

Customer Relationship Management Tool

A PROJECT REPORT

submitted by

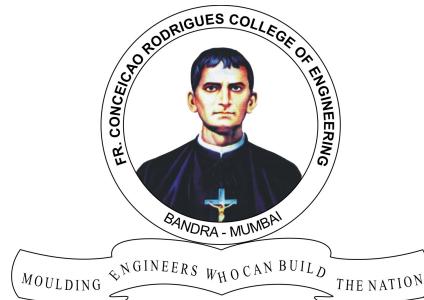
ASHTAMI BHULESKAR (ROLL NO.:6123)

VIBHATI JOSHI (ROLL NO.:6135)

SANIKA SAVE (ROLL NO.:6154)

**in partial fulfillment for the award of the degree
of**

**BACHELOR OF ENGINEERING
IN
INFORMATION TECHNOLOGY**

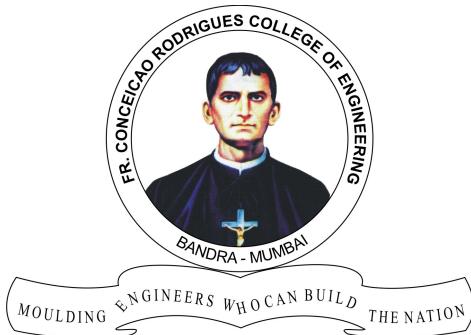


**Department of Information Technology
Fr. Conceicao Rodrigues College Of Engg
Fr. Agnel Ashram, Bandstand, Bandra (W),
Mumbai - 400050
April - 2014**

CERTIFICATE

Certified that this project report "**CUSTOMER RELATIONSHIP MANAGEMENT TOOL**" is the bonafide work of "**ASHTAMI BHULESKAR(6123)**, **VIBHATI JOSHI(6135)** , **SANIKA SAVE(6154)**" who carried out the project work under my supervision.

Certified by



Internal Guide

Prof. Dharmesh Rathod
Information Technology

Principal

Dr. Srika Unnikrishnan

HOD

Prof. Mahesh Sharma
Information Technology



.....
Internal Examiner

.....
External Examiner

Abstract

A huge amount of research has been done in the field of Customer Relationship Management in the recent years from a business point of view. These efforts have been towards determining factors of CRM, efficiency parameters, how it affects market trends and company profits, etcetera. The fields of technology and business in CRM have been fairly disparate in most of these efforts. This project aims to bridge this gap of the technical implementation and its subsequent business implications by integrating them together in a fulledged CRM solution. It is an attempt to take the middle path between software development and management issues and delicately balancing them both in a unique two way system that fulfils the criteria of both the company and the customer. This software aims to perform analysis of market trends and subsequent customer habits also finding new potential profitable customers for the company using data mining . On the whole, the project offers a proficient and dynamic merger of the fields of technology and business perspectives. Thus, it provides a wholesome solution to tackle customer relationship strategies thereby making the model as profitable for the company and the same time gaining high levels of customer satisfaction.

Acknowledgement

We have great pleasure in presenting the report on "Customer Relationship Management Tool". I take this opportunity to express my sincere thanks towards the guide [Prof. Dharmesh Rathod], C.R.C.E, Bandra (W), Mumbai, for providing the technical guidelines, and the suggestions regarding the line of this work. We enjoyed discussing the work progress with him during our visits to department.

We thank Prof. Mahesh Sharma, Head of Information Technology Dept., Principal and the management of C.R.C.E., Mumbai for encouragement and providing necessary infrastructure for pursuing the project .

We also thank all non-teaching staff for their valuable support, to complete our project.

Ashtami Bhuleskar	Roll No. 6123
Vibhati Joshi	Roll No. 6135
Sanika Save	Roll No. 6154

Contents

List of Figures	iii
Glossary	iv
1 Introduction	1
1.1 Data Mining	1
1.2 Motivation	2
1.3 Objectives	2
1.4 Scope of the project	2
1.4.1 Functional Requirements	3
1.4.2 Non-Functional Requirements	3
2 Literature Review - Both Apriori And K-Mean Clustering Algorithms	4
2.1 Improving Efficiency of Apriori Algorithm Using Transaction Reduction	4
2.2 Analysis of Customer Behavior using Clustering and Association Rules	5
2.3 Architecture for Customer Relationship Management Approaches in Financial Services	5
2.4 Data Mining model for Trade in CRM System	6
3 Problem Statement	7
3.1 Drawbacks of Apriori Algorithm For Frequent Itemset	7
3.1.1 Typical Apriori algorithm	7
3.1.2 Shortcomings	8
3.2 Modification in the algorithm	8
4 Project Description	9
4.1 Overview of the project	9
4.2 Module Description	10
4.2.1 Module 1: Detail extraction	10
4.2.2 Module 2 : Customer Tracking	10
4.2.3 Module 3 : Lead capture	11
4.2.4 Module 4 : Report generation	12
4.2.5 Module 5 : User Interface	12
4.2.6 Architecture	13
4.2.7 UML Diagram	14
4.2.8 Class Diagram	15
4.2.9 Data Flow Diagrams	16

4.2.10	Activity Diagram	18
4.2.11	Database Design	19
4.2.12	Input Design	21
4.2.13	Output Design	21
4.2.14	System In Work	25
4.2.15	System In Work - Customer Tracking	25
4.3	Algorithms	27
4.3.1	Apriori Algorithm	27
4.3.2	K-Means Algorithm	28
5	System Testing	31
5.1	Unit Testing	31
5.2	Integration testing	31
5.3	Load Testing	31
5.4	Black Box Testing	32
6	Implementation Details	33
6.1	Project Lifecycle	33
6.2	System Specifications	34
6.2.1	Minimum Hardware Requirements	34
6.2.2	Minimum Software Requirements	34
6.3	System Description	34
6.3.1	Front End	34
6.3.2	Back End	35
6.4	Methodology	36
6.4.1	Microsoft Excel	36
6.4.2	Adobe PDF	36
6.4.3	Object Oriented Programming Techniques:	37
6.4.4	NetBeans 7.3	37
6.4.5	Java Swings	37
6.4.6	Apache POI Libraries	37
6.4.7	iText Libraries	38
7	Conclusion And Future Enhancements	39
7.1	Conclusion	39
7.2	Future Enhancements	40
Appendix A - Source Code		46
Bibliography		48

List of Figures

4.1	Architecture	13
4.2	UML Diagram	14
4.3	Class Diagram	15
4.4	Data Flow Diagram : Level 0	16
4.5	Data Flow Diagram : Level 1	17
4.6	Activity Diagram	18
4.7	Master Database	19
4.8	Car Loan Database	19
4.9	Master Database	19
4.10	Education Loan Database	20
4.11	Fixed Deposit Database	20
4.12	Home Loan Database	20
4.13	Insurance Database	21
4.14	Mutual Fund Database	21
4.15	Personal Loan Database	22
4.16	Recurring Account Database	22
4.17	Saving Account Database	23
4.18	Lead Capture Database	23
4.19	Detail Extraction User Interface - Input	24
4.20	Customer Tracking User Interface - Input	24
4.21	Lead Capture User Interface - Input	25
4.22	Detail Extraction User Interface - Output	25
4.23	Customer Tracking User Interface - Output1	26
4.24	Customer Tracking User Interface - Output2	26
4.25	Transaction Database	27
4.26	Lead Capture User Interface - Output	27
4.27	Lead Capture Output PDF - Cluster 1	28
4.28	Lead Capture Output PDF - Cluster 2	28
4.29	Lead Capture Output PDF - Cluster 3	29
4.30	Lead Capture Output PDF - Cluster 4	29
4.31	Customer Tracking - Console Output	29
4.32	Lead Capture - Console Output 1	29
4.33	Lead Capture - Console Output 2	30
6.1	Spiral Model	33

Glossary

BI Business Intelligence.

CRM Customer Relationship Management.

DB DataBase.

DM Data Mining.

DW Data Warehousing.

KM Knowledge Management.

SOT Size Of Transaction.

Chapter 1

Introduction

Customer Relationship Management Tool is a relatively new but highly potent field of emerging business models. In todays competitive business environment, it is highly crucial to find new customers and retain the old ones with quality service and target on customer satisfaction. CRM is the perfect tool and strategy to focus on the customers in order to maximise profits and strengthen a businesss market position considerably. Since customer satisfaction is a very subjective aspect, it is ambiguous to quantify the success or failure of the CRM systems. CRM solutions also include features and functions for sophisticated reporting and analytics, with configurable dashboards that help you control your view of the data. Businesses often face a challenge to get the balance right between customer service and sales to ensure they deliver their targets. Whether your challenge is lead generation, capturing customer data or measuring how the effectiveness of your marketing campaigns, the solution is :

- understand customer behaviour and buying habits
- persuade your prospects that you are the right company to do business with
- move them into the sales cycle

We are planning to combine the intricacies of the Information Technology along with the subtleties of the business perspectives. Our project aims at comparing how a business model that inculcates CRM fares better in terms of profits, market trends and customer satisfaction than a traditional supply driven model that ignores the customer.[1][2][3]

1.1 Data Mining

Customer Relationship Management is used to make more efficient business-customer relationships in order to maximize client satisfaction and thereby improve customer loyalty and retention. Data mining techniques are useful to analyze the customer behavior. Data Mining grew from the persistent growth of techniques used to interrogation masses of data. Prediction is done using the relevant data set taken from the database on the basis of the attributes. Customer data and customer relationships should be stored and maintained. Customer loyalty and retention are improved when customer satisfaction is more. Customer satisfaction and loyalty are attracted more attention in recent years.[4]

Data mining creates the ideal environment for making customer relationship management (CRM). The approaches of how data analytics can be used to make various CRM functions like customer segmentation, communication targeting, retention, and loyalty much more effective are present and implemented before. CRM is considered as the database marketing of an organization with the database of customers. CRM identifies the most profitable customers and provides the highest level of service to them. Customer centric approach helps to improve customer satisfaction and thereby maximize customer retention.[5]

1.2 Motivation

Customer Relationship Management is an enterprise wide effort to acquire and retain profitable customers. CRM focuses on forging meaningful, long term and sustainable customer relationships. These relationships focus on deriving a value addition for both the company and the customer. The business models that adopt CRM as a central strategy are often from those sectors that accept the customer to be the core of their market performance. CRM is a new yet rapidly evolving field to optimise business models. It is thus essential to have a complete integrated CRM solution that not just caters to the needs of the customers but also provides ease of use to the servicers. These analysis measures will extensively help in drawing a comparison between the performance of CRM centric businesses and the rest. This thing motivated us to choose this topic.

1.3 Objectives

Project is designed to find existing customers which can be more profitable for the company if tracked in a proper way. This can be achieved by analyzing their product usage habits. Such frequently appearing combinations can be termed as common behaviour which can be used to track down other customers. Also the other objective is to target people of the region under consideration based on the existent schemes of bank. These people can be future aspirants of the bank. The project focuses on combining the business aesthetics of the customer relationship module along with its actual technical implementation. It aims at using simple methodologies and data mining algorithms to effectively extract and analyse data. The resultant sets give an in depth knowledge of customer trends, existing patterns and probable targets to be achieved. These patterns, as visually represented, will be used in contributing to company growth and further marketing strategies. In short objective of project is:

- To track existing customers.
- To find potential customers.
- To analyse trends to formulate popular patterns.

1.4 Scope of the project

The system has a very concise scope but appeals to a large audience. Something that is very common and baseline for the success of any organization is to find new and retain old customers.

We plan to use this golden rule to make this process simpler. However this system can be easily extrapolated to any system that has a database of customers that needs to be tracked down. For e.g. Supermarkets.

1.4.1 Functional Requirements

Whenever user sends the request, system has to load database, perform algorithm computation on that data and return back the results in a visually comprehensive way. User will interact with system using application interface. System has to recognise the request and forward it to corresponding module. Server will generate output and send back.

1.4.2 Non-Functional Requirements

- Accessibility :- The user should be able to access the system whenever he wants .
- Availability :- The records of all customers and reports generated should be available for themanager for access round the clock.
- Maintainability :- The system should be easy to maintain. No supervision should be needed once server is started.
- Performance :- The system should show optimum performance under heavy load.
- Compatibility :- The system should be compatible with all operating systems supporting .xls file format.
- Response Time :- The system should take minimum response time to compute and send reply to user.

Chapter 2

Literature Review - Both Apriori And K-Mean Clustering Algorithms

Customer Relationship Management is very vast concept and lot of work has been done in this sector. Following papers were very useful in understanding the basics as well as the detailed knowledge of algorithms.

2.1 Improving Efficiency of Apriori Algorithm Using Transaction Reduction

Association Rules:

The concept of the association rules was first proposed by R.Agrawal. It is used to describe the patterns of customers' purchase in the supermarket. The association rules can be formally defined as Definition 1: Let $I=\{i_1, i_2, i_3, \dots, i_m\}$ be finite itemsets. D is a transactional database. Where $i_k (k \in 1, 2, \dots, m)$ is an item, and T_{id} is the exclusive identifier of transaction T in transactional database.

Definition 2: Let $X \in I, Y \in I$, and $X \cap Y = \emptyset$. The implication of the form $X \geq Y$ is called an association rules.

Definition 3: Let D is a transactional database. If the percentage of transactions in D that contain $X \cup Y$ is $s\%$, the rule $X \geq Y$ holds in D with Support s . If the percentage of transactions in D containing X that also contain Y is $c\%$, the rule $X \geq Y$ has Confidence c . The definitions of probability are, $\text{Support}(X \geq Y) = P(X \cup Y)$ (1) $\text{Confidence}(X \geq Y) = P(Y | X)$ (2) Rules that satisfy both minimum support threshold (minsup) and minimum confidence threshold (minconf) are called strong rules.

Definition 4: If the support of itemsets X is greater than or equal to minimum support threshold, X is called frequent itemsets. If the support of itemsets X is smaller than the minimum support threshold, X is called infrequent itemsets.

Association rules are the main technique to determine the frequent itemset in data mining. Apriori algorithm is a classical algorithm of association rule mining. This classical algorithm is inefficient due to so many scans of database. And if the database is large, it takes too much time to scan the database. In this paper, we proposed an Improved Apriori algorithm which reduces the scanning time by cutting down unnecessary transaction records as well as reduce the redundant

generation of sub-items during pruning the candidate itemsets, which can form directly the set of frequent itemsets and eliminate candidate having a subset that is not frequent.[6]

2.2 Analysis of Customer Behavior using Clustering and Association Rules

The analysis of customer behavior is used to maintain good relationship with customers. It maximizes the customer satisfaction. We can also improve customer loyalty and retention. The aim of this paper is to develop a very useful trend for launching products with configurations for customers of different gender based on past transactions. Based on the previous transactions of the customers, prediction is done and data is estimated with the help of clustering and association rules. This paper proposes an effective method to extract knowledge from transactions records which is very useful for increasing the sales. Customer details are segmented using k-means and then Apriori algorithm is applied to identify customer behavior. This is followed by the identification of product associations within segments. This paper aims to develop a new trend and launch a new series of products using the previous transactions of the customers.[7]

Predictive analytics analyzes historical data to make good predictions. Such predictions rarely take the form of absolute statements, and are expressed as values that correspond to the behavior taking place in the future.

