**University of Mumbai**

**E-Banking Security**

Project – A Report

Submitted in partial fulfilment of requirements

for the degree of

Bachelor of Engineering

By

**Namrata Devadiga - 1311077**

**Hardik Jain - 1311087**

**Harshad Kothari - 1311094**

Under the guidance of

**Prof. Smita Sanke**



**Department of Computer Engineering**

**K. J. Somaiya College of Engineering, Mumbai-77**

(Autonomous College Affiliated to University of Mumbai)

**2016 - 2017**

Project A Report entitled E-Banking Security

Submitted by:

Namrata Devadiga - 1311077

Hardik Jain - 1311087

Harshad Kothari – 1311094

in partial fulfilment of the degree of B.E. in Computer Engineering is approved.

Internal Guide Examiners

Prof. Smita Sankhe --------------------

--------------------

-------------------------- ---------------------

Head of Department Principal

**TABLE OF CONTENTS**

1. Abstract………………………………………………………………………………………………………………………………………..4
2. Existing System…………………………………………………………………………………………………………………………….4
3. Problem Definition……………………………………………………………………………………………………………………….5
4. Scope……………………………………………………………………………………………………………………………………………5
5. Proposed System………………………………………………………………………………………………………………………….6
6. Programming Tools………………………………………………………………………………………………………………………8
7. Hardware and Software Requirements…………………………………………………………………………………….....8
   1. Hardware Requirements………………………………………………………………………………………………….8
   2. Software Requirements…………………………………………………………………………………………………..8
8. Software Project Management Plan…………………………………………………………………………………………….9
   1. Project Description………………………………………………………………………………………………………….9
   2. Project Deliverables………………………………………………………………………………………………………..9
   3. Software Process Model………………………………………………………………………………………………..10
   4. Roles and Responsibilities……………………………………………………………………………………………..11
   5. Tasks……………………………………………………………………………………………………………………………..11
   6. Timeline………………………………………………………………………………………………………………………..14
   7. Gantt chart…………………………………………………………………………………………………………………...16
9. Software Requirement Specification Document…………………………………………………………………………17
   1. Project Overview…………………………………………………………………………………………………………..17
   2. User Interfaces………………………………………………………………………………………………………………17
   3. Functional Requirements………………………………………………………………………………………………17
   4. Non Functional Requirements……………………………………………………………………………………...20
10. Software Design Document……………………………………………………………………………………………………….21
    1. Class Diagram……………………………………………………………………………………………………………....21
    2. Entity Relation Diagram………………………………………………………………………………………………..22
    3. Use Case Diagram……………………………………………………………………………………………………..….22
    4. Data-Flow Diagram…………………………………………………………………………………………………..…..23
11. Work Done in Semester VII and Current Status of the project……………………………………………….....24
12. Additional details as suggested by Project Guide………………………………………………………………………24
13. Conclusion…………………………………………………………………………………………………………………………….….24
14. References………………………………………………………………………………………………………………………….…….24
15. **ABSTRACT**

The growth of E-Banking has led to an ease of access and 24-hour banking facility for one and all. However, this has led to a rise in e-banking fraud which is a growing problem that affects users around the world. As card becomes the most prevailing mode of payment for online as well as regular purchase, fraud related with it is also increasing. Sophisticated online banking fraud reflects the integrative abuse of resources in social, cyber and physical worlds. However, there is very limited information available to distinguish dynamic fraud from genuine customer behaviour in such an extremely sparse and imbalanced data environment, making instant and effective detection more and more important and challenging. Thus, we propose a system which uses steganography technology along with various data mining techniques to effectively secure the e-banking process and prevent online fraud.

1. **EXISTING SYSTEM**

The major difference between the proposed system and existing system is that the existing system requires the user to directly enter the card details while making the payment. This puts the user details on high risk of being hacked and misused. The existing payment module is as follows:

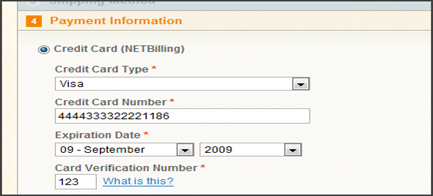


Fig 2.1 Existing Payment Module

* Once the user enters his details and proceeds with the transaction the existing system encrypts the details and forwards it to the bank server. However even though the way of encryption of the data and the algorithms used promise high level of security they are yet hacked by professional hackers. This is primarily because the data is just encrypted using a key and hacking this key is not a difficult task for high level professional hackers. Apart from encrypting the user details using a key there is no further mechanism to ensure safety of data from hackers.
* The existing system generates an OTP almost every time a transaction is done to add a level of security along with encryption of data. However, the generation of OTP is not done on some international sites and in case of network issues. In this situation, the user may use his master key to complete the transaction. The major drawback of using the master key is that then there is no OTP generation for that particular transaction. So, if the master key is hacked the card can be used for multiple fraudulent transactions.
* Additional features for ensuring secure transactions in the existing system include checking the regular pattern of usage of the card by the particular user. In case there is a change in the pattern of usage like a large transaction after a very small transaction, sudden increase in the frequency of purchase by the user or change in the location of user then the existing system suspects it to be a fraudulent transaction and immediately notifies the user with a call. In case it is not a fraud the execution of transaction is continued else the transaction is blocked immediately in order to avoid the improper usage.
* However, the main drawback of the existing system is that it fails to prevent the hacking of the user card details in the very first place. Our proposed system thus overcomes this drawback by encryption of the user details in an image using steganography. This makes hacking of details much more difficult thus adding an additional feature of security along with behaviour mining techniques to prevent e-banking frauds.

