respect | Integrity | Service | Excellence | Stewardship
Helping people achieve their ambitions – in the right way

Figure 7.2: Hackathon

7.3 AFS paper presentation at BESSH 2016, Kuala Lumpur



afs

Office Address: Academic Fora
129 Jalan Setia 5, Jinjang Utara 52000
Kuala Lumpur Malaysia
Contact: +6 3 362 528131
Email: contact@academicfora.com

March 28, 2016

Venali Sonone

Veermata Jijabai Technological Institute, India

Email: venalisono@gmail.com

Submission Reference Number: KLS-146-132

Submission Title: M-Suvidha [Fintech Application]

Co-author(s): Venali Sonone, Hitesh Parmar, Priyanka Parle, Tejas Sankhe

Dear Venali Sonone

On behalf of the conference organizers, I am pleased to write that your proposal "M-Suvidha [Fintech Application]", having met the accepted international academic standards of blind peer review, has been accepted for oral Presentation at BESSH-2016.The Conference will be held in Kuala Lumpur at Pearl International Hotel, Kuala Lumpur Malaysia on April 01-02, 2016. Your paper will be included in the conference proceeding with ISBN No. 978-969-670-363-1 provided in a USB and online publication on the website.

You are requested to improve your research in view of reviewers feedback provided with this acceptance letter. As a presenter you are requested to bring your power point presentation Slides with you on the day of conference. You are advised to submit your improved / proof edited version of full paper to us at least one week before the conference so that it can be included in full paper conference proceeding. Please make sure to pay the registration fee and send us the payment proof as soon as possible. Your registration will only be confirmed after receiving the fee payment proof from your side. For details of registration please visit at http://academicfora.com/bessh-kuala-lumpur-malaysia-1-2-april-2016/.In case your paper is multi-authored, each attending author have to register separately and pay the registration fee for the conference.

Please note that securing visa is the responsibility of the individual author (s). Upon registration we can provide you invitation /guarantee letter however It is solely based on discretion of the respective embassy to grant your visa. We can only request the embassy to facilitate your application for this purpose.

Looking forward to welcome you in Kuala Lumpur, Malaysia on conference day. Thank you for participating in the International Conference on "Business, Economics, and Social Science & Humanities"- BESSH-2016. Should you require any further assistance, please contact the conference secretariat at: kls146@academicfora.com.

Kind regards

Grace

Ms. Grace Ooi
Conference coordinator

BESSH 2016- Secretariat

Email: kls146@academicfora.com

info@academicfora.com | www.academicfora.com

Figure 7.3: AFS

Chapter 8

Conclusion

8.1 Conclusion

m-Suvidha ensures smooth on-boarding of users to create KYC fulfilled bank accounts through an android application. m-Suvidha also allows the customer to enjoy the benefits of banking and online shopping through an integrated wallet. Thus, it eliminates the tedious paperwork and efforts required for opening a bank account and getting the KYC done. m-Suvidha also ensures unique identification of users through the authentication using Aadhar. This will aid in prevention of money laundering. m-Suvidha will revolutionize the way in which banking and e-KYC is done.

8.2 Future Scope

- Integrate the m-Suvidha with the banks and other business processes.
- Provide a Image Processing recognition for authentication apart from the e-Mudhra to trace the authenticity of document.
- e-Mudhra API acceptance for our application and its interfacing is a major future work we hope to be accomplished.
- Integrate merchants for wallet transactions.
- Integrate and interface all available biometric devices for verification.
- Make it adaptable to current regulatory rule [Basel 4]

Chapter 9

References

9.1 References

1. AL-Majali, M., & Mat, N. K. (2011), "Modeling the antecedents of internet banking service adoption (IBSA) in Jordan: A Structural Equation Modeling (SEM) approach", *Journal of Internet Banking and Commerce*, pp.1-16

2. Alpesh Patel, (2013), "M-Banking and M-Payments:The Next Frontier", Deloitte, Delhi

3. Bhatti T., (2007), "Exploring Factors Influencing the Adoption of Mobile Commerce", *Journal of Internet Banking and Commerce*, pp.1-13

4. Chaipoopirutana S., Combs H., Chatchawanwan Y., & Vij V., (2009), "Diffusion of innovation in Asia: A study of Internet banking in Thailand and India", *Innovative Marketing*, pp.27-31

5. Chugh V., (2014/02/17), Reserve bank of India, Retrieved from RBI Website: http://www.rbi.org.in/Scripts/bs_viewcontent.aspx?Id=1660

6. Kalakota R., & Robinson M., (2001), "M-Business: The Race to Mobility", *McGraw-Hill Companies*, New York

7. Kapania H., (2012-13), "COAI Annual Report 2012-13", *Cellular Operators Asociation of India*, Delhi

8. Kim G., Shin B., & Lee H. G., (2007), "Understanding dynamics between initial trust and usage intentions of mobile banking", *Information Systems Journal*, pp.283–311

9. Lin H.-F., (2010),"An empirical investigation of mobile banking adoption: The effect of innovation", *International Journal of Information Management*, pp.252-260

10. Mr. V. Vaidyanathan, (2008), "ICICI Bank launches iMobile: First bank in India to introduce complete", ICICI Bank, MUMBAI

11. Reserve Bank of India, (2014/02/17), Retrieved from RBI Website: http://www.rbi.org.in/scripts/bs_viewcontent.aspx?Id=2463

12. Rogers E. M., (2003), "Diffusion of Innovations", *Free Press*, New York

13. S.Samudra M., & Phadtare M., (2012), "Factors Influencing the Adoption of Mobile Banking with SpecialReference to Pune City", *ASCI Journal of Management*, pp.51-65

14. Sadi A., & Noordin, M. F., (2011)," Factors influencing the adoption of M-commerce: An exploratory Analysis", *International Conference on Industrial Engi-*

neering and Operations Management, pp.492-498, Malaysia

15. Safeena R., Date H., Kammani A., & Hundewale, N, (2012), Technology Adoption and Indian Consumers:Study on *International Journal of Computer Theory and Engineering*, pp.1020-1024

16. Sahin I., (2006), ”Detailed Review of Roger’s Diffusion of Innovations Theory and Educational Technology-Related Studies Based on Rogers”, *The Turkish Online Journal of Educational Technology*, pp.14-22

17. Singh S., Srivastava V., & Srivastav R., (2010), ”Customer Acceptance of Mobile Banking: A Conceptual Framework”, *SIES Journal of Management*, pp.55-64

18. TRAI, (2013), ”The Indian Telecom Services Performance Indicators”, *Telecom Regulatory Authority of India*, Delhi

19. Venkatesh V., Morris M. G., Davis G. B., & Davis F. D., (2003), ”User Acceptance of Information Technology:Toward a Unified View”, *MIS Quarterly*, pp.425-478

20. Vinayagamoorthy A., & Sankar C., (2012), ”Mobile Banking – An Overview”, *Advances In Management*, pp.24-29