The customers with similar purchasing behavior are first grouped by means of clustering techniques. Finally, for each cluster, an association rules are used to identify the products that are frequently bought together by the customers.[8]

2.3 Architecture for Customer Relationship Management Approaches in Financial Services

Increasing competition and decreasing customer loyalty have led to the emergence of Customer Relationship Management (CRM), a concept that focuses on the nurturing of customer relationships. To build long lasting, profitable relationships with customers, CRM requires the management of customer-related knowledge which entails knowledge from, about and for customers Over the past few years CRM has played an increasingly important role in the financial services industry. During this time the majority of financial services companies have, with varying success, conducted CRM implementation projects. Only a few companies have particularly realized the implementation of a CRM strategy aligned with company profitability and employing integrated information systems for both performance measurement as well as for the control of marketing, sales, and service processes The objectives of this contribution are:

- the development of a CRM reference architecture on the process and systems levels for the description and classification of CRM approaches in financial services companies, and
- the derivation of the types of CRM approaches in the financial services industry and their description by means of the CRM architecture.

The developed CRM architecture and defined CRM types can be used in the analysis of CRM in companies to determine the actual state and to define a target state. In the next section we describe the foundations and strategic objectives of CRM. We thereafter develop the CRM architecture for the process and system levels. Thus, we provide an holistic picture of CRM over the levels strategy, processes and information systems In the third section we describe our case study sites, as well as the results of our analysis in the form of three CRM types. Section four concludes with a summary, a critical reflection of the results, and further research opportunities.[2]

2.4 Data Mining model for Trade in CRM System

Data Mining is a process of extracting previously unknown, valid, potential useful and hidden patterns from large data sets (Connolly, 1999). As the mount of data stored in CRM databases is increasing rapidly. In order to get required benefits from such large data and to find hidden relationships between variables using different data mining techniques developed and used (Han and Kamber, 2006). Clustering and decision tree are most widely used techniques for future prediction. The main goal of clustering is to partition customers into homogeneous groups according to their characteristics and abilities (Kifaya, 2009). These applications can help both advisors and customers to enhance their decision making over the products.

This study aims to analyze how different factors effect a customers preference over products using k-means in CRM System. It is a popular data mining technique that can be used to explain the interdependencies among different variables such as financial status and product preference. Clustering is one of the basic techniques often used in analyzing data sets. This study makes use of cluster analysis to segment customers into groups according to their characteristics. The remaining parts of the paper are organized as follow: one section describes the related work in CRM Data Mining. Another provides a general description of the data we used. Other describes the process stage of data used. And last section reports our experimental analysis of data mining methods applied on CRM data set.[5]

Chapter 3

Problem Statement

3.1 Drawbacks of Apriori Algorithm For Frequent Itemset

Association rules are the main technique to determine the frequent itemset in data mining. Apriori algorithm is a classical algorithm of association rule mining. From database scan transaction database is generated and that transaction database is scanned for candidate dataset at every iteration. For more number of iterations of algorithm, many such intermediate datasets will get formed .Since in our project , We are considering static but large dataset to formulate patterns , Apriori would take large computation time and give results at slower pace.

3.1.1 Typical Apriori algorithm

Apriori employs an iterative approach known as a levelwise search, where k-itemsets are used to explore $(k+1)$ -itemsets. First, the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted L_1 . Next, L_1 is used to find L_2 , the set of frequent 2-itemsets, which is used to find L_3 , and so on, until no more frequent k-itemsets can be found. The finding of each L_k requires one full scan of the database. To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property, presented is used to reduce the search space. Apriori property: All nonempty subsets of a frequent itemset must also be frequent. A two-step process is used to find the frequent itemsets: join and prune actions.

The join step

To find L_k a set of candidate k-itemsets is generated by joining L_{k-1} with itself. This set of candidates is denoted C_k .[6]

The prune step

The members of C_k may or may not be frequent, but all of the frequent k-itemsets are included in C_k . A scan of the database to determine the count of each candidate in C_k would result in the determination of L_k (i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to L_k). To reduce the size of C_k , the Apriori

property is used as follows. Any (K-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset. Hence, if any (K-1)-subset of a candidate k-itemset is not in L_{k-1}, then the candidate cannot be frequent either and so can be removed from C_k.[6]

3.1.2 Shortcomings

- Apriori algorithm generates large number of candidate sets for large database.
- Due to large number of records in database , results in much more I/O cost
- Apriori algorithm represents the candidate generation approach. It generates candidate (k+1) itemsets based on frequent k-itemsets.[9]

3.2 Modification in the algorithm

We are proposing a modified version of Apriori algorithm which reduces memory consumption caused by storing candidate database. To do so we have created a method which takes Vector as a input and that vector is only referred for computation of next iteration. Since at every step new instance of vector is created, the storage space requirement is same as memory required to store one vector. Which cancels out storing all intermediate databases of candidate sets to a greater extent. For marketing application, it is very significant for the manager to lookout for the schemes which are not creating any valuable impact. Those schemes are either discarded or boosted up. For our application purpose we are computing and displaying all such combinations which are not crossing support count threshold. [4][6][9][10]

Chapter 4

Project Description

The project focuses on combining the business aesthetics of the customer relationship module along with its actual technical implementation. It focuses on using simple methodologies and data mining algorithms to effectively extract and analyse data. The resultant sets give an in depth knowledge of customer trends, existing patterns and probable targets to be achieved. These patterns, as visually represented, will be used in contributing to company growth and further marketing strategies.

4.1 Overview of the project

The domain under consideration for all purposes related to the above mentioned project is the RETAIL BANKING SECTOR. The shortcomings of the current scenario are:

- A holistic view of CRM is absent. The perspective is either business point of view or implementation point of view.
- The change to one small aspect or component can have far reaching effects on the business on the whole. These things are often overlooked and there is a mismatch in the two.

The project aims at bifurcating the functionalities of the CRM application into clear areas of algorithms and implementation. The modules include the following four main functionalities:

- Lead capture: to identify potential customers in order to increase market reach.
- Customer tracking: to observe customer transaction patterns and trends in scheme utilisation. The bank can benefit from these associations established by targeting other customers with similar interests.
- Report generation: the above mentioned functionalities will refine and consolidate the database as per our specific needs. The resultant sets have to be then projected in an easy to comprehend manner. This is the part of report generation which aims at simplifying the visualisation of these results.
- Marketing: Though marketing is a major aspect of the CRM application, it is not the programmable part of our project. While marketing strategies will be briefly touched upon, the methodology for the same will not be coded. The project thus helps in overcoming above

mentioned problems to a great extent. The developers and users will get a complete view of all the parts of the same.

4.2 Module Description

4.2.1 Module 1: Detail extraction

In this module, We have provided the user with an easy accessibility to customer information based only on the account number as the input. For this, the UI has been made that provides a text box to accept the input from the user. This input has to be a valid account number. Once the account number has been entered, clicking on the get details button will retrieve the corresponding customers personal details as well as the schemes that he is using within the bank domain. This enables the manager or end user to get a comprehensive knowledge about each person and how valuable he can be to the bank.

4.2.2 Module 2 : Customer Tracking

Generation of Transaction Database

In the project, we have made use of the fundamentals of the Apriori algorithm in order to find frequently occurring item-sets and strong associations of existing customers product patterns. In order to apply an algorithm of this sort, we need to first have a transaction database ready. Normally Apriori algorithms take a transaction database as the input in order to create associations.

Considering a real world scenario of a banking system, it is not conventional to have a transaction database present in the system. Every product has its own data and thus creating a transaction database from these disparate tables was a crucial step in the development of this project. Also, Apriori results and the associations generated cannot be predicted. This was a major setback in terms of getting haphazard results that might be of little or no use to the user as it would prove to be a trial and error mechanism. Hence to overcome this and provide some sort of a structure to the algorithm outcome, we have provided a sort of guidance for the execution.

The UI allows the user to select minimum 1 and maximum 3 schemes that he necessarily wants to be a part of his associations that are formed. This helps in streamlining the results according to the needs of the user. Selecting these fixed tables helps the user to determine which schemes he is analysing and thereby provides him with a clear view of what the application has achieved for him. This completely removes the uncertainty and trial and error nature of the Apriori algorithm. These fixed tables now greatly influence the creation of the transaction database on which the final algorithm must be implemented to generate the associations of customer trends.

The input for this sub-module is an Excel spreadsheet with an extension of .xls. This file should contain the detailed information of all customers of the bank. A separate excel spreadsheet for separate products only increases the space consumption. Also, it would need the creation of as many number of FileInputStream and Workbook objects in java in order to carry out the further execution. Thus, we have incorporated all the information in a single Excel spreadsheet by keeping separate sheets for separate products. This single spreadsheet method helps in saving space by creating only one FileInputStream and workbook object respectively. Also the various sheets can be referenced using their particular index where the object instance remains same but

sheet referenced to changes. This helps in saving space and overhead time of the creation of these multiple objects.

Implementation of Apriori on the generated transaction database

The above sub-module generates a transaction database in the form of an Excel spreadsheet. This spreadsheet is then taken as an input to the main Apriori algorithm. The transaction database basically represents in each tuple an account number and the subsequent schemes that the customer has enrolled for with the bank. This then serves as the base for the further implementation of the algorithm. Keeping the bank system in mind, we have designed the application to give associations of maximum five schemes combination. Thus the application terminates in the case of following conditions:

- If no more combinations can be formed for new associations.
- If no combinations cross the specified support count.
- If the loop finishes iterating for 5 number of times.

In order to do so, the support count is taken as an input from the user and the corresponding value is calculated by applying the formula on the number of tuples generated in the transaction database. This value is then taken as a standard against which all the combinations formed will be compared in order to determine whether they pass or not.

In the banking system, we had to keep in mind that the failed schemes at each stage were also as important as the ones that pass. Thus a Result object was assigned with two vectors in order to store both passed and failed schemes. This assists the manager in a dual way by helping him boost weak scheme associations and gain profits by further enhancing existing strong associations.

Each iteration of the Apriori gives outputs that vary in their nature in terms of number of items in each association at each stage. Thus the incorporation of vectors has helped in keeping a check on this dynamic nature of the algorithm. We have considered these aspects in order to maintain the correctness of the implementation.

4.2.3 Module 3 : Lead capture

Algorithm used: K-means clustering.

This module basically aims at creating clusters of similar customers based on their age. Here the project adopts clustering algorithm instead of classification so as to dynamically vary the cluster centroids based on the density of the distribution taking age as the main parameter on the x axis and frequency on the y axis. The project has employed the K means algorithm in order to carry out this functionality. It provides the user flexibility in terms of selecting the number of clusters that he wants to form. Also the module enables the user to select whichever file he wants to upload in order to maintain a dynamic environment. The file uploaded only has to be an Excel spreadsheet with a .xls extension.

4.2.4 Module 4 : Report generation

The outputs of the modules are generated in an Excel file containing the subsequent details as required. Two terms of information security are extremely important in terms of the result that is generated. The confidentiality and integrity of the result should be maintained. While this application is designed for managers, the wrong use of it might result in breach of customer security and if unchecked, might lead to scams. In order to prevent this, we have converted the actual excel output into a PDF form. The final file that is displayed to the user is a PDF document. Thus the application consists of a code in built in it to convert the output excel spreadsheet to a non-editable PDF format. Also the space occupied by the PDF file is lesser than the Excel spreadsheet. This factor has been considered with keeping scalability in mind. The expansion of the data sets will generate an output file of higher size requirements.

4.2.5 Module 5 : User Interface

The user interface of the project has been developed by using the Java Swing methodology. These third party components have helped in making an effective and concise user interface. The user interface completely handles all of the users needs in a simple way. The UI has been designed keeping in mind the technical amateurism of the user. Thus all components are interactive and cater to all needs such as uploading file, and using buttons in order to create clusters, generate transactions, view results, etc.

4.2.6 Architecture

The following diagram shows the module wise break-down of the system architecture.

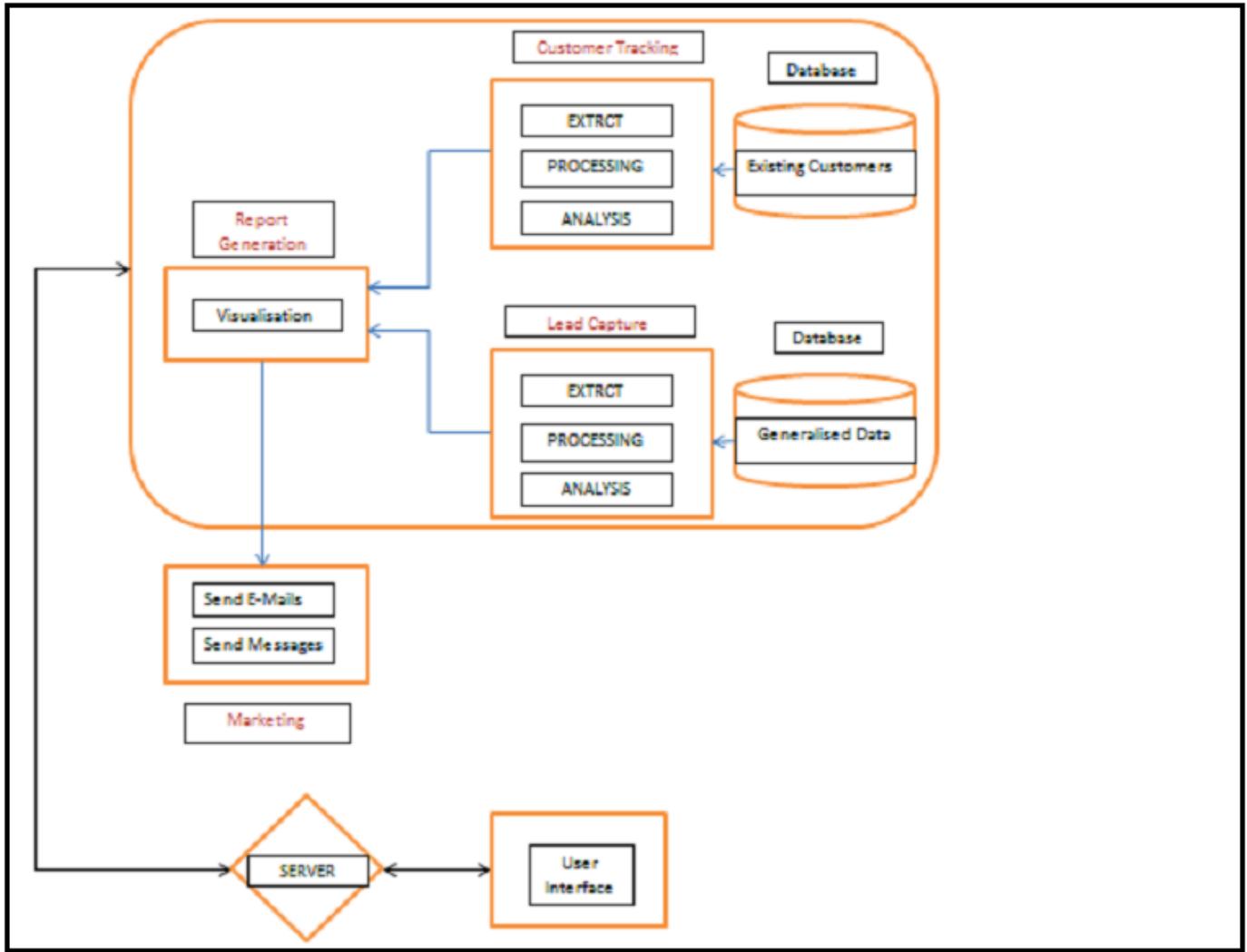


Figure 4.1: Architecture

This diagram gives brief idea about use cases that are involved in the system.