1. **PROBLEM DEFINITION**

The need for e-banking security is increasing exponentially due to the threats of potential hackers. The user entered details can be hacked even before they reach the intended destination. This is where steganography that is encrypting data in an image plays a pivotal role in prevention of such frauds. The image will appear just as a normal image and the data stored in it can be decrypted only by the concerned authorities using the intended key thus preventing a large number of frauds. Our system will also be using behaviour mining to keep track of the user’s previous transactions thus ensuring verification of every new transaction before confirming and committing it. This ensures implementation of fraud detection for every transaction. This encryption of data inside an image will be done using an android application making it easily available for all users. The advantage of steganography over cryptography alone is that the intended secret message does not attract attention to itself as an object of scrutiny. Thus, our proposed system ensures e-banking security by using both prevention and detection techniques.

1. **SCOPE**

The proposed system deals with fraud prevention and detection using steganography for encryption of data in an image and behaviour mining for detecting major deviations in the user's transaction pattern.

The three main modules of our system are as follows:

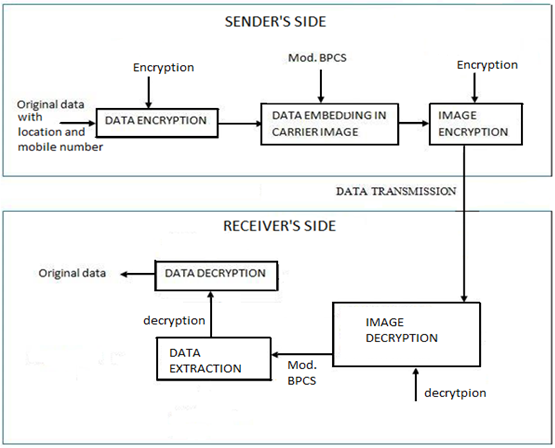
* Data encryption using image steganography.
* Behaviour mining for detecting fraud in spending pattern.
* Checking change in user location

1. **PROPOSED SYSTEM**

Proposed System consists of three main modules which are listed below:

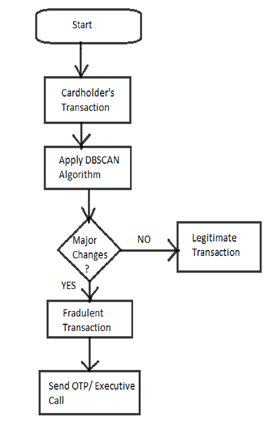
* **Data encryption using image steganography:**

The first module of our system deals with using steganography for secure transmission of data thus ensuring fraud prevention. The user enters his card details in the android application. These details along with the location of the device and the mobile number are encrypted using the AES encryption algorithm. The key used for encryption is encrypted using the RSA algorithm. This encrypted data is now embedded in a carrier image selected by the user by applying the modified BPCS algorithm and the image containing the user data is again encrypted using the AES encryption algorithm thus ensuring another level of security for the data. This image is now sent to the bank and the bank decrypts the data from the image using the secret key. It extracts the encrypted data from the image and decrypts it to access the entire data.



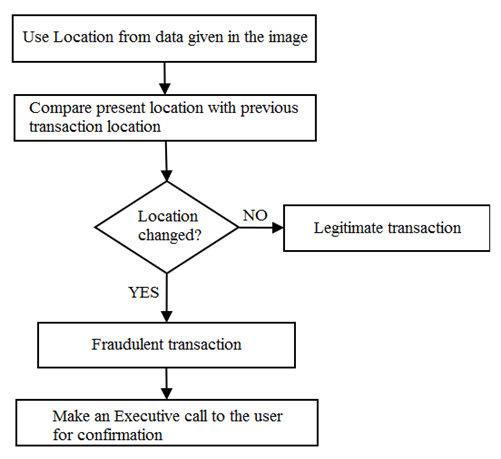
* **Behaviour mining for detecting fraud in spending pattern:**

The next module of our proposed system deals with behaviour mining of user’s previous transactions in order to detect frauds. This is implemented using DBSCAN Clustering algorithm. It uses cardholder’s spending behaviour to detect fraud.



* **Checking changes in user location:**

The final module deals with location checking of the transaction. The location embedded in the image is compared with the location of the user’s previous transactions and in case the location varies then a call is made to the user from an executive for confirmation of transaction validity otherwise the transaction is considered legitimate.