4.2.7 UML Diagram

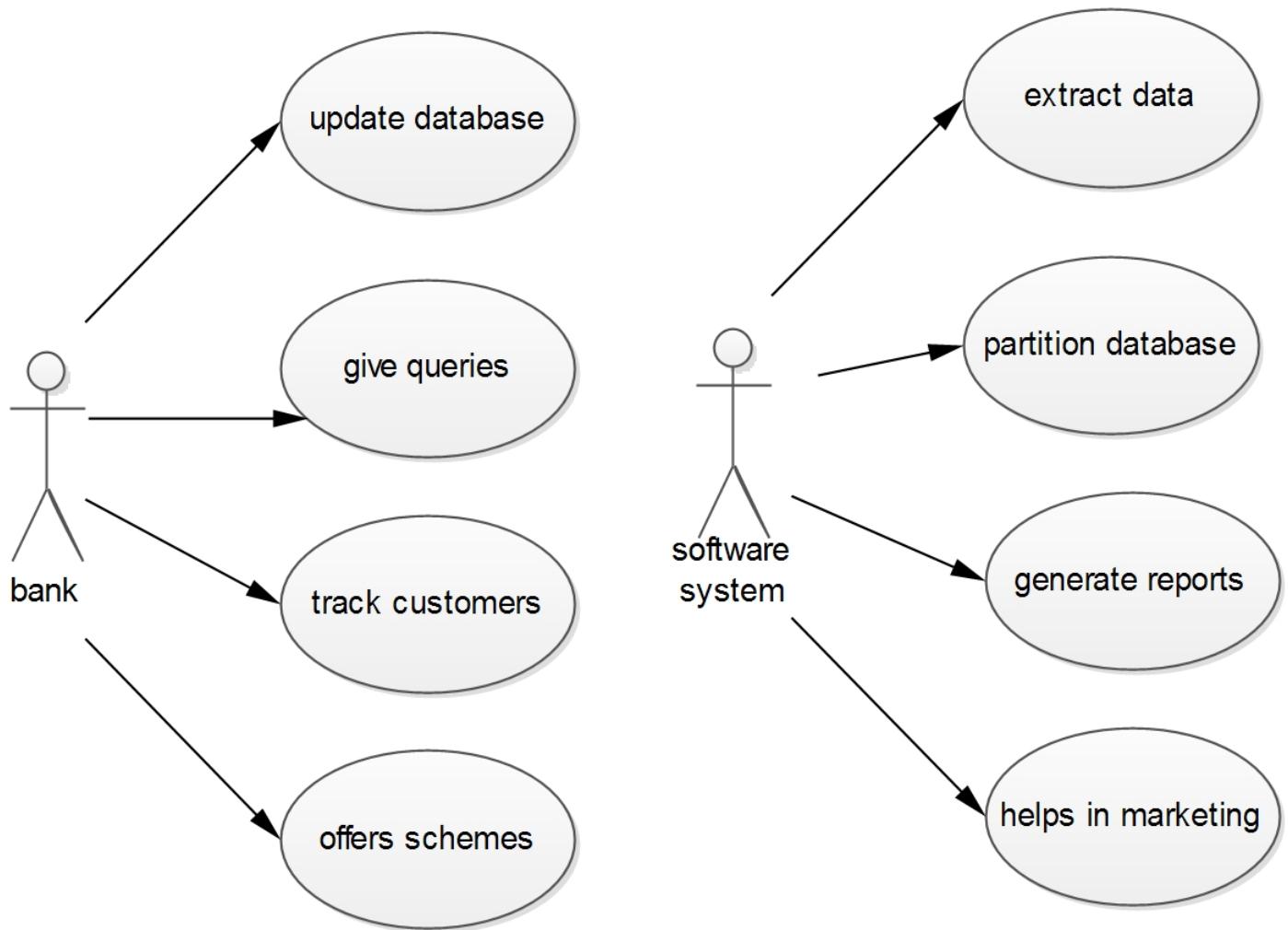


Figure 4.2: UML Diagram

Class diagram gives fair idea of classes taking part to build system infrastructure.

4.2.8 Class Diagram

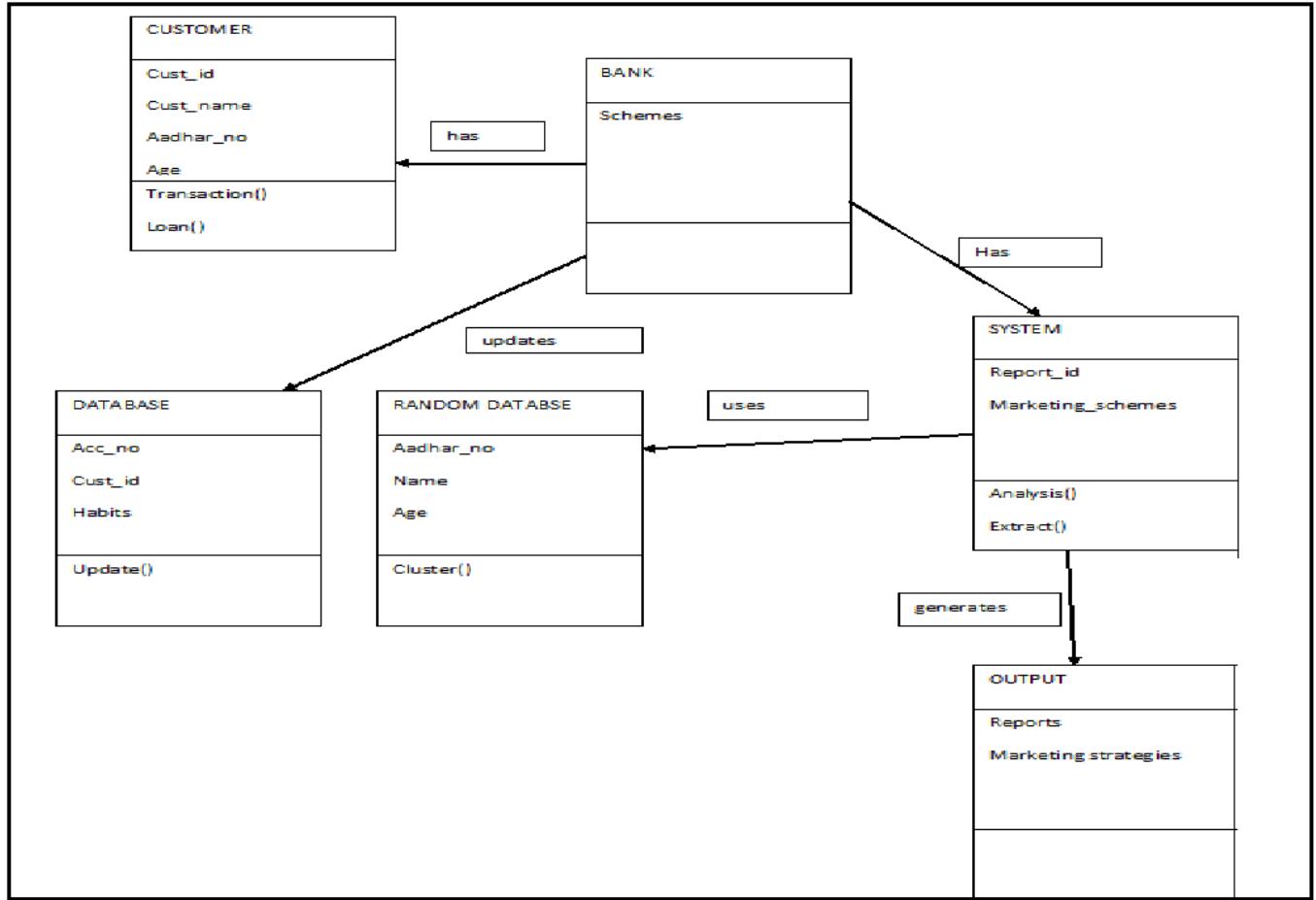


Figure 4.3: Class Diagram

Level 0 Data Flow Diagram is used to represent system with highest level of abstraction.

4.2.9 Data Flow Diagrams

Level 0

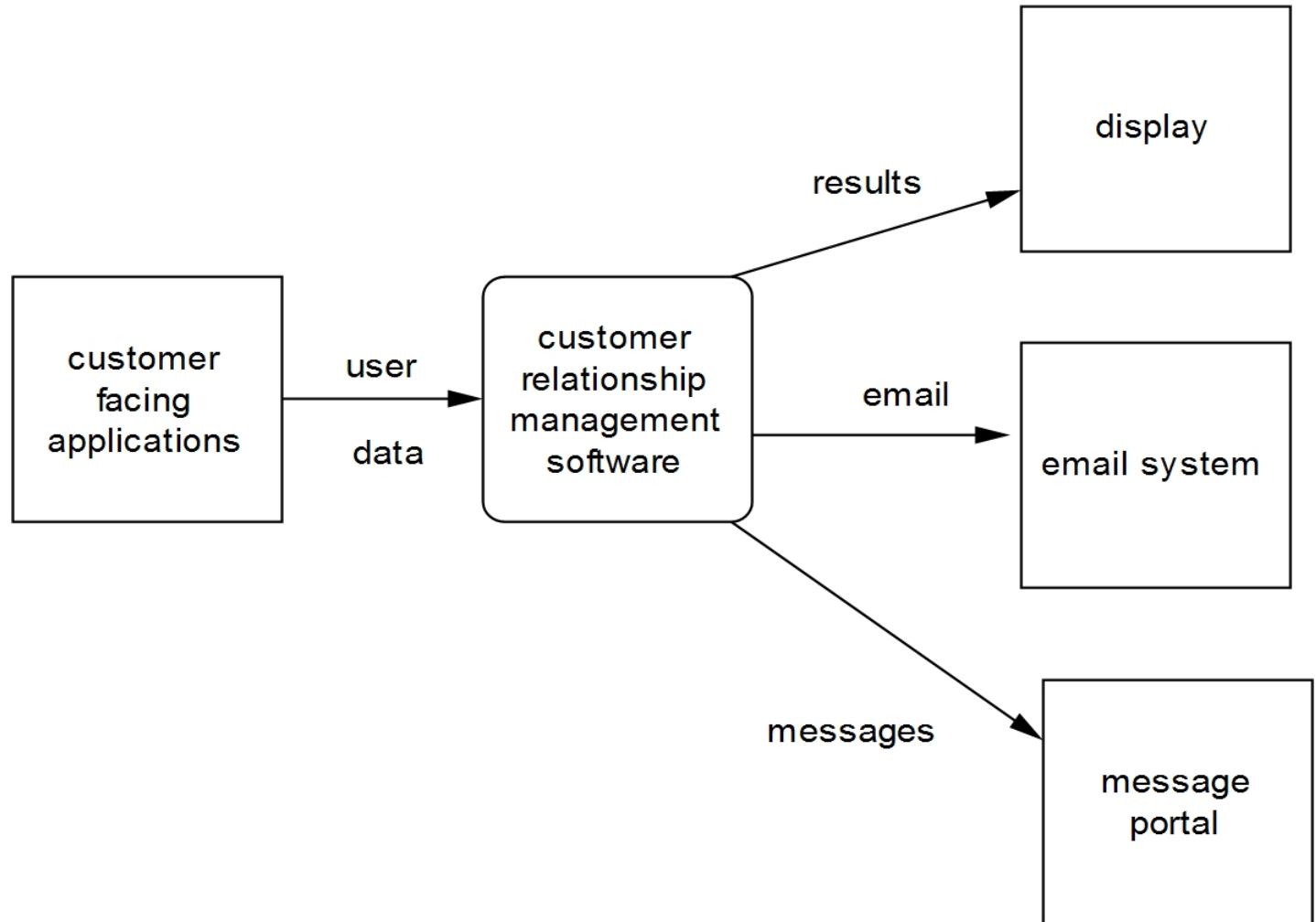


Figure 4.4: Data Flow Diagram : Level 0

Level 1 Data Flow Diagram explains the system working with depth.

Level 1

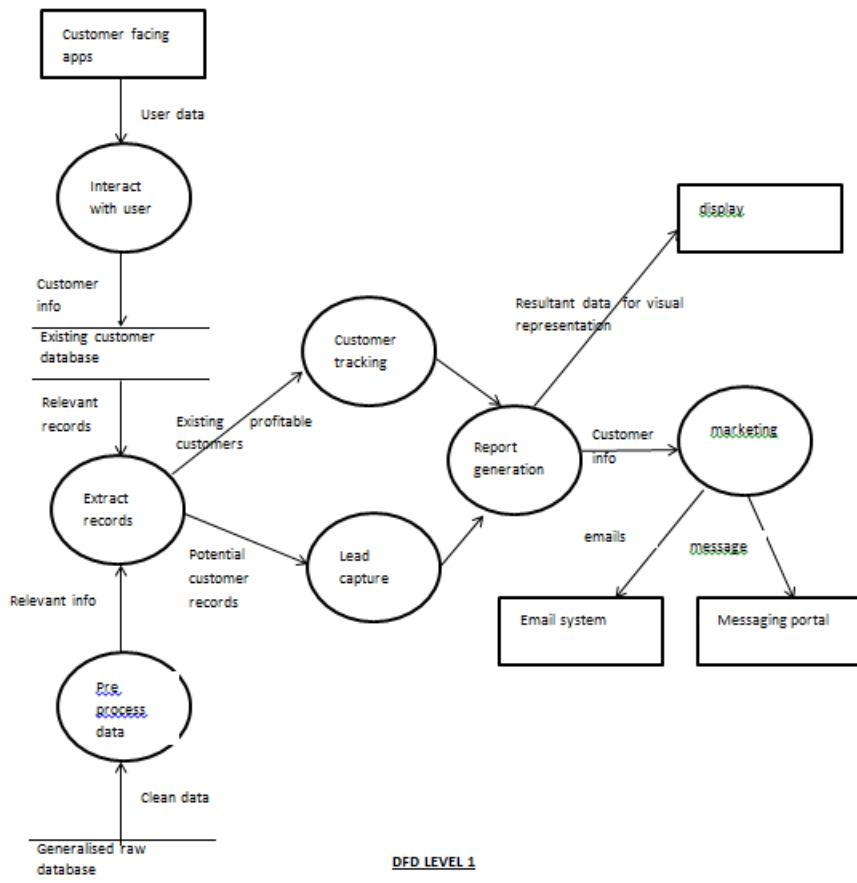


Figure 4.5: Data Flow Diagram : Level 1

Activity diagram depicts the conditions and order steps to be followed.

4.2.10 Activity Diagram

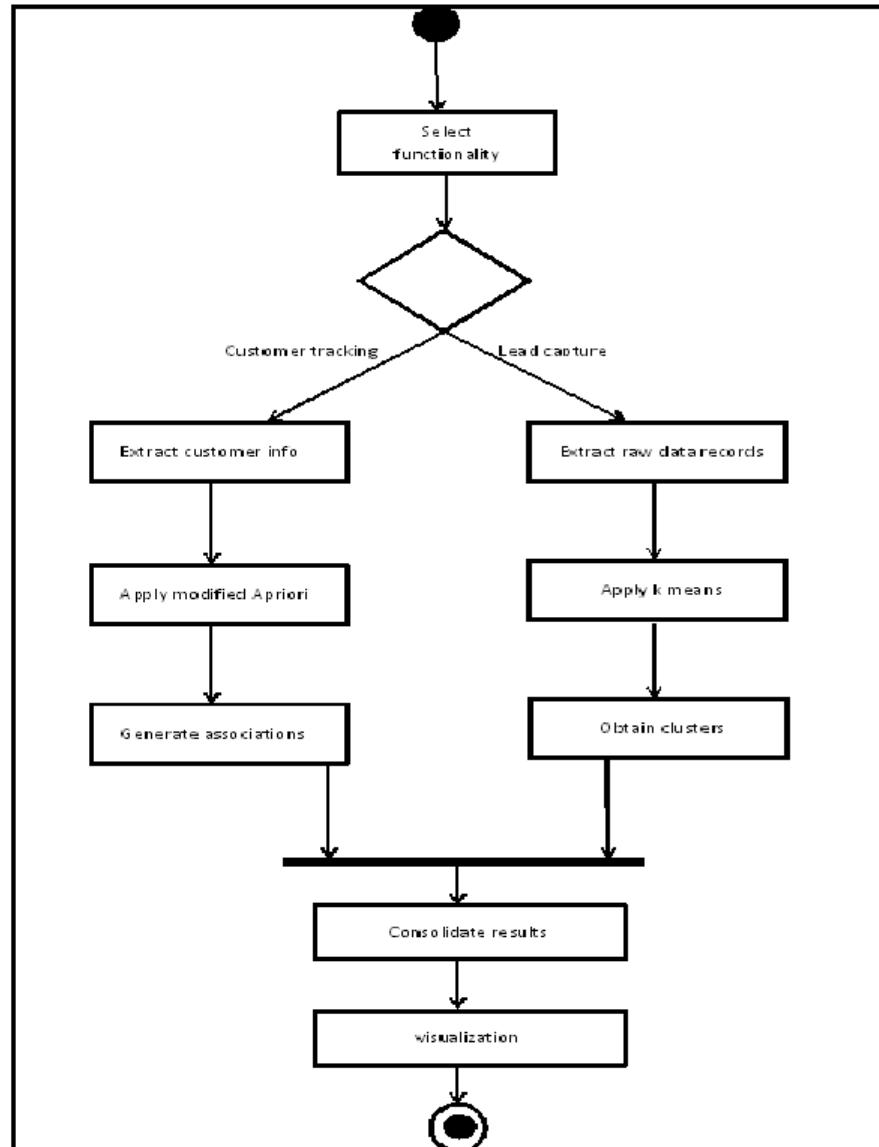


Figure 4.6: Activity Diagram

4.2.11 Database Design

Personal Details Extraction Database

Database : Detail Extraction

Table : Personal Details

Data : Holds the deatails of all the customers of bank.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
206	204 Mahend KAMDI			9217968277													
207	205 Renu Dhuwan			9091524599	F												
208	206 Anya Galle			9491735465													
209	207 Ishqia Taneja			9967499885													
210	208 Nisha Patel			9830220229	M												
211	209 Nimbahita Chaurasia			9110532178	M												
212	210 NATHI Gayatri			9108952321	M												
213	211 Ovalake Pawar			9361956578	M												
214	212 Ravi Agarwal			9106195664	F												
215	213 Diana Dossa			9106389777													
216	214 Rishabh Agarwal			9106389778													
217	215 Hirali Patel			9104130311	M												
218	216 Oh Meeketa Chawla			9112347326	M												
219	217 Deepanshu Chaudhary			9105203086	M												
220	218 Deep Gite			9126179381	F												
221	219 Om Prakash Pandey			9126247919	F												
222	220 Smita Jadhav			9645208347													
223	221 Vinita Patel			9105491020	M												
224	222 Yashika Reddy			9170041735	F												
225	223 Swapnilit Fargoe			9168840622	F												

Figure 4.7: Master Database

Customer Tracking Database design

• Database : Customer Tracking

Table : Car Loan

Data : Holds the deatails of the customers those who have taken car loan.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	CAR LOAN															
2	ACCOUNT NO.	LOAN AMOUNT	INTEREST	DURATION (MONTHS)												
3	51	200000	8.5	12												
4	311	400000	8.5	12												
5	1492	550000	8.5	24												
6	183	600000	8.5	60												
7	2060	350000	8.5	36												
8	1280	250000	8.5	48												
9	529	200000	8.5	24												
10	1512	150000	8.5	24												
11	1020	100000	8.5	12												
12	1263	350000	8.5	36												
13	2623	150000	8.5	36												
14	185	100000	8.5	48												
15	2066	400000	8.5	60												
16	2804	450000	8.5	48												
17	224	500000	8.5	24												
18	2799	650000	8.5	12												
19	640	250000	8.5	12												
20	968	350000	8.5	36												
21	846	300000	8.5	24												
22	2887	450000	8.5	60												
23	643	600000	8.5	48												

Figure 4.8: Car Loan Database

- Database : Customer Tracking

Table : Customer Details(Master Database)

Data : Holds the deatails of all the customers of bank.

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	DETAILS																
2	Customer Name		Number	Gender													
3	1 Anjali Shah	9339140559	F														
4	2 Manish Tripathi	9718725698	M														
5	3 Sunanya Deshmukh	9863454954	F														
6	4 Rahul Dua	9890269489	M														
7	5 Chinmay Patalkar	9564848078	M														
8	6 Chintan Patel	9751474342	M														
9	7 Sanya Khamna	9601080423	F														
10	8 Preeti Desai	9871496117	F														
11	9 Raj Mehta	9069791212	M														
12	10 Sonia Walia	9265073838	F														
13	11 Jay Shah	9922347370	M														
14	12 Naresh Twari	9115470147	M														
15	13 Rajesh Shukla	9252499078	M														
16	14 Sanjana Dutta	9672683244	F														
17	15 Sonal Mohnite	9648757199	F														
18	16 Rupesh Shetty	9703378366	M														
19	17 Adah Guereshi	9551690783	F														
20	18 Sonal Vakheria	9300373676	F														
21	19 Milind Patkar	9096519112	M														
22	20 Rituparna Dutta	9619209305	F														
23	21 Meenal Verma	9646813434	F														

Figure 4.9: Master Database

- Database : Customer Tracking

Table : Education Loan

Data : Holds the deatils of customers who have taken education loan.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	EDUCATION LOAN																		
2	ACCOUNT NO.	AMOUNT	INTEREST	DURATION															
3	268	241477	7.50%	1															
4	1459	758335	7.50%	8															
5	208	843577	7.50%	5															
6	1190	925224	7.50%	3															
7	355	406886	7.50%	10															
8	2163	380146	7.50%	2															
9	230	375390	7.50%	7															
10	2096	968389	7.50%	5															
11	988	785331	7.50%	4															
12	1720	942477	7.50%	3															
13	632	531195	7.50%	5															
14	1976	496348	7.50%	1															
15	277	559973	7.50%	3															
16	472	239307	7.50%	10															
17	609	828588	7.50%	10															
18	2642	869179	7.50%	4															
19	2648	857173	7.50%	1															
20	35	602682	7.50%	10															
21	484	615411	7.50%	5															
22	928	917131	7.50%	2															
23	1484	262435	7.50%	1															

Figure 4.10: Education Loan Database

- Database : Customer Tracking

Table : Fixed Deposit

Data : Holds the deatails of customers having fixed deposit in the bank.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	FD																			
2		ACCOUNT BALANCE	INTEREST RATE																	
3	386	12925240	9.5																	
4	2622	19751767	9.5																	
5	1550	23455189	9.5																	
6	1970	47990137	9.5																	
7	2588	45106700	9.5																	
8	2841	1779214	9.5																	
9	252	37796179	9.5																	
10	1101	3482631	9.5																	
11	2066	47534339	9.5																	
12	214	24248309	9.5																	
13	2698	28785504	9.5																	
14	2580	38332117	9.5																	
15	2725	4584312	9.5																	
16	2615	23310341	9.5																	
17	1676	31805932	9.5																	
18	2933	44076707	9.5																	
19	464	13576402	9.5																	
20	1403	502426	9.5																	
21	832	30638542	9.5																	
22	2904	49422476	9.5																	
23	1388	4417085	9.5																	

Figure 4.11: Fixed Deposit Database

- Database : Customer Tracking

Table : Home Loan

Data : Holds the deatails of customers who have taken home loan.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	HOME LOAN																	
2		ACCOUNT NO	LOAN AMOUNT	INTEREST	DURATION (YEARS)	INTEREST												
3	449	400000	10	16	400000													
4	1814	500000	10	12	500000													
5	475	200000	10	14	200000													
6	1049	700000	10	4	700000													
7	1107	300000	10	13	300000													
8	868	400000	10	23	400000													
9	1782	500000	10	6	500000													
10	1720	1000000	10	6	1000000													
11	1603	750000	10	22	750000													
12	1902	600000	10	11	600000													
13	2289	770000	10	16	770000													
14	541	2000000	10	18	2000000													
15	290	800000	10	23	800000													
16	219	400000	10	6	400000													
17	445	800000	10	4	800000													
18	246	900000	10	21	900000													
19	2739	3000000	10	13	3000000													
20	9	500000	10	3	500000													
21	2426	600000	10	11	600000													
22	1297	800000	10	17	800000													
23	680	300000	10	4	300000													

Figure 4.12: Home Loan Database

- Database : Customer Tracking

Table : Insurance

Data : Holds the deatails of customers who have taken insurance policy schemes of bank.

INSURANCE	ACCOUNT PLAN	SUM	PREMIUM	YEARS
3	921	155	325918	13842
4	759	239	214395	14454
5	427	138	1526559	16560
6	1101	284	338880	23328
7	68	124	1492055	17751
8	1547	284	1898355	5437
9	2585	204	572295	10052
10	1309	251	1249830	14714
11	2041	117	1316762	5672
12	272	212	1454648	21481
13	2275	299	210801	21855
14	1342	298	537976	7072
15	1465	245	293548	2253
16	1657	265	711914	21263
17	1065	249	1046890	20660
18	2391	188	1418148	17221
19	1558	271	1333109	21819
20	1599	233	746188	21033
21	97	204	1474977	23055
22	2716	256	587086	16999
23	1226	233	1569777	11588

Figure 4.13: Insurance Database

- Database : Customer Tracking

Table : Mutual Fund

Data : Holds the deatails of customers having mutual fund schemes of bank.

MUTUAL FUND	ACCOUNT	AMOUNT	PREMIUM	DURATION	INTEREST
3	2737	12917	2655	9	7.50%
4	98	73451	3881	9	7.50%
5	2868	16004	4503	9	7.50%
6	1441	99553	2921	5	7.50%
7	791	49554	3632	7	7.50%
8	2180	16471	4154	2	7.50%
9	2594	79188	3568	6	7.50%
10	561	24493	1594	1	7.50%
11	193	62729	1891	8	7.50%
12	2421	20922	3622	9	7.50%
13	2492	6093	4099	5	7.50%
14	1155	25747	4076	6	7.50%
15	2703	2569	3	7.50%	
16	1430	77884	2988	4	7.50%
17	339	14908	1568	6	7.50%
18	798	83349	1204	1	7.50%
19	1901	42598	3088	9	7.50%
20	633	56326	1243	2	7.50%
21	1788	83922	3826	3	7.50%
22	1847	14286	4621	10	7.50%
23	2805	93315	3312	7	7.50%

Figure 4.14: Mutual Fund Database

- Database : Customer Tracking

Table : Personal Loan

Data : Holds the deatails of customers who have taken personal loan from bank.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	PERSONAL LOAN																	
2	ACCOUNT	AMOUNT	INTEREST	DURATION														
3	1757	537946	10%	3														
4	1090	113188	10%	10														
5	1632	670592	10%	4														
6	163	178168	10%	2														
7	2127	715116	10%	6														
8	691	415395	10%	2														
9	1786	173270	10%	3														
10	1351	725126	10%	1														
11	712	711205	10%	6														
12	1704	877939	10%	7														
13	2343	830047	10%	9														
14	1037	250504	10%	1														
15	1868	903903	10%	6														
16	1835	596108	10%	7														
17	933	712544	10%	4														
18	2768	262121	10%	5														
19	616	943581	10%	9														
20	2093	130740	10%	1														
21	490	225951	10%	1														
22	1605	772466	10%	4														
23	2467	255212	10%	4														

Figure 4.15: Personal Loan Database

- Database : Customer Tracking

Table : Recurring Account

Data : Holds the deatails of the customers having recurring account in the bank.

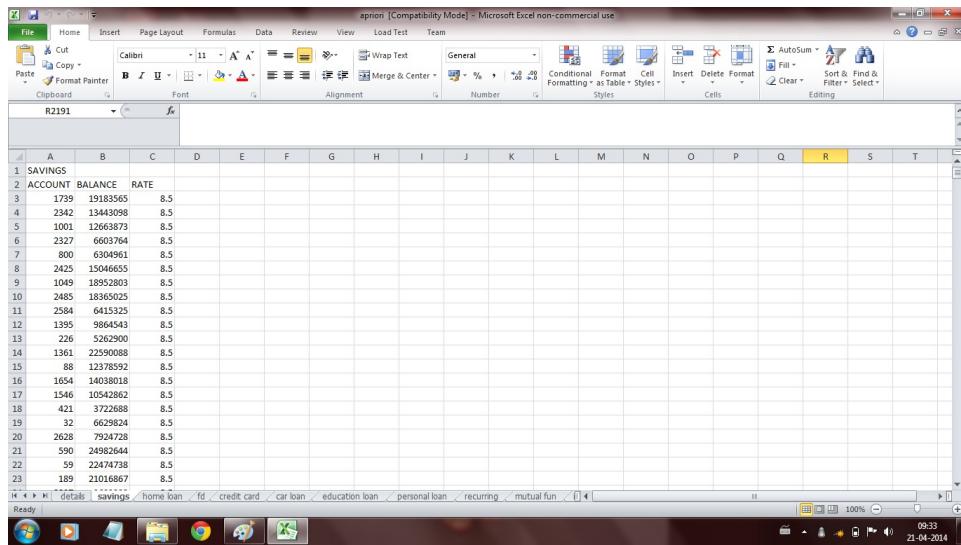
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	RECURRING																		
2	ACCOUNT NUMBER	AMOUNT	DURATION	INTEREST															
3	605	32069	7	9.30%															
4	2209	16928	1	9.30%															
5	357	23771	3	9.30%															
6	400	9164	3	9.30%															
7	2450	24188	6	9.30%															
8	2881	17534	5	9.30%															
9	2498	11824	4	9.30%															
10	1734	29266	2	9.30%															
11	1274	27680	6	9.30%															
12	450	20770	1	9.30%															
13	2248	26466	7	9.30%															
14	2619	31345	1	9.30%															
15	101	34996	4	9.30%															
16	2203	7349	2	9.30%															
17	332	5315	1	9.30%															
18	632	25069	4	9.30%															
19	13	39385	1	9.30%															
20	1067	24649	1	9.30%															
21	431	32968	5	9.30%															
22	2327	18466	5	9.30%															
23	1687	22169	7	9.30%															

Figure 4.16: Recurring Account Database

- Database : Customer Tracking

Table : Saving Account

Data : Holds the deatails of the customers having savings account in the bank.



A screenshot of Microsoft Excel showing a table titled "SAVINGS". The table has columns for ACCOUNT, BALANCE, and RATE. The data includes rows for various accounts with their respective balances and rates. The Excel interface shows the ribbon, toolbars, and a status bar indicating the date and time.