1. **PROGRAMMING TOOLS**

|  |  |  |
| --- | --- | --- |
| Sr. No. | Tool/Software | Reason |
| 1. | Star UML | UML Diagrams |
| 2. | Microsoft Project Professional | Project Plan |
| 3. | Microsoft Word | Project Report |
| 4. | Android Studio | Android App Development |
| 5. | JavaScript | Code Development Language |
| 6. | PHP | Server-Side Scripting Language |
| 7. | Python | Code Development Language |
| 8. | Java | Code Development Language |
| 9. | MySQL | Database Management |

1. **HARDWARE AND SOFTWARE PLATFORM REQUIREMENTS**

* **Hardware Requirements:**

o Server m/c

* + i3 processor (4th generation)
  + 50GB hard disk
  + 1GB RAM
  + Working Internet connection

o User device

* 1GB RAM
* Dual Core processor
* GPS enabled
* Net connectivity
* **Software Requirements:**

o Server machine

* Windows 7
* Android Software Development Kit (SDK)
* SQL/Oracle Server
* SMS gateway

o User device

* Android v4.1 and above

1. **SOFTWARE PROJECT MANAGEMENT PLAN** 
   1. Project Description

E-Banking Security is a project that aims at providing a secure e-banking system to all end users. The user enters his/her card details in the android application of our system wherein his details are encrypted and then embedded in an image using steganography. This image used by the user for carrying out the respective transaction ensures that user details are securely transmitted. At the server end once the details are extracted from the given image the transaction is checked for being a fraudulent or honest transaction and action is taken accordingly. Thus, both fraud prevention and fraud detection is done by the system. This project is expected to be delivered in the month of April 2017 before the ending of the even semester of KJSCE Calendar.

* 1. Project Deliverables

|  |
| --- |
| **Deliverable** |
| Software Requirements Specification |
| Software Project Management Plan |
| Software Design Document |
| Final Prototype and User Guide |
| Test Cases |
| Maintenance Plan and Acceptance Report |

* 1. Software Process Model

A software process shows the basic structure for software project development. The most suitable process model for the development of our system will be Waterfall model. The reason being, all the requirements are known upfront and system can be built through this method in short span of time.

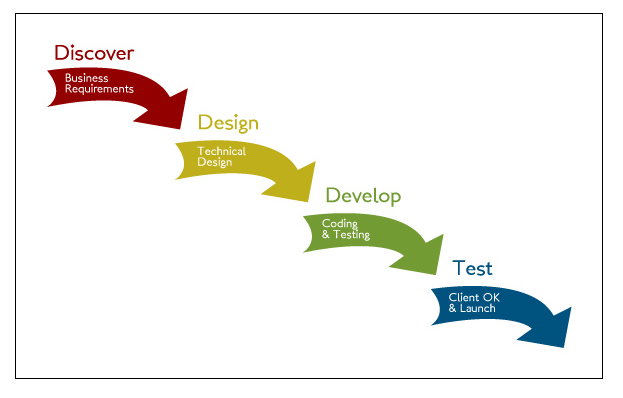


Fig 8.3.1. Waterfall Model

Advantages of waterfall model are as follows:

* Waterfall model is simple to implement and also the amount of resources required for it are minimal.
* This methodology is preferred in projects where quality is more important as compared to schedule or cost
* Project management, both at internal level and client's level, is easy again because of visible outputs after each phase. Deadlines can be set for the completion of each phase and evaluation can be done from time to time, to check if project is going as per milestones.
  1. Roles and Responsibilities

The work is evenly divided among three members as follows:

|  |  |
| --- | --- |
| Roles | Team member |
| Database Administrator | 1. Namrata Devadiga |
| Development and Coding | 1. Harshad Kothari 2. Hardik Jain 3. Namrata Devadiga |
| Designing | 1. Hardik Jain 2. Harshad Kothari |
| Testing | 1. Hardik Jain 2. Namrata Devadiga 3. Harshad Kothari |

* 1. Tasks

The initial functional requirements of the project can be divided into a number of tasks as seen below:

|  |  |
| --- | --- |
| Sr. No. | Task |
| 1. | Requirement Gathering and Analysis |
| 2. | Project Planning |
| 3. | Design |
| 4. | Implementation |
| 5. | Testing |

* Task 1:

Name: Requirement Gathering and Analysis

Description: This task mainly deals with understanding the scope of project and the basic requirements related to the input output of the system. It involves a detailed study of some of the existing systems, their drawbacks and detailed gathering of requirement of our entire system.

Deliverables: Software Requirement Specification Document

Resources: IEEE papers related to similar concepts, reference Books containing required concepts, previous projects of similar kind

Dependencies and Constraints: The detailed project planning and design is not possible with proper understanding of all system requirements and detailed study of the similar existing systems.

Risks: Requirements of the project are partially understood and there is no proper knowledge of the given system. The information collected to be analysed for understanding of the project is insufficient for providing comprehensive learning of the system.

* Task 2:

Name: Project Planning

Description: Project planning includes identifying various methodologies for implementing the project, such as finalizing a software development plan and acting accordingly. Selecting appropriate algorithms and identifying different tools required for project development are also identified during planning phase. The project plan should be drafted in a formal document prepared for the complete project development and execution.

Deliverables: Literature Survey and Software Project Management Plan.

Resources: IEEE papers related to similar concepts, reference Books containing required concepts, template of SPMP Document.

Dependencies and Constraints: Thorough understanding of all project requirements is a necessity. If requirements are not understood properly then project plan will not represent the correct proposal for development of the project. Project Planning needs to be done entirely before documentation of SPMP. The SPMP document depends completely on correct project planning.

Risk: Project plan is unrealistic with deadlines which are unachievable, no proper understanding of the algorithms, tools and technologies to be used in the project, unacceptable allocation of duties to team members. Improper estimation of budget and resources required.

* Task 3

Name: Design

Description: Identifying different components of the system and decide a proper flow which has to be followed during the course of the project development. It gives a complete description of the course of execution of various modules of the system.