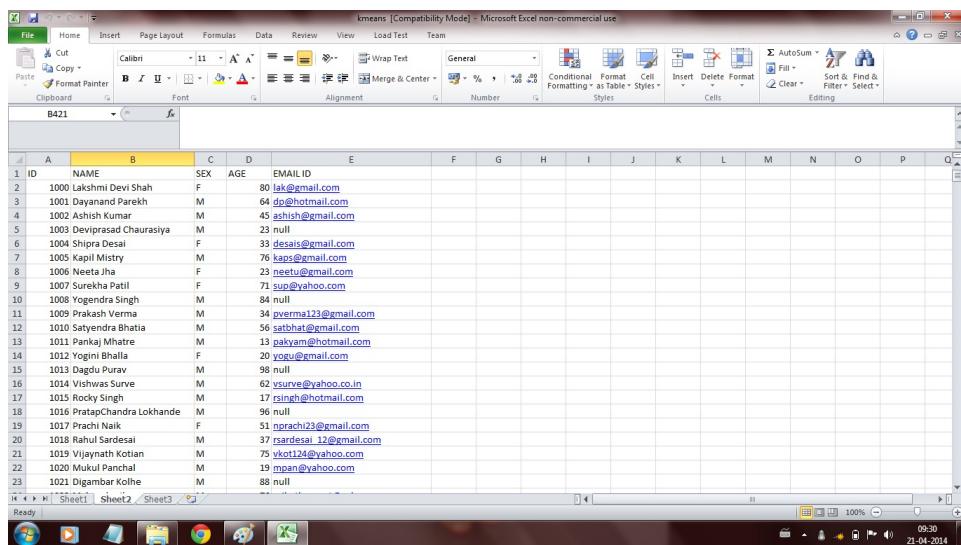
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	ACCOUNT	BALANCE	RATE																	
2																				
3	1739	19183565	8.5																	
4	2342	13443098	8.5																	
5	1001	12663873	8.5																	
6	2327	6603764	8.5																	
7	80	6304961	8.5																	
8	2425	15046655	8.5																	
9	104	18952803	8.5																	
10	248	18365025	8.5																	
11	2584	6415325	8.5																	
12	1395	9864543	8.5																	
13	226	5262900	8.5																	
14	1361	22590088	8.5																	
15	88	12378592	8.5																	
16	1654	14038018	8.5																	
17	1546	10542862	8.5																	
18	421	3722688	8.5																	
19	32	6629824	8.5																	
20	2628	7924728	8.5																	
21	590	24982644	8.5																	
22	59	22474738	8.5																	
23	189	21016867	8.5																	

Figure 4.17: Saving Account Database

Lead Capture Database design

Database : Lead Capture

Data : Holds the deatails of the people who can be potential customers of the bank.



A screenshot of Microsoft Excel showing a table titled "Leads". The table has columns for ID, NAME, SEX, AGE, and EMAIL ID. The data includes rows for various leads with their names, genders, ages, and email addresses. The Excel interface shows the ribbon, toolbars, and a status bar indicating the date and time.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q		
1	ID	NAME	SEX	AGE	EMAIL ID														
2	1009	Lakshmi Devi Shah	F	80	lakshmi@gmail.com														
3	1001	Dayanand Parekh	M	64	dps@hotmail.com														
4	1002	Ashish Kumar	M	45	ashish@gmail.com														
5	1003	Deviprasad Chaurasiya	M	23	null														
6	1004	Shipra Desai	F	33	desai@gmail.com														
7	1005	Kapil Mistri	M	76	kapil@gmail.com														
8	1006	Neeta Jha	F	23	neetu@gmail.com														
9	1007	Surekha Patil	F	76	sup@yahoo.com														
10	1008	Yogendra Singh	M	84	null														
11	1009	Prakash Verma	M	34	verma123@gmail.com														
12	1010	Satyendra Bhatia	M	56	sathat@gmail.com														
13	1011	Pankaj Mhatre	M	13	pakayam@hotmail.com														
14	1012	Yogini Bhalla	F	20	yogini@gmail.com														
15	1013	Dagdu Purav	M	98	null														
16	1014	Vishwas Surve	M	62	vsurve@yahoo.co.in														
17	1015	Rocky Singh	M	17	rinsing@hotmail.com														
18	1016	PratapChandra Lokhande	M	96	null														
19	1017	Prachi Naik	F	51	prachi23@gmail.com														
20	1018	Rahul Sardesai	M	37	rsardesai_12@gmail.com														
21	1019	Vijaynath Kotian	M	75	vkot123@yahoo.com														
22	1020	Mukul Panchal	M	19	mpan1@yahoo.com														
23	1021	Digamber Kolhe	M	88	null														

Figure 4.18: Lead Capture Database

4.2.12 Input Design

Detail extraction Input Design

Diagram depicts user interface to retrieve personal information of a customer by entering account number.

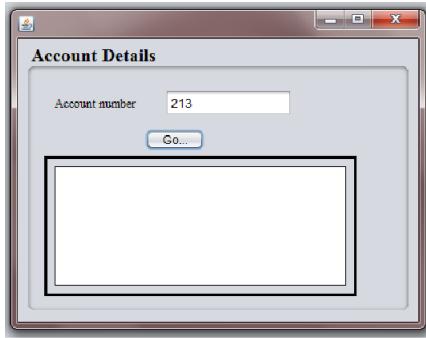


Figure 4.19: Detail Extraction User Interface - Input

Customer Tracking Input Design

Diagram depicts user interface to select at the most three schemes and support count.

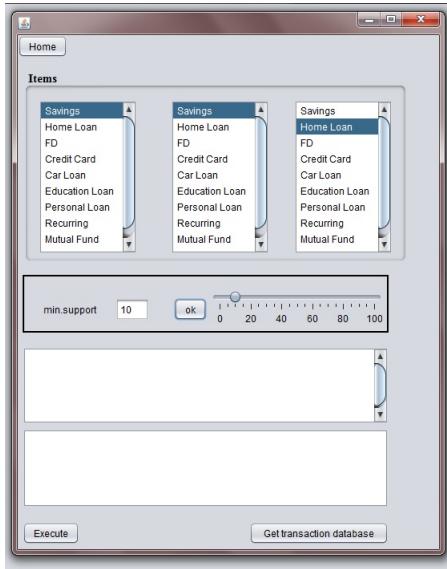


Figure 4.20: Customer Tracking User Interface - Input

Lead Capture Input Design

Diagram depicts user interface to browse and upload database as an input to clustering process.

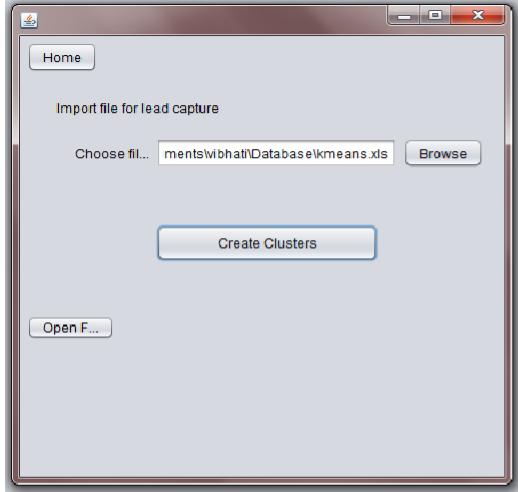


Figure 4.21: Lead Capture User Interface - Input

4.2.13 Output Design

Detail extraction Output Design

Diagram depicts user interface of retrieved personal information of a customer.

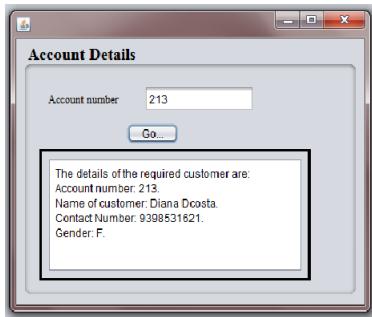


Figure 4.22: Detail Extraction User Interface - Output

Customer Tracking Output Design

- Diagram depicts user interface to notify that the process has been completed and associations have been formed.



Figure 4.23: Customer Tracking User Interface - Output1

- Diagram depicts user interface to display generated associations.

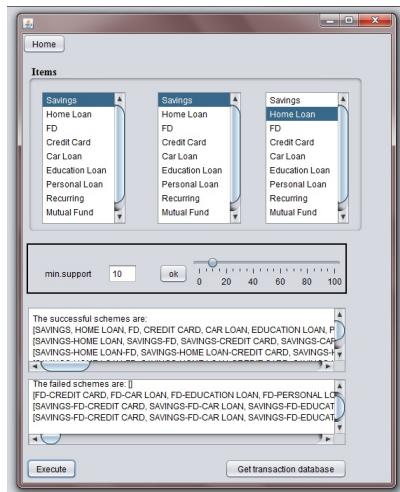
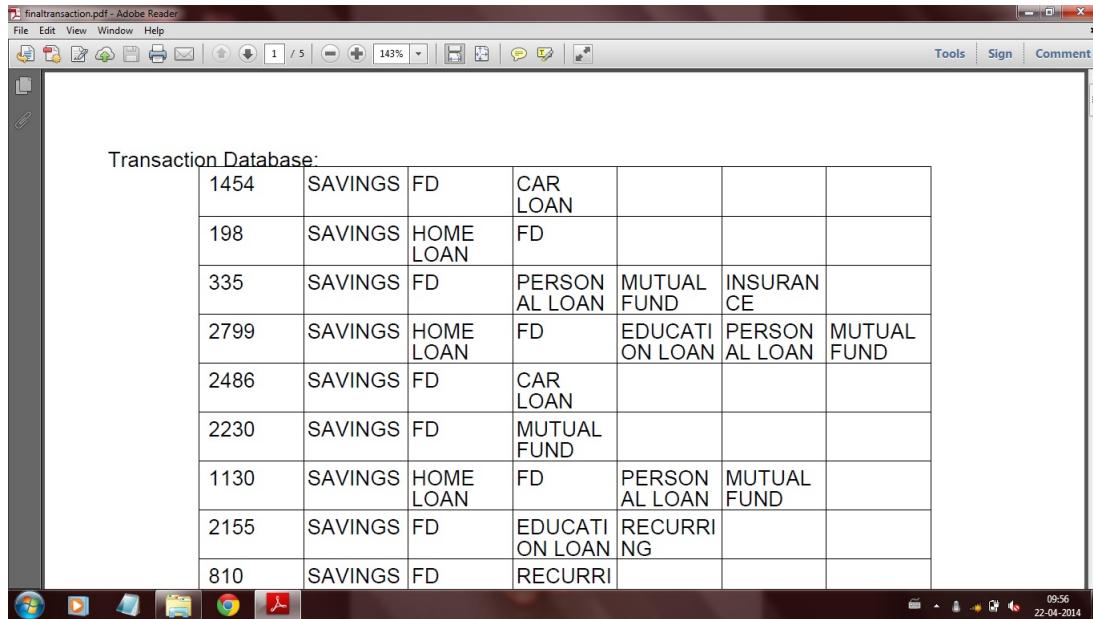


Figure 4.24: Customer Tracking User Interface - Output2

- Diagram depicts generated transaction dataset represented visually as PDF.



The screenshot shows a table titled "Transaction Database" within an Adobe Reader window. The table has 8 columns and 9 rows of data. The columns are labeled: Row ID, Type, Sub-Type 1, Sub-Type 2, Sub-Type 3, Sub-Type 4, Sub-Type 5, and Sub-Type 6. The data is as follows:

1454	SAVINGS	FD	CAR LOAN				
198	SAVINGS	HOME LOAN	FD				
335	SAVINGS	FD	PERSON AL LOAN	MUTUAL FUND	INSURAN CE		
2799	SAVINGS	HOME LOAN	FD	EDUCATI ON LOAN	PERSON AL LOAN	MUTUAL FUND	
2486	SAVINGS	FD	CAR LOAN				
2230	SAVINGS	FD	MUTUAL FUND				
1130	SAVINGS	HOME LOAN	FD	PERSON AL LOAN	MUTUAL FUND		
2155	SAVINGS	FD	EDUCATI ON LOAN	RECURRI NG			
810	SAVINGS	FD	RECURRI				

Figure 4.25: Transaction Database

Lead Capture Output Design

- Diagram depicts user interface to notify that the process has been completed and clusters have formed.



Figure 4.26: Lead Capture User Interface - Output

- Diagram depicts generated cluster no.1 represented visually as a PDF.

1000	Lakshmi Devi Shah	F	80	lak@gmail.com
1005	Kapil Mistry	M	76	kaps@gmail.com
1008	Yogendra Singh	M	84	null
1013	Dagdu Purav	M	98	null
1016	PratapChandra Lokhande	M	96	null
1019	Vijaynath Kotian	M	75	vkot124@yahoo.com
1021	Digambar Kolhe	M	88	null
1022	Mahendra Ibra	M	76	mihantkumar@gmail.com

Figure 4.27: Lead Capture Output PDF - Cluster 1

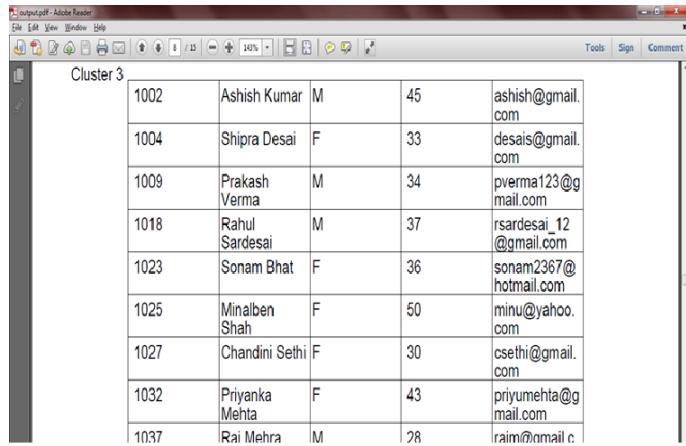
- Diagram depicts generated cluster no.2 represented visually as a PDF.

1001	Dayanand Parekh	M	64	dp@hotmail.com
1007	Surekha Patil	F	71	sup@yahoo.com
1010	Satyendra Bhatia	M	56	satbhat@gmail.com

1014	Vishwas Surve	M	62	vsurve@yahoo.co.in
1017	Prachi Naik	F	51	nprachi23@gmail.com
1026	Rita Farmica	F	65	r.farmica@gmail.com

Figure 4.28: Lead Capture Output PDF - Cluster 2

- Diagram depicts generated cluster no.3 represented visually as a PDF.

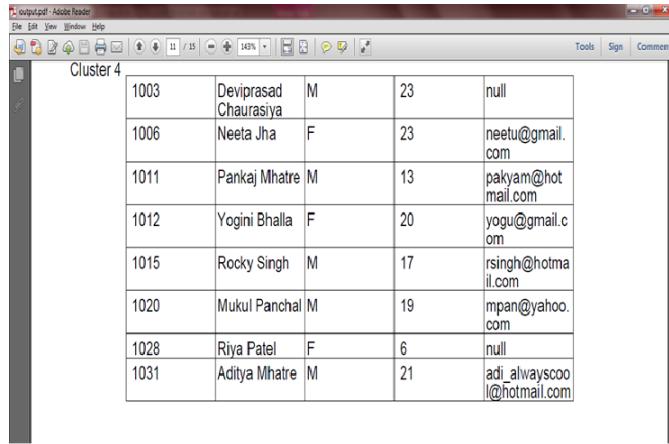


The screenshot shows a PDF document titled "Cluster 3" in Adobe Reader. The table contains the following data:

1002	Ashish Kumar	M	45	ashish@gmail.com
1004	Shipra Desai	F	33	desais@gmail.com
1009	Prakash Verma	M	34	pverma123@gmail.com
1018	Rahul Sardesai	M	37	rsardesai_12@gmail.com
1023	Sonam Bhat	F	36	sonam2367@hotmail.com
1025	Minalben Shah	F	50	minu@yahoo.com
1027	Chandini Sethi	F	30	csethi@gmail.com
1032	Priyanka Mehta	F	43	priyumehta@gmail.com
1037	Rai Mehra	M	28	raim@gmail.com

Figure 4.29: Lead Capture Output PDF - Cluster 3

- Diagram depicts generated cluster no.4 represented visually as a PDF.