Deliverables: Software Design Document

Resources: SRS document, UML diagrams

Dependencies and constraints: All design parts and necessary diagrams such as use case diagram, class diagram, ER diagram, data flow diagram etc. must follow established standard. Complete knowledge of the SRS Document is required for signifying complete scope of the project.

Risk: The flow of process execution not demonstrated precisely by the models may lead to confusion regarding the order of carrying out various processes.

* Task 4:

Name: Implementation

Description: In this phase we convert the system design into source code. It includes various front end and back end tools and technologies for effective execution of the proposed system. Here the task is to develop the main programs and codes which perform various operations in the project and to implement it along with required user interface.

Deliverables: Source code, executable files, library files, android application.

Resources: DFD, necessary software.

Dependencies and Constraints: Software development team should have proper in depth knowledge of the technologies used. There should be proper compatibility between the different languages used.

Risk: Not being able to complete programming of the system within the given deadline, code not being implemented correctly in specified language, change in scope of the system, maintenance of proper quality while balancing the time and scope constraints. Different modules are not being combined properly to give the system as a whole.

* Task 5:

Name: Testing

Description: Unit testing, documentation, integration of each class, integration testing and documentation inspection. Final step before deployment is testing the software on various parameters by making a test which checks every aspect of scope defined.

Resources: Completely developed system, test procedures, expert guidance

Dependencies and Constraints: Testing should be performed not only by the developer of the system but also by the user of the system or an external expert. This is done in order to get a complete insight on whether the system is working as desired. Knowledge about the working of the system and the expected outputs also need to be known to the one testing the system.

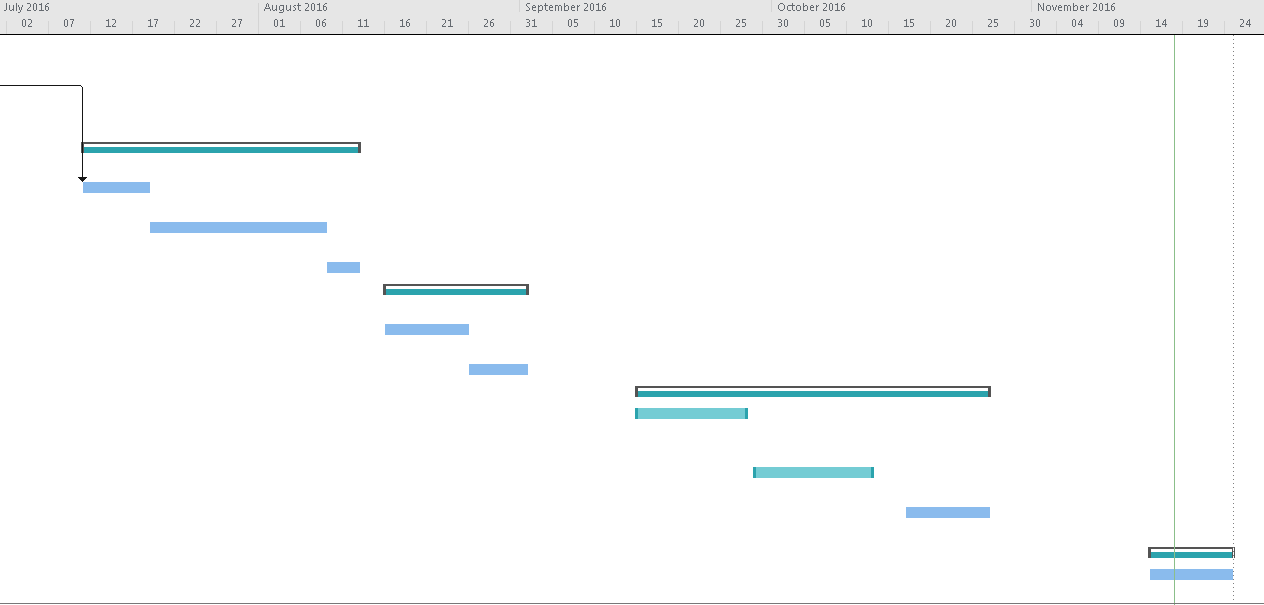
Risk: The testing cases or plan does not cover all types of errors and some major or minor error bypass the testing phase thus increasing the risk of mistakes in the system after deployment and increasing cost of dealing with such issues.