The screenshot shows a PDF document titled "Cluster 4" in Adobe Reader. The table contains the following data:

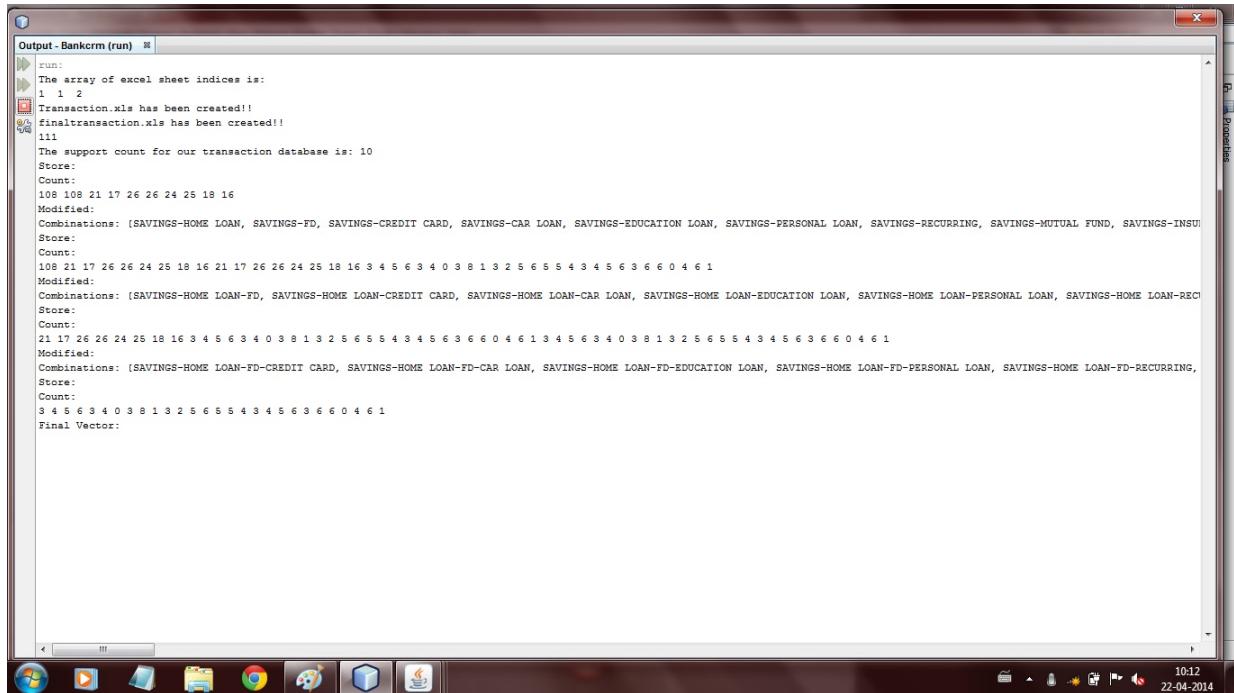
1003	Deviprasad Chaurasiya	M	23	null
1006	Neeta Jha	F	23	neetu@gmail.com
1011	Pankaj Mhatre	M	13	pakyam@hotmail.com
1012	Yogini Bhalla	F	20	yogu@gmail.com
1015	Rocky Singh	M	17	rsingh@hotmail.com
1020	Mukul Panchal	M	19	mpan@yahoo.com
1028	Riya Patel	F	6	null
1031	Aditya Mhatre	M	21	adi_alwayscool@hotmail.com

Figure 4.30: Lead Capture Output PDF - Cluster 4

4.2.14 System In Work

4.2.15 System In Work - Customer Tracking

Diagram depicts console output while algorithm is progressing.



```
Output - Bankcrm (run) ■
run:
The array of excel sheet indices is:
1 1 2
Transaction.xls has been created!!
finaltransaction.xls has been created!!
111
The support count for our transaction database is: 10
Store:
Count:
108 108 21 17 26 26 24 25 18 16
Modified:
Combinations: (SAVINGS-HOME LOAN, SAVINGS-FD, SAVINGS-CREDIT CARD, SAVINGS-CAR LOAN, SAVINGS-EDUCATION LOAN, SAVINGS-PERSONAL LOAN, SAVINGS-RECURRING, SAVINGS-MUTUAL FUND, SAVINGS-INSU
Store:
Count:
108 21 17 26 26 24 25 18 16 21 17 26 26 24 25 18 16 3 4 5 6 3 4 0 3 8 1 3 2 5 6 5 5 4 3 4 5 6 3 6 6 0 4 6 1
Modified:
Combinations: (SAVINGS-HOME LOAN-FD, SAVINGS-HOME LOAN-CREDIT CARD, SAVINGS-HOME LOAN-CAR LOAN, SAVINGS-HOME LOAN-EDUCATION LOAN, SAVINGS-HOME LOAN-PERSONAL LOAN, SAVINGS-HOME LOAN-REC
Store:
Count:
21 17 26 26 24 25 18 16 3 4 5 6 3 4 0 3 8 1 3 2 5 6 5 5 4 3 4 5 6 3 6 6 0 4 6 1
Modified:
Combinations: (SAVINGS-HOME LOAN-FD-CREDIT CARD, SAVINGS-HOME LOAN-FD-CAR LOAN, SAVINGS-HOME LOAN-FD-EDUCATION LOAN, SAVINGS-HOME LOAN-FD-PERSONAL LOAN, SAVINGS-HOME LOAN-FD-RECURRING,
Store:
Count:
3 4 5 6 3 4 0 3 8 1 3 2 5 6 5 5 4 3 4 5 6 3 6 6 0 4 6 1
Final Vector:
```

Figure 4.31: Customer Tracking - Console Output

System In Work - Lead Capture

- Diagram depicts console output while data is being divided into clusters.

```
Output - BankCRM (run) #  
run  
C:\Users\vibhuti.joshi\Documents\vibhuti\Database\kmeans.xls  
init  
Initial centroids:  
80.0 44.0 45.0 23.0  
100.0  
Lakshmi Devi Shah  
F  
80.0  
Lalit@gmail.com  
100.0  
Dayasand Parekh  
M  
66.0  
dp@hotmail.com  
100.0  
Anil Kumar  
M  
45.0  
anilish@gmail.com  
100.0  
Deviprasad Chaurasiya  
M  
29.0  
null  
100.0  
Shubhra Desai  
F  
88.0  
desais@gmail.com  
100.0  
Mangil Mistri  
M  
76.0
```

Figure 4.32: Lead Capture - Console Output 1

- Diagram depicts console output showing completion of process with final centroids.

```
Output - BankCRM (run) #  
78.0  
laxmi_desai@gmail.com  
100.0  
Alka Sonidarey  
F  
45.0  
null  
100.0  
Aneela Sangihi  
F  
87.0  
null  
100.0  
Rahani Soundip  
F  
91.0  
null  
100.0  
Zeenatika Seeks  
M  
69.0  
null  
100.0  
Cakesandmilkz Divedi  
F  
74.0  
cakesandmilkz.mrdivedi@gmail.com  
C:\Users\vibhuti.joshi\Documents\vibhuti\Database\Output\output.xls has been created  
Output file has been successfully created  
The new centroids and corresponding counts are:  
A 60 40 15  
B 88 88 96 113  
C 85 85 85 85  
D 85 85 85 85  
E 85 85 85 85  
F 85 85 85 85  
G 85 85 85 85  
H 85 85 85 85  
I 85 85 85 85  
J 85 85 85 85  
K 85 85 85 85  
L 85 85 85 85  
M 85 85 85 85  
N 85 85 85 85  
O 85 85 85 85  
P 85 85 85 85  
Q 85 85 85 85  
R 85 85 85 85  
S 85 85 85 85  
T 85 85 85 85  
U 85 85 85 85  
V 85 85 85 85  
W 85 85 85 85  
X 85 85 85 85  
Y 85 85 85 85  
Z 85 85 85 85  
(A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z) Centroids are matching!  
C:\Users\vibhuti.joshi\Documents\vibhuti\Database\Output\output.xls
```

Figure 4.33: Lead Capture - Console Output 2

4.3 Algorithms

4.3.1 Apriori Algorithm

Input: D: Database of transactions;

minsup: minimum support threshold

Output: L: frequent itemsets in D

Method:

```
1:  $L_1 = \text{find\_frequent\_1-itemsets}(D);$ 
2: for ( $k=2; L_{k-1} \neq \emptyset; k++$ ) { do
3:    $C_k = \text{apriorigen}(L_{k-1}, \text{min\_sup});$ 
4:   for each transaction  $t \in D$  do
5:      $C_t = \text{subset}(C_k, t);$ 
6:     for each candidate  $c \in C_t$ 
7:        $c.\text{count}++;$ 
8:     }
9:   end for
10:   $L_k = \{ c \in C_k \mid c.\text{count} \geq \text{min\_sup} \};$ 
11:  if ( $k \geq 2$ ) { then
12:     $\text{delete\_datavalue}(D, L_k, L_{k-1})$ 
13:     $\text{delete\_datavalue}(D, L_k, L_{k-1})\};$ 
14:  end if
15: }
16: end for
17: return  $L = \cup_k L_k ;$ 
```

Algorithm 1: Proposed Algorithm For Modified Apriori

```

1: for each itemset  $l1 \in Lk-1$  { do
2:   for each itemset  $l2 \in Lk-1$  { do
3:     if ( $l1[1] = l2[1]$ ) AND ( $l1[2] = l2[2]$ ) AND ( $l1[k-2] = l2[k-2]$ ) AND ( $l1[k-1] \leq l2[k-1]$ ) {
4:       then
5:          $c = l1 \text{ inf } l2;$ 
6:         for each itemset  $l1 \in k-1$  { do
7:           for each candidate  $c \in k$  { do
8:             if  $l1$  is the subset of  $c$  then
9:                $c.\text{num}++$  } } } } };
10:          end if
11:        end for
12:      end for
13:    end if
14:  end for
15:  $C'k = (c \in Ck \text{ OR } c.\text{num} = k)$  ;
16: return  $C'k$ ;

```

Algorithm 2: Procedure apriori_gen($Lk-1$:frequent($k-1$)-itemsets)

```

1: for each itemset  $i \in Lk-1$  and  $i \in Lk$  { do
2:   for each transaction  $t \in D$  { do
3:     for each datavalue  $\in t$  { do
4:       if datavalue =  $i$  then
5:         update datavalue = null; } } };
6:     end if
7:   end for
8: end for
9: end for

```

Algorithm 3: Procedure delete_datavalue (D:Database; Lk : frequent (k)-itemsets; $Lk-1$: frequent($k-1$) - itemsets)

```

1: for each transaction  $t \in D$  { do
2:   for each datavalue  $\in t$  { do
3:     if datavalue  $\neq$  null and datavalue  $\neq 0$  { then
4:       datarow.count++; } } };
5:     end if
6:   end for
7:   if datarow.count  $\leq k$  { then
8:     delete datarow } };
9:   end if
10: end for

```

Algorithm 4: Procedure delete_datarow (D: Database; Lk :frequent(k) - itemsets)

4.3.2 K-Means Algorithm

k: the number of clusters

Input: D: Dataset containing n objects;

Output: A set of K clusters.

Method:

- 1: *arbitrarily choose k objects from D as the initial cluster centers ;*
- 2: *repeat;*
- 3: *(re)assign each object to the cluster to which the object is the most similar,based on the mean value of the objects in the cluster;*
- 4: *update the cluster means, that is, calculate the mean value of the object for each cluster;*
- 5: *until no change;*

Algorithm 5: Algorithm For K-Means

Chapter 5

System Testing

For any project to be successfully deployed and used by the user, it needs to be as close to perfect as possible. Testing thus plays a very crucial role in ascertaining that the project developed does not have any bugs or application errors. Various testing scenarios are carried out at several levels of abstraction in order to deliver a quality product. This is test

5.1 Unit Testing

The project, from its inception itself, has been divided into modules and sections in order to facilitate development and easy connectivity using object oriented programming concepts. We have carried out unit testing of each class separately in order to ensure that each one of them was working smoothly in its own self-contained environment. Modules of user interface, lead capture, detail extraction and all sub-modules of customer tracking have been developed separately and only after ensuring their individual sanctity, they were combined together one at a time to give the entire finished code.

5.2 Integration testing

The modules mentioned above were integrated with their dependant classes in the first level of integration testing. Here, all separate modules were completely constructed and combined using their sub modules, data inputs, etc. The methodology followed in this project for that of integration testing was the bottom-up integration testing. Once these individual whole modules were tested for correctness, the project was integrated in the sense that all functionalities were taken under one project and connected with the UI so as to complete the product deliverable.

5.3 Load Testing

It is very important to guarantee that the application gives correct and concise results even if the input load on it is increased. In order to do this, a technique called load testing was adopted. Since our project connects to a backend database of Microsoft Excel, the traversing of rows was the main load on the application. To carry out load testing, small samples of data input were taken initially

like 20-30 to ensure the correctness of the code. This was further increased gradually to meet the final load capacities planned (6500 for customer tracking and 2000 for lead capture).

5.4 Black Box Testing

The black box testing refers to an outer layman testing of the project in which the output is checked for correctness with the corresponding input. This has been extensively carried out on all modules by varying input parameters such as:

- Input file
- Value of k for lead capture
- Fixed table selection for customer tracking
- Account number for detail extraction.

Thus, the project has been tested over a wide range of input values so as to confirm that the output generated by the project remains fault free and consistent.

Chapter 6

Implementation Details

6.1 Project Lifecycle

The lifecycle model we have used in this project is SPIRAL MODEL. The spiral model is a software development process combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. Also known as the spiral lifecycle model (or spiral development), it is a systems development method (SDM) used in information technology (IT). This model of development combines the features of the prototyping and the waterfall model. The spiral model combines the idea of iterative development (prototyping) with the systematic, controlled aspects of the waterfall model. It allows for incremental releases of the product, or incremental refinement through each time around the spiral. The spiral model also explicitly includes risk management within software development. Identifying major risks, both technical and managerial, and determining how to lessen the risk helps keep the software development process under control.

The spiral model is based on continuous refinement of key products for requirements definition and analysis, system and software design, and implementation (the code). At each iteration, around the cycle, the products are extensions of an earlier product. This model uses many of the same phases as the waterfall model, in essentially the same order, separated by planning, risk assessment, and the building of prototypes and simulations.

Documents are produced when they are required, and the content reflects the information necessary at that point in the process. All documents will not be created at the beginning of the process, nor all at the end (hopefully). Like the product they define, the documents are works in progress. The idea is to have a continuous stream of products produced and available for user review.

The spiral lifecycle model allows for elements of the product to be added in when they become available or known. This assures that there is no conflict with previous requirements and design. This method is consistent with approaches that have multiple software builds and releases and allows for making an orderly transition to a maintenance activity. Another positive aspect is that the spiral model forces early user involvement in the system development effort. For projects with heavy user interfacing, such as user application programs or instrument interface applications, such involvement is helpful.

- Starting at the center, each turn around the spiral goes through several task regions.