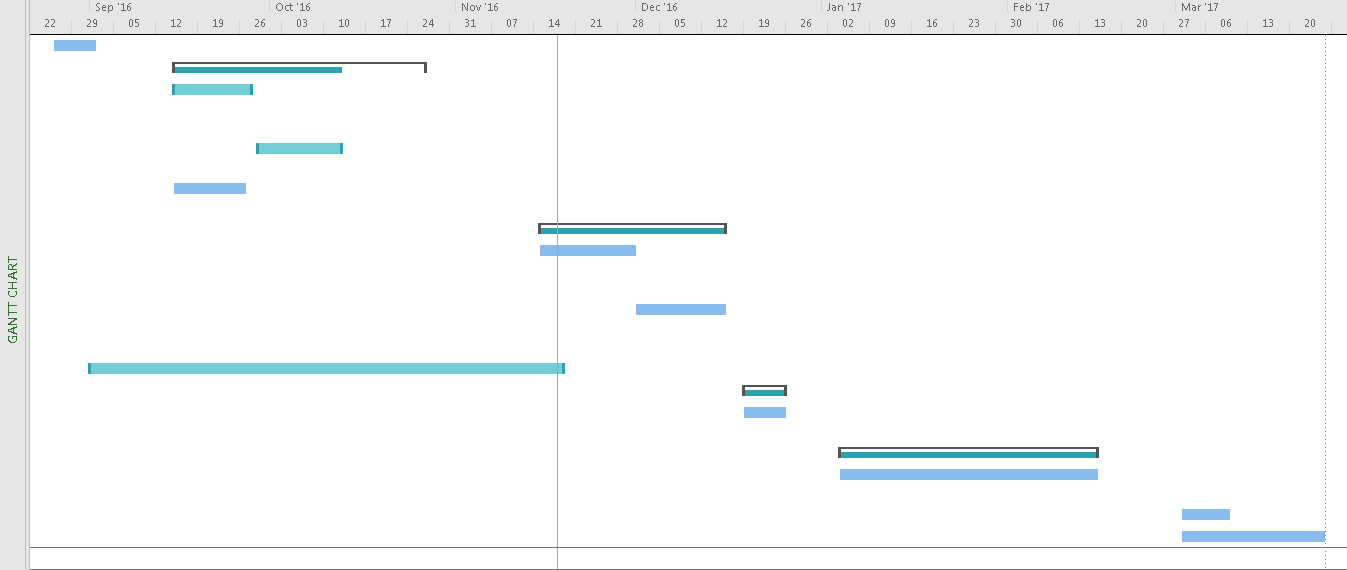
* 1. Timetable

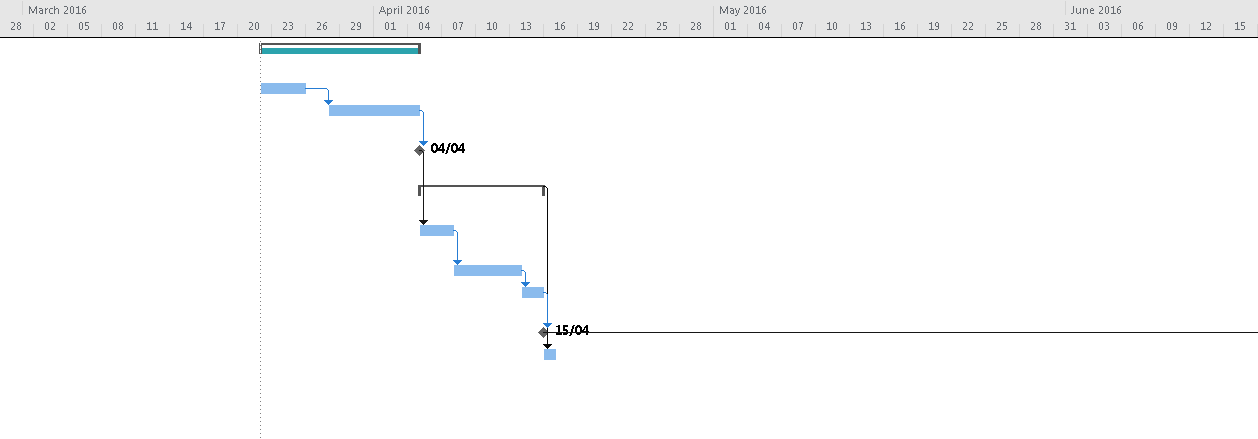
The following table shows the tentative schedule for development and related activities:

|  |  |  |  |
| --- | --- | --- | --- |
| **TASK** | **DESCRIPTION** | **START DATE** | **END DATE** |
| **Problem Formulation and Definition.** | Deciding topic for project and preliminary research | March 2016 | April 2016 |
| **Project Objective and Scope Declaration.** | Detailed research and understanding requirements. | March 2016 | April 2016 |
| **Project Approval Seminar.** | Acceptance of project scope with some suggestions. | - | 16th April 2016 |
| **Requirement Gathering** | Collection of information, Study of Algorithms and Literature Survey | July 2016 | August 2016 |
| **Designing-I** | Designing basic layout (UI) of Android App | August 2016 | September 2016 |
| **Coding (Phase 1)** | Implementation of Data Encryption Algorithms | September 2016 | October 2016 |
| **Coding (Phase 2)** | Implementation of Clustering Algorithm | November 2016 | December 2016 |
| **Documentation** | Generation of various reports as per requirements | September 2016 | November 2016 |
| **Designing-II** | Designing the execution platform on server side | December 2016 | December 2016 |
| **Coding (Phase 3)** | Implementation of BPCS Algorithm (Steganography) | January 2016 | February 2016 |
| **Testing** | Testing performance of various models | March 2016 | March 2016 |
| **Final Documentation** | Preparing the final project document (Black Book) | March 2016 | March 2016 |

Gantt Chart:







1. **SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT**
   1. Product Overview

The proposed system deals with fraud prevention and detection using steganography for encryption of data in an image and behaviour mining for detecting major deviations in the user’s transaction pattern.

* 1. User Interfaces

The system shall provide easy option for user data entry ensuring convenience of all the users. The system shall provide a range of images for user to choose from for the encryption process. The UI shall provide simple graphical interfaces, similar to a material design, to allow the user to easily enter details and choose an image for encryption process.

* 1. Functional Requirements:

The system must have following modules:

* **Data Encryption:**

**Description:** This module is a part of the android application at the user end. The app takes the user card details along with user location. The user then selects an image from the provided range of images. The app first encrypts the user entered data. This data is then embedded in the selected carrier image. The image is then encrypted again. This is the final image to be used by the user for a given transaction. The priority of this module is high as user data needs to be encrypted properly for safe and secure transaction.