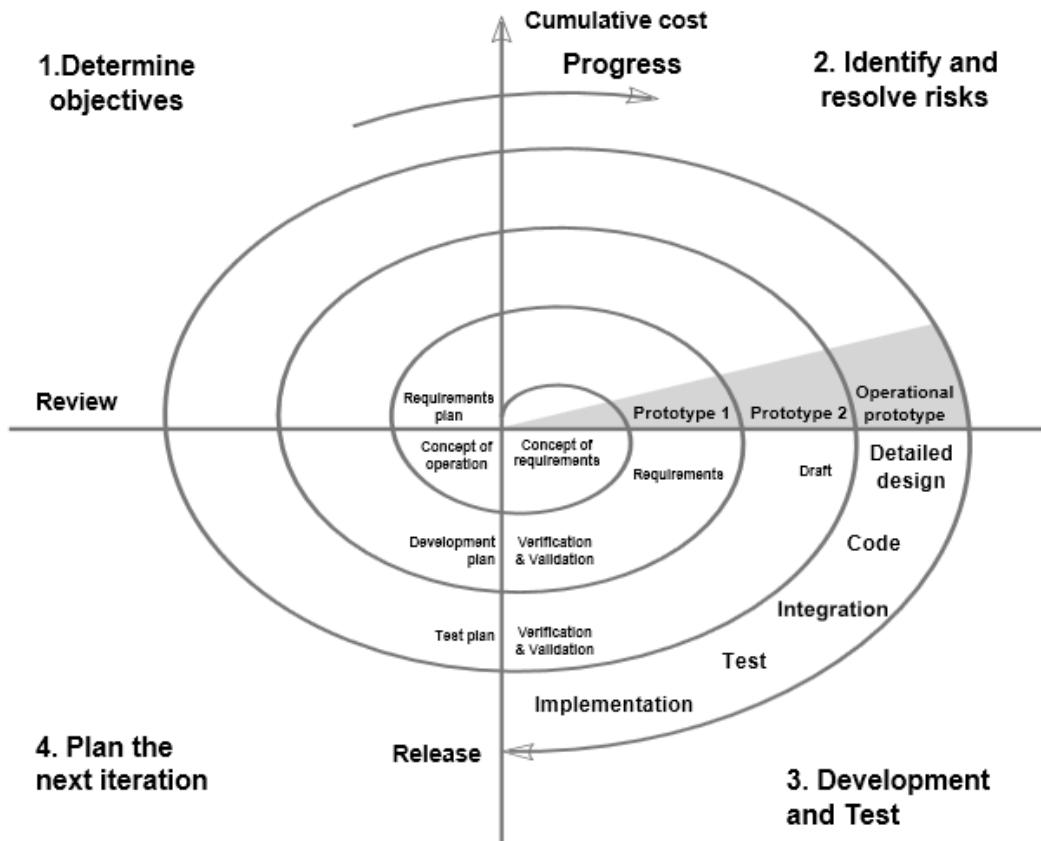


Figure 6.1: Spiral Model

- Determine the objectives, alternatives, and constraints on the new iteration.
- Evaluate alternatives and identify and resolve risk issues.
- Develop and verify the product for this iteration.
- Plan the next iteration.

The requirements activity takes place in multiple sections and in multiple iterations, just as planning and risk analysis occur in multiple places. Final design, implementation, integration, and test occur in iteration. The spiral can be repeated multiple times for multiple builds. Using this method of development, some functionality can be delivered to the user faster than the waterfall.

6.2 System Specifications

6.2.1 Minimum Hardware Requirements

- GHz 32-bit processor
- 2 GB of RAM

6.2.2 Minimum Software Requirements

- Any operating system that supports Java
- JDK version 1.6 or higher.
- Microsoft Excel 97-2003 Workbook or higher
- PDF reader
- Java Flash Player

6.3 System Description

6.3.1 Front End

Front end of the project is designed using JAVA SWING. We have used JFrames , JPannel , JButtons , List , TextArea etc. for the user interaction with the system.Swing provides a more sophisticated set of GUI components and a native look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists. We have used many of swing features to its best extent possible to develop GUI for our application.

6.3.2 Back End

Since the database requirement was huge for project, we have simulated database using Microsoft Office's Excel. The main customer database is of size 3000. Also there are 10 different tables, each having 650 tuple entries. Those tables are listed below:

- Personal Details
- Savings Account
- Education Loan
- Car Loan
- Home Loan
- Fixed Deposit
- Credit Card
- Insurance
- Mutual Fund

- Recurring Account

All these different schemes of bank are stored on the same excel file , on different sheets. The reason to store them on same file is , Apache POI Workbook make use of sheet indices to access these sheets. In order to generate transaction database from these tables, iterating through these schemes was required. In Apache POI , accessing every new excel files, needs to create new object. Hence requires more space for every object that is created for every new scheme which bank adds. Instead we have saved all schemes under one file with different sheets.

For lead capturing, we have simulated another database which has 6000 entries. For bank , it is highly important to find new customers from the near vicinity. Database comprises of people from the area under consideration which the software will be using in order to cluster them according to user requirements. Parameter on which clustering will take place has been set as 'Age'.

6.4 Methodology

In our project, we have developed our application in such a way as to make it manager-centric. Considering the end user to be a person with minimalistic technical knowledge about databases, queries, etc. we have tried to develop all aspects of the project keeping this assumption in mind. The application allows the user to browse and upload any file of his liking as input to the system. Since this is of extreme importance, to allow flexibility and freedom of choosing the file, we have used spreadsheets for the same. Some of the methodologies that we have used are as follows:

- Microsoft Excel.
- Adobe PDF.
- Object Oriented Programming.
- NetBeans 7.3.
- Java Swing.
- Apache POI Libraries.
- Lowagie Text Libraries (iText Libraries).

6.4.1 Microsoft Excel

Microsoft Excel is a spreadsheet application developed by Microsoft for Microsoft Windows and Mac OS. It includes calculation, graphing tools and pivot tables. It has been a very widely applied spreadsheet and is now accepted as the industry standard for spreadsheets. Microsoft Excel has the basic features of all spreadsheets, using a grid of cells arranged in numbered rows and letter-named columns to organise data manipulations like arithmetic operations.

The main challenge for our project from a data point of view was to create tabular information that could closely mimic real world data as possible. While manually fed data stood the risk of

being partial and repetitive, the choice of Excel helped in making the same with in built functions so as to create an environment that would provide close results to that of real-life situations.

The RANDBETWEEN function of Excel was used in order to uniformly distribute the account numbers across all schemes and banking products so as to give unbiased associations between them. This provides an in depth understanding of real-world banking scenarios, problems and probable outcomes.

6.4.2 Adobe PDF

Portable Document Format (PDF) is a file format used to present documents in a manner independent of application software, hardware, and operating systems. Each PDF file encapsulates a complete description of a fixed-layout flat document, including the text, fonts, graphics, and other information needed to display it.

The UI has been designed in order to provide the user with everything through the application itself, including the result tuples. Java supports the opening of files through UI such as text files, pdf, etc. The results generated in our program are excel based but for the ease of reading and display to the user, the results are displayed in the PDF format so as to maintain the integrity and sanctity of the results generated.

6.4.3 Object Oriented Programming Techniques:

Object oriented programming (OOP) is a programming paradigm that represents concepts as objects that have data fields and associated procedures known as methods. Objects, which are usually instances of classes, are used to interact with one another to design applications and computer programs. Since our project combines a few functionalities together, it was essential to use OOPs in order to create efficient code and separate modules. These facilitate in easily connecting the GUI with the application code and also in segregating similar functionality methods into relevant classes.

6.4.4 NetBeans 7.3

NetBeans is an integrated development environment(IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5. It is an application platform framework for Java desktop applications and others. The NetBeans IDE is written in Java and can run on Windows, OS X, Linux, Solaris and other platforms supporting a compatible JVM.

The GUI design tool enables developers to prototype and design Swing GUIs by dragging and positioning GUI components. This tool has been used extensively in the project to create an effective yet aesthetic user interface.

6.4.5 Java Swings

Swing is the primary GUI widget toolkit. It is part of Oracles Java Foundation Classes (JFC) an API for providing a graphical user interface for Java programs. Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT). Unlike AWT elements, Swing components are not implemented by platform-specific code. Instead

they are written entirely in Java and therefore are platform independent. The term lightweight is used to describe such an element.

In our project we have exclusively used Swing in order to develop a user friendly application interface that gives the user the freedom to operate the application without any detailed knowledge of any coding skills or database connectivity. We have incorporated several swing components such as lists, jButtons, jFrames, jPanels, etc. to make an easy to understand and use GUI.

6.4.6 Apache POI Libraries

Apache POI, a project run by the Apache Software Foundation, and previously a sub-project of the Jakarta Project, provides pure Java libraries for reading and writing files in Microsoft Office formats, such as Word, PowerPoint and Excel. The inclusion of Apache POI Libraries was essential for our project since they helped in the connectivity of the Java application with the Excel database.

Widely used packages of these would be HSSFRow, HSSFCCell, HSSFWorkbook, HSSFSheet. The workbook creates an instance of the workbook which is then mapped onto the Excel file that has to be taken as the input. Similarly a workbook instance is created to store all the resultant information that is later stored into an output file that is created from within the code itself. The sheet, row and cells instances, as the names suggest are used to read and manipulate the sheet, rows and cells of the input excel file.

6.4.7 iText Libraries

iText is an open source library for creating and manipulating PDF files in Java. This library can be extensively used to create, manipulate and add features to PDF files. Since Java facilitates the direct opening of PDF files through its Swing GUI components, this library has been used majorly for transforming our results from the Excel spreadsheet into a PDF file.

The importance of generating easily accessible result is very crucial in terms of serving the needs of the manager in terms of his application needs. Various functions like those of creating a new document, creating and adding tables, and assigning this document created to an output file with .pdf extension.

Chapter 7

Conclusion And Future Enhancements

7.1 Conclusion

Analysis of customer behavior enables companies to improve support of their customer oriented business processes, which aims to improve the overall performance of the enterprise. This paper also focuses on getting more customer satisfaction.

Data mining methodology has a tremendous contribution for researchers to extract the hidden knowledge and information. The research described in this paper also identified significant product association rules. Product association rules can be used to motivate customers to increase their scheme acquisition and keep loyal to the company. The behavior of customers can be easily identified.

The project has achieved the objective that it had set out to meet. It provides an insight into a very powerful business tool using IT techniques such as data mining, report generation, etc. It has integrated technology and business objectives to give a comprehensive Customer Relationship Management tool. The use of classical data mining techniques has helped achieve effective results for the user.

With the module of lead capture, we have addressed one of the key interests of the user by capturing and segregating potential customers into similar groups based on their demographical densities. This helps in dynamically creating groups and helping the customer to group unknown data as well. This will help the banking organisation to take its business forward and expand its base by tapping the huge population of this nation.

With the module of detail extraction, the user is offered a one stop access to all his needs. Often, managers need specific customer information and all they have is the bank-issued account number to go by. Thus, the application works and demands only what the user has in order to give him what he wants.

With the module of customer tracking, the project handles the very crucial task of retaining old customers and making them more profitable to the organisation by identifying user patterns, strengthening strong associations and boosting weak schemes. While such association mining needs a transaction database, we have given the freedom to inculcate existing user tables in order to create such a transaction database from which the associations can be mined and pattern analysis done.

The result sets obtained from the application are extremely essential and this information mined needs to be secure. In order to maintain the confidentiality and integrity of the project, we have added the module of report generation that generates reports in the non-editable PDF

format so as to prevent any modifications of the data generated in these results.

Overall, the project showcases and implements the importance of data mining schemes in order to take intelligence business decisions. These intelligent, well-informed decisions can be used to the best possible outcome in order to progressively take the business ahead by maximising profits and cutting the losses wherever possible.

Apriori algorithm is very helpful to find the best associations. The most frequent itemsets can be easily found out from the database.

K-means algorithm helped to generate clusters of the data points available in the database around the centroids considering age as a parameter.

7.2 Future Enhancements

The sector that the project caters to is a dynamically growing sector. Due to this reason, the project application can be further modified and improved as per changing market needs. The application currently remains focused on the technical aspects of the project majorly. The section of marketing strategies has been kept as a third part external source that is not currently being considered under the scope of the project.

In future, coding methodologies can further be applied in order to make the application a holistic one with respect to the marketing point of view as well. Currently, the application aims at providing intelligence to the user in order to arm him with knowledge that he can use in order to positively affect the growth of the organisation in terms of profits, technology and brand value.

While this knowledge forms the crux of any organisations business needs, the project can be taken a step ahead by providing auxiliary functionalities as added supplements. This will aid the user to get the best possible results and intelligent marketing strategies from the application itself with minimum human intervention.

Appendix A

SOURCE CODE

```
//for extracting customer details by giving account number

package crm;

import java.io.BufferedReader;
import java.io.File;
import java.io.InputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import org.apache.poi.hssf.usermodel.HSSFCCell;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;

public class Details {

    public static String getDetails(int acc) throws IOException, NullPointerException,
FileNotFoundException
    {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        //System.out.println("Please enter the account number!!");
        //int a = Integer.parseInt(br.readLine());

        //for loading corresponding data file in excel format.

        FileInputStream file = new FileInputStream(new File("C:\\\\Users\\\\vibhati
joshi\\\\Documents\\\\NetBeansProjects\\\\Apriori\\\\apriori.xls"));

        //creating new workbook to assign the file
```

```

HSSFWorkbook detail = new HSSFWorkbook(file);

//get sheet in which data is present

HSSFSheet detailsheet = detail.getSheetAt(0);

String set = null;

for(int i =2;i<2938;i++)

{

    HSSFCell detailcell = detailsheet.getRow(i).getCell(0);

    if((int)detailcell.getNumericCellValue()== acc)

    {

        set = "The details of the required customer are: \n";

        //System.out.println("The details of the requested customer are : ");

        for(int j = 0;j<4;j++)

        {

            detailcell = detailsheet.getRow(i).getCell(j);

            switch (j)

            {

                case 0:

                    int no = (int)detailcell.getNumericCellValue();

                    //System.out.println("Account number : " + no);

                    set = (set + "Account number: " + no + ".\n");

                    break;

                case 1:

                    String name = detailcell.getStringCellValue();

                    //System.out.println("Name of customer : " + name);

                    set = (set + "Name of customer: " + name + ".\n");

            }

        }

    }

}


```

```
        break;

case 2:
    long number =(long)detailcell.getNumericCellValue();
    //System.out.println("Contact Number : " + number);
    set = (set + "Contact Number: " + number + ".\n");
    break;

case 3:
    String gender = detailcell.getStringCellValue();
    //System.out.println("Gender : " + gender);
    set = (set + "Gender: " + gender + ".\n");
    break;

}

}

}

}

return set;
}

}



---




---


//to execute k means algorithm to create clusters
package CRM;
import java.io.File;
```

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.hssf.usermodel.HSSFRow;

public class K {
    public static String kexecute(String s) throws FileNotFoundException, IOException,
    NullPointerException {
        String s1;
        System.out.println("hi!!!");
        //try
        //{
            //Reading Excel file
            FileInputStream file = new FileInputStream(new File(s));
            HSSFWorkbook wbin = new HSSFWorkbook(file);
            HSSFSheet sheet = wbin.getSheetAt(1);

            //Creating new excel workbook
            HSSFWorkbook wbout = new HSSFWorkbook();

            //Creating new sheet for the output data
            HSSFSheet sheetout = wbout.createSheet("Sheet Cluster");
    }
}
```

```
//sample extraction

HSSFCell cell = sheet.getRow(1).getCell(3);

double age = cell.getNumericCellValue();

System.out.println("First age is " + age);

//HSSFCell cell = null;

//extract value of k from UI

int k = 4 ;

double c[] = new double [k];

//for assigning initial centroid values

System.out.println("Initial centroids: ");

for(int i = 1;i<=k;i++)

{

    cell = sheet.getRow(i).getCell(3);

    c[i-1] = cell.getNumericCellValue();

    System.out.print(c[i-1] + " ");

}

//for storing differences

double d[] = new double [k]; //for storing differences

int w=0;

do{

    int num = 0;