**Priority:** High

**Stimulus**: User enters a user id that has already been taken by another individual.

**Response**: A message saying “This user id has been taken. Re-enter another id.” will be displayed.

**Stimulus**: User fills in information partially (i.e. only in few input fields) and clicks on the enter data button.

**Response**: A message saying “Fill all the \* input fields” will be displayed.

**Stimulus**: User fills in all the information fields and clicks on enter data button

**Response**: User will be directed to selecting an image page, where in user will have to choose an image for data encryption.

**Stimulus**: User chooses an image from the selected range of pictures and clicks on enter.

**Response**: User will be shown the final image which can be used for the transaction.

**Stimulus**: User does not choose an image from the selected range of pictures and clicks on enter.

**Response**: User will be prompted to choose a picture to proceed further.

**Requirements:** A number of input fields for taking user credit card credentials will be present in this module. An “Enter Data” button will be provided which the user must click on once he’s done filling in all the information. The user must choose an image from the selected set of images. The user location services should be on for the application to access user location.

* **Data Decryption:**

**Description:** The data decryption module is part of the bank system. The system first decrypts the image entered by the user. It then extracts the data from the image. Lastly it decrypts the data for retrieval of the user card credentials and location. In case of valid details ensure proper execution of transaction.

**Priority:** High.

**Stimulus**: The system decrypts the user data and checks validity.

**Response**: In case of valid data the data is moved to the next module for checking for fraudulent transactions.

**Requirements:** The only input to this module of the system is the encrypted image sent by the user. If the user details entered are faulty system intimates the user regarding wrong details. If all the details are correct the data extracted is moved to the next module for checking of frauds.

* **Fraud Detection using spending behaviour**

**Description:** This module of our proposed system deals with behaviour mining of user’s previous transactions in order to detect frauds. This module mainly deals with a user’s previous spending pattern whether high, medium or low. In case there is a minor deviation from the users usual behaviour the system generates an OTP to ensure there is no fraudulent activity even in case of a small variation. In case there is a major deviation from the user’s usual behaviour the system considers it as a major threat and an executive call will be made directly to the registered user number, thus ensuring that the security measures do not become a troublesome experience for the system user. Priority of this module is high for separating fraudulent transactions from real ones.

**Priority:** High.

**Stimulus**: The user spending pattern varies highly from previous transactions.

**Response**: An executive call is made to the user to ensure user is legitimate.

**Stimulus**: The user spending pattern varies moderately from previous transactions.

**Response**: OTP is made and sent to the user to ensure user is legitimate.

**Stimulus**: The user spending pattern does not vary from previous transactions.

**Response**: The user data is forwarded to the next module for checking fraudulent transactions.

**Requirements:** It accesses the user’s previous transactions from the database along with current transaction details. In case doubts of fraudulent transaction either otp or executive call is made depending on the seriousness of the transaction. If the data is correct then it is forwarded to the next module.

* **Fraud Detection using location:**

**Description:** This module checks whether a transaction is fraudulent or not based on the location of the current transaction as well as all the previous transactions. The user credentials as well as location is used. This is compared with all other previous transaction locations to ensure not any major deviation in location of the user.

**Priority:** High.

**Stimulus:** The user location deviates majorly from all previous transactions.

**Response:** An executive call is made to the user to ensure validity of user.

**Stimulus:** The user location matches with all previous transactions.

**Response:** The user data is transferred to the next module for checking for any more frauds.

**Requirements:** This module accesses the user location for current transaction and location for all previous transactions from the database. The location embedded in the image is compared with the location of the user’s previous transactions and in case the location varies then a call is made to the user from an executive for confirmation of transaction validity otherwise the transaction is considered legitimate.

* 1. Non-Functional Requirements:
* **Performance:**

The app will lag in performance if Internet speed is below 5 kbps, as the login credentials and payment options have to be done via Internet.

* **Safety:**

User should not share encrypted image with any other user. User should enter the otp when needed. User should answer executive call when needed.

* **Security:**

User should not share card details with any other user.

* **Availability:**

The system is available 100% for the user and can be used all 24 hours a day.

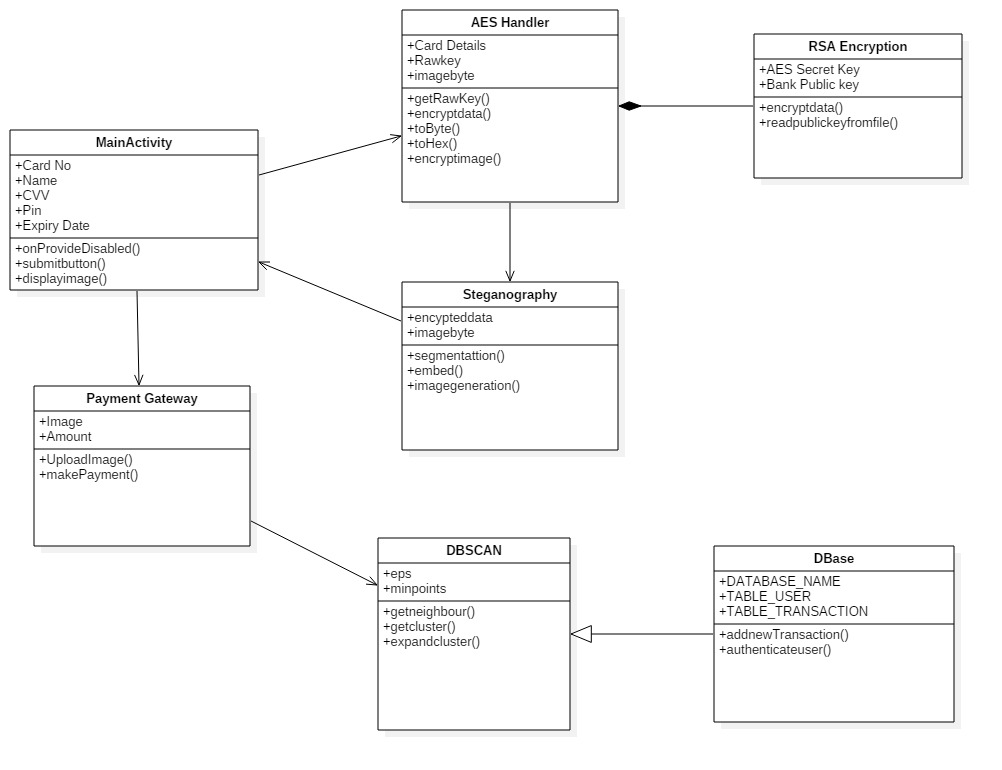
* **Reliability:**

The system is reliable because of the secure transmission of data by the admin.

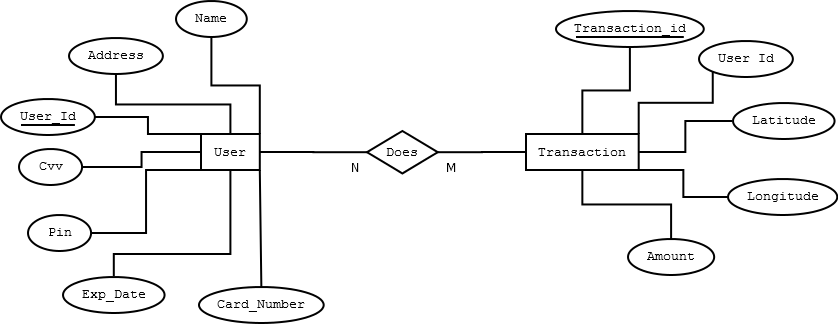
* **Usability:**

The system allows the user to access the system using Internet services. The system uses a web browser as an interface. The system is user friendly and self-explanatory.

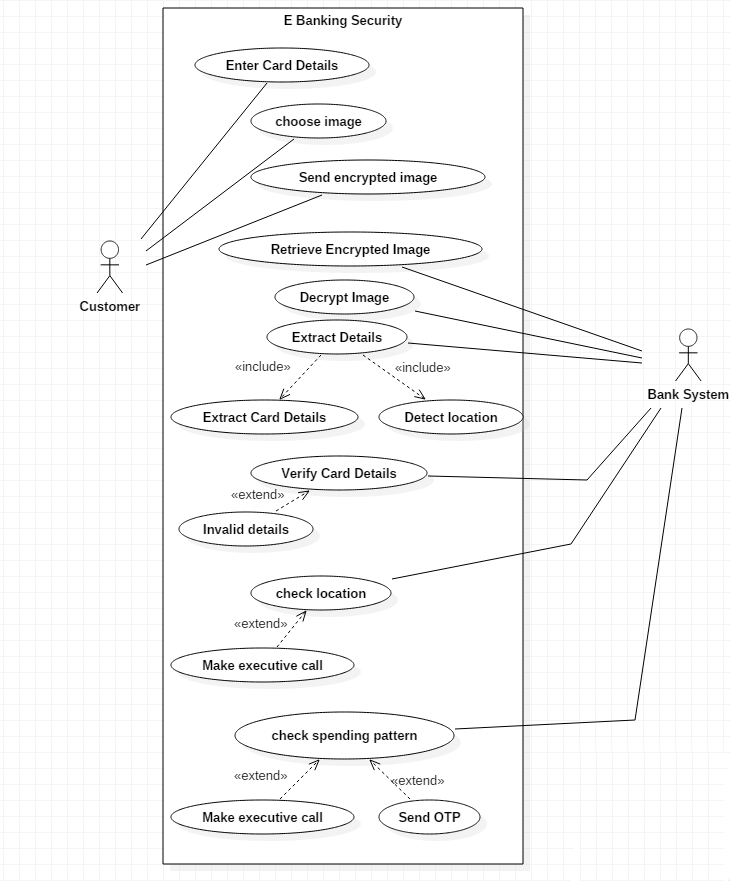
1. **SOFTWARE DESIGN DOCUMENT (ALL APPLICABLE UML DIAGRAMS)**
   1. Class Diagram



**10.2** ER Diagram

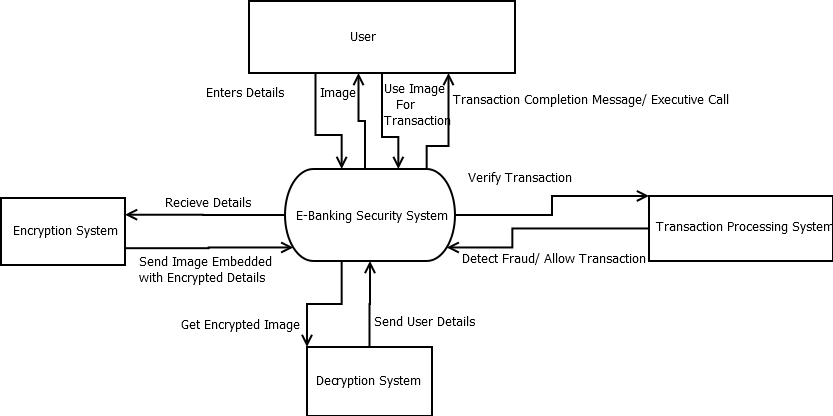


**10.3** Use Case Diagram

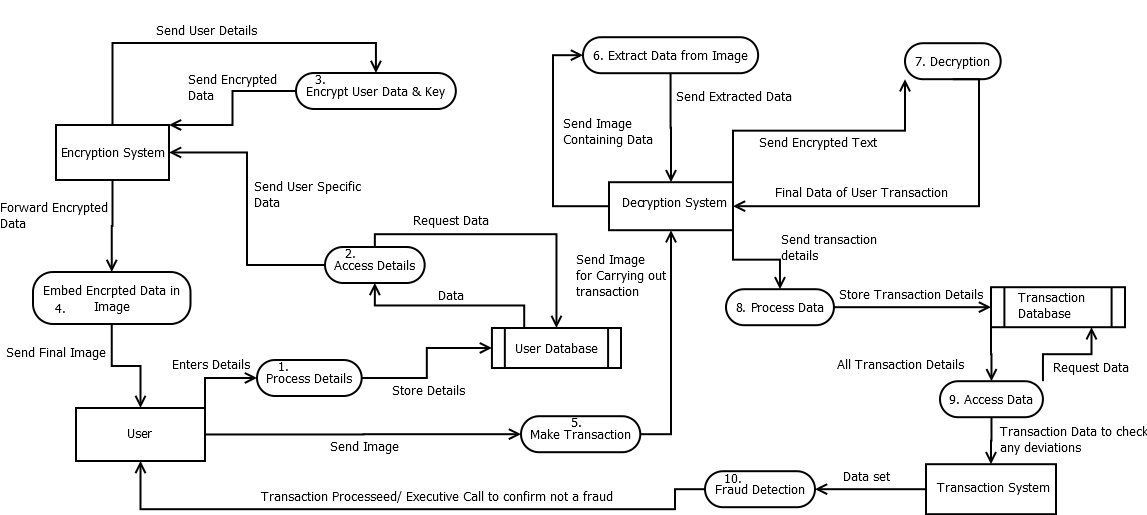


**10.4** DFD diagram

Level 0



Level 1



1. **WORK DONE IN SEMESTER VII AND CURRENT STATUS OF THE PROJECT**

We started by studying various problems and solutions presently deployed in banks. After conducting extensive research for defining the requirements and objectives we finalized the algorithms we will be using for our project. We designed the basic UI for the required android application which asks the user for his/her card details (name, card number, pin, expiry date, cvv) and gets the location of the user with the help of internet and GPS without the knowledge of the user. These card details are then encrypted using the Advanced Encryption Standard (AES) Algorithm with a key size of 256 bits. The algorithm is implemented using Java. Now, this AES key is encrypted using RSA 2048 Algorithm in order to share it with the bank using the bank’s public key. Even this public key is only known to the application and not the user and is hidden in the root folder of the mobile. For the Data Mining part on the server side, we have started implementation of DBScan algorithm with respect to the location of the user.

1. **ADDITIONAL DETAILS AS SUGGESTED BY PROJECT GUIDE**

After the approval seminar, we were suggested for a change in the clustering algorithm as we had first decided to use K-means algorithm. We also switched from RSA to AES for data encryption after being asked to replace RSA.

We got a few suggestions in the first progress seminar to look for other options in order to share the key with the bank as RSA would be a little slow with more processing overhead. However, after referring the suggested book and a few technical papers, we came to conclusion that RSA would not slow down our process as it was only the key that had to be encrypted using it. Suggestions on timestamp and hashing for image encryption are yet to be worked on.

1. **CONCLUSION (ON LITERATURE SURVEY)**

As the need for security in online transactions has increased manifold with increased threats and attempts to steal user’s data for fraudulent transactions, it is important to upgrade security mechanisms accordingly from time to time. Thus, our system recognizes the vulnerabilities or threats in the existing system and uses latest technologies to implement and combat such threats to give more and more security to each and every user and keeps attackers away.

1. **REFERENCES**

* Mandeep Singh, Perminderpal Singh, Rajan Kumar “Fraud Detection By Monitoring User Behaviour And Activities” International Conference on Computer and Intelligent Systems May 30-31, 2014.
* Vaishali, Abhishek Kajal “Increasing Data Hiding Capacity of Carrier Image Using BPCS Steganography” International Journal of Science and Research (IJSR) Volume 4, Issue 5, May 2015.
* Rituraj Rusia, Munendra Kumar Mishra, R. K. Tiwari “More Advanced Steganography Using BPCS” International Journal of Computer Engineering and Applications, Volume VII, Issue II, Part II, August 2014. ArshiaAzam, RumanaFirdous, F. Asma Begum “High Security Image Steganography Using RSA Algorithm” Journal of Innovation in Electronics and Communication, Vol 4, July-Dec ’14.