    System.out.println();

    System.out.println("Here now!!");

    //JUMBO for loop
```

```

for(int i=(1);i<=400;i++)
{
    HSSFRow row = sheetout.createRow(num);
    int l=0;
    int col =0;
    //for determining differences
    for(int j =0;j<k;j++)
    {
        cell = sheet.getRow(i).getCell(3);
        d[j]=Math.abs(cell.getNumericCellValue()-c[j]);
    }
    double min = d[0];
    for(l=0;l<d.length;l++)
    {
        if (d[l]<min)
        {
            min = d[l];
            col = l;
        }
    }
    //extracted data has to be stored in corresponding cluster
    int start = 0;
    start = 5*col;
    int st = start;
}

```

```
System.out.println("Start: " + start);

for(int m = 0;m<=4;m++)

{

    cell = sheet.getRow(i).getCell(m);

    //double s = cell.getNumericCellValue();

    //System.out.println("Value cell: " + s);

    //checking type of cell data

    switch (cell.getCellType())

    {

        case Cell.CELL_TYPE_NUMERIC:

            double no = cell.getNumericCellValue();

            System.out.println(no);

            row.createCell(start).setCellValue(no);

            start = start+1;

            break;

        case Cell.CELL_TYPE_STRING:

            String name = cell.getStringCellValue();

            System.out.println(name);

            row.createCell(start).setCellValue(name);

            start = start +1;

            break;

    }

}
```

```

        }

        //for filling buffer values in order to calculate new mean

        for(int z = 3;z<(k*5);z=z+5)

        {

            if(z!=st && z!=st+1 && z!=st+2 && z!=st+3 && z!=st+4)

                row.createCell(z).setCellValue(0);

        }

        num = num+1;

    }

    s1 = "C:\\\\Users\\\\vibhati joshi\\\\Documents\\\\vibhati\\\\Database\\\\Output\\\\output.xls";

    System.out.println(s1+ "in kexecute");

    FileOutputStream fileOut = new FileOutputStream("C:\\\\Users\\\\vibhati\n
joshi\\\\Documents\\\\vibhati\\\\Database\\\\Output\\\\output.xls");

    wbout.write(fileOut);

    fileOut.close();

    System.out.println("Output file has been successfully created");

    //computing new centroids

    int counter[] = new int[k];

    int cnew[] = new int[k];

    System.out.println("The new centroids and corresponding counters are: " );//to see new
centroids

    int p = 0;

```

```
for(int y = 3; y<k*5 ; y=y+5)

{

    for(int x = 0; x<400;x++)

    {

        cell = sheetout.getRow(x).getCell(y);

        if(cell.getNumericCellValue()!=0)

            counter[p]++;

        cnew[p] += (int)cell.getNumericCellValue();

    }

    cnew[p] = cnew[p]/counter[p];

    p++;

}

for(int o:cnew)

    System.out.print(o+"\t");

    System.out.println();

for(int o:counter)

    System.out.print(o+"\t");



for(int u = 0;u<k;u++)

{

    if(cnew[u]==c[u])

    {

        w++;

    }

}
```

```
        continue;

    }

    else

        c[u]=cnew[u];

    }

    System.out.println();

    System.out.println("Updated Centroids: ");

    for(int q=0;q<k;q++)

    {

        System.out.print(c[q] + " ");

    }

    /*if(w!=3)

    {

        System.out.println();

        System.out.println("inside if");

    }

*/



}while(w!=k);

System.out.println("Centroids are matching!!");



return s1;

}

}
```

```
//main module for execution of apriori algorithm for generation of associations

package crm;

import java.io.*;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Vector;
import org.apache.poi.hssf.usermodel.HSSFCCell;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.hssf.usermodel.HSSFRow;

public class AprioriExecute {

    public static void doApriori(int a1[], int x)throws IOException, FileNotFoundException,
    NullPointerException {
        /*
        System.out.println("Enter the number of fixed tables!!");

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int n = Integer.parseInt(br.readLine());

        System.out.println("Select the sheet numbers from below menu!!");
        System.out.println("1. Savings!");
        System.out.println("2. Home Loan!");

    }
}
```

```
System.out.println("3. FD!");
System.out.println("4. Credit Card!");
System.out.println("5. Car Loan!");
*/
//array for storing sheet indices
int a[] = new int[a1.length];
a = a1;

/* for(int i=0;i<n;i++)
{
    a[i] = Integer.parseInt(br.readLine());
} */

System.out.println("The array of excel sheet indices is: ");
a = sort(a);
for(int i:a)
    System.out.print(i+" ");
System.out.println();

// System.out.println("Enter the non fixed indices!!");
int b[] = new int[10];
for(int i=0;i<b.length;i++)
{
    //b[i]= Integer.parseInt(br.readLine());
    b[i] = (i+1);
}
```

```

//call method to get intersection table of fixed sheets

Transaction.intersect(a);

//System.out.println("Program end!!");

//for reading table apriori

FileInputStream apfile = new FileInputStream(new File("C:\\\\Users\\\\vibhati
joshi\\\\Documents\\\\vibhati\\\\Database\\\\apriori.xls"));

HSSFWorkbook apin = new HSSFWorkbook(apfile);

HSSFSheet apsheet = null;

//to acces intersection file for further transactional file

FileInputStream apfile1 = new FileInputStream(new File("C:\\\\Users\\\\vibhati
joshi\\\\Documents\\\\vibhati\\\\Database\\\\Transaction\\\\transaction.xls"));

HSSFWorkbook apin1 = new HSSFWorkbook(apfile1);

HSSFSheet apsheet1 = apin1.getSheetAt(0);

//for final transaction table in excel

HSSFWorkbook apout2 = new HSSFWorkbook();

HSSFSheet apsheet2 = apout2.createSheet("Transaction");

int z=2;

for(int i=2;i<apsheet1.getLastRowNum();i++)

{

    int column =0;

    HSSFCell apcell1 = apsheet1.getRow(i).getCell(0);

    int compare = (int)apcell1.getNumericCellValue();

```

```
HSSFRow aprow2 = apsheet2.createRow(z);

aprow2.createCell(0).setCellValue(compare);

column++;

for(int j=0;j<b.length;j++)

{

    if(Transaction.search(compare,b[j]))

    {

        apsheet = apin.getSheetAt(b[j]);

        HSSFCell apcell = apsheet.getRow(0).getCell(0);

        String table = apcell.getStringCellValue();

        /*switch (table)

        {

            case "SAVINGS":


                column = 1;

                break;





            case "HOME LOAN":


                column = 2;

                break;





            case "FD":


                column = 3;

                break;





            case "CREDIT CARD":


                column = 4;

                break;

        }

    }

}
```

```
        break;

    case "CAR LOAN":
        column = 5;
        break;

    case "EDUCATION LOAN":
        column = 6;
        break;

    case "PERSONAL LOAN":
        column = 7;
        break;

    case "RECURRING":
        column = 8;
        break;

    case "MUTUAL FUND":
        column = 9;
        break;

    case "INSURANCE":
        column = 10;
        break;
    }*/  
aprow2.createCell(column).setCellValue(table);
```

```

        column++;

    }

}

z++;

}

//Creation of final transaction database file in excel.

FileOutputStream apfileOut = new FileOutputStream("C:\\\\Users\\\\vibhati
joshi\\\\Documents\\\\vibhati\\\\Database\\\\Transaction\\\\finaltransaction.xls");

apout2.write(apfileOut);

apfileOut.close();

System.out.println("finaltransaction.xls has been created!!");

//String transaction = "C:\\\\Users\\\\vibhati
joshi\\\\Documents\\\\NetBeansProjects\\\\Apriori\\\\Transaction\\\\finaltransaction.xls";



//Calculating support count for the algorithm further.

int last = apsheet1.getLastRowNum();

System.out.println(last);

int supp = Parameters.support(last-2,x);

System.out.println("The support count for our transaction database is: " + supp);

//to get frequencies

ResultApriori r = new ResultApriori();

ResultApriori rfinal = new ResultApriori();

//r = Execute.count(supp);

//r.display();

```

```

Vector v = new Vector();

for(int i=0;i<10;i++)
{
    apsheet = apin.getSheetAt(i+1);

    HSSFCell apcell = apsheet.getRow(0).getCell(0);
    v.add(i, apcell.getStringCellValue());
}

for(int i = 0; i<5;i++)
{
    rfinal = r;
    r = Execute.check(v, supp);

    if(r.pass.size()!=0)
    {
        System.out.println("Modified: ");
        r.display();
    }
    else
    {
        r = rfinal;
        System.out.println("Final Vector: ");
        r.display();
        break;
    }
}

//new support count
//supp = Parameters.support(r.pass.size());
//System.out.println("Revised support: " + supp);

```

```
v = Execute.create(r,i);

if(v.size()!=0)

{

    System.out.println("Combinations: " + v);

}

else break;
```

```
}
```

```
}
```

```
public static int[] sort(int s[])
```

```
{

    for(int i = 0;i<s.length;i++)

    {

        for(int j=i+1;j<s.length;j++)

        {

            if(s[i]>s[j])

            {

                int temp=s[j];

                s[j]=s[i];

                s[i]=temp;

            }

        }

    }

}
```

```
    return s;
```

```
}
```

```
}
```

```
//class Result for creation of result object to store pass and failed associations at every stage
```

```
package crm;
```

```
import java.util.Vector;
```

```
public class ResultApriori {
```

```
    Vector pass;
```

```
    Vector fail;
```

```
    ResultApriori()
```

```
{
```

```
    pass = new Vector();
```

```
    fail = new Vector();
```

```
}
```

```
    public void display()
```

```
{
```

```
        apriori.jTextArea1.append(pass + "\n");
```

```
        apriori.jTextArea2.append(fail + "\n");
```

```
}
```

```
}
```

```
//calculating support count
```

```
package crm;
```

```
import java.io.BufferedReader;
```

```
import java.io.IOException;
```

```
import java.io.InputStreamReader;
```

```
public class Parameters {
```

```
    public static int support(int l, int x) throws IOException
```

```
{
```

```
    /*
```

```
        System.out.println("Enter the support % for association rules!!");
```

```
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
```

```
        int supp = Integer.parseInt(br.readLine());*/
```

```
    int supp;
```

```
    //System.out.println(supp);
```

```
    supp = (int)((x * (l)) / 100);
```

```
    return supp;
```

```
}
```

```
//creating intermediate intersection and final transaction database
```

```
package crm;
```

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import com.lowagie.text.Document;
import com.lowagie.text.DocumentException;
import com.lowagie.text.Paragraph;
import com.lowagie.text.Phrase;
import com.lowagie.text.pdf.PdfPCell;
import com.lowagie.text.pdf.PdfPTable;
import com.lowagie.text.pdf.PdfWriter;
import java.io.FileOutputStream;
import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;

public class Conversion {

    public static String kconvert(String s) throws IOException, FileNotFoundException, DocumentException, NullPointerException {
        FileInputStream file = new FileInputStream(new File(s));
        String s1 = s.replaceAll(".xls", ".pdf");
        int index = s.indexOf(".xls");
    }
}
```

```
System.out.println("index : "+ index);

System.out.println("in conversion" + s1);

HSSFWorkbook wbin = new HSSFWorkbook(file);

//wbin.save("abc.pdf", SaveFormat.PDF);

HSSFSheet sheet = wbin.getSheetAt(0);

HSSFCell cell;

HSSFRow row;

Document doc = new Document();

PdfWriter.getInstance(doc, new FileOutputStream(s1));

doc.open();

System.out.println("pdf run");

//PdfPTable table = new PdfPTable(5);

PdfPCell tablecell;

int clusters =4;

for(int i =0; i<clusters;i++)

{

    int st = 5*i;

    doc.add(new Paragraph("Cluster " + (i+1) + "\n"));

    PdfPTable table = new PdfPTable(5);

there:for(int j = 0;j<sheet.getLastRowNum();j++)

{
```

```

for(int k=st;k<(st+5);k++)

{

cell = sheet.getRow(j).getCell(k,Row.RETURN_BLANK_AS_NULL);

if(cell==null)

    continue there;

else

{

switch(cell.getCellType())

{



case Cell.CELL_TYPE_NUMERIC:

    int num = (int)cell.getNumericCellValue();

    String set = (" " + num + " ");

    tablecell = new PdfPCell(new Phrase(set));

    table.addCell(tablecell);

    break;




case Cell.CELL_TYPE_STRING:

    tablecell = new PdfPCell(new Phrase(cell.getStringCellValue()));

    table.addCell(tablecell);

    break;

}

}

}

}

```

```
    doc.add(table);

}

doc.close();
file.close();

return s1;
}

public static String aconvert(String transaction) throws IOException, FileNotFoundException, DocumentException, NullPointerException
{
    FileInputStream file = new FileInputStream(new File(transaction));
    String pdf = transaction.replaceAll(".xls", ".pdf");

    HSSFWorkbook wbin = new HSSFWorkbook(file);
    HSSFSheet sheet = wbin.getSheetAt(0);
    HSSFCell cell;
    HSSFRow row;

    Document doc = new Document();
    PdfWriter.getInstance(doc, new FileOutputStream(pdf));
    doc.open();
    PdfPTable table = new PdfPTable(7);
    PdfPCell tablecell;
    doc.add(new Paragraph("Transaction Database: \n"));
}
```

```
there: for(int i=2;i<sheet.getLastRowNum();i++)

{

//PdfPTable table = new PdfPTable(6);

here: for(int j=0;j<7;j++)

{

cell = sheet.getRow(i).getCell(j,Row.RETURN_BLANK_AS_NULL);

if (cell==null)

{

table.addCell("");

continue here;

}

else

{



switch(cell.getCellType())

{



case Cell.CELL_TYPE_NUMERIC:

int num = (int)cell.getNumericCellValue();

String set = (" " + num + " ");

tablecell = new PdfPCell(new Phrase(set));

table.addCell(tablecell);

break;

}
```

```
        case Cell.CELL_TYPE_STRING:  
  
            tablecell = new PdfPCell(new Phrase(cell.getStringCellValue()));  
            table.addCell(tablecell);  
            break;  
  
        /* case Cell.CELL_TYPE_BLANK:  
            table.addCell("");  
            break; */  
    }  
}  
  
}  
  
//doc.add(table);  
}  
  
doc.add(table);  
doc.close();  
file.close();  
  
return pdf;  
}  
}  


---



---



```
//creating combinations and checking associations for support count at each stage
```


```

```
package CRM;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Vector;
import org.apache.poi.hssf.usermodel.HSSFCCell;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;

public class Execute {

    public static Vector create(ResultApriori rcreate, int z)
    {
        Vector v1 = new Vector();
        int count;

        if(z==0)
        {
            for(int i=0;i<rcreate.pass.size();i++)
            {
                for(int j=i+1;j<rcreate.pass.size();j++)
                {
                    v1.addElement(rcreate.pass.elementAt(i).toString() + "-" +
rcreate.pass.elementAt(j).toString());
                }
            }
        }
    }
}
```

```
    }

}

}

else

{

for(int i = 0;i<rcreate.pass.size();i++)

{



String temp1 = rcreate.pass.get(i).toString();

String a1[] = temp1.split("-");



for(int j = i+1; j<rcreate.pass.size();j++)

{



count = 0;

String temp2 = rcreate.pass.get(j).toString();

String a2[] = temp2.split("-");



for(int k=0; k<z;k++)

{



if(a1[k].equals(a2[k]))



{



count++;

continue;





}

else

break;

}
```

```
        }

        if(count==z)

        {

v1.addElement(rcreate.pass.get(i).toString() + " - " + a2[z]);


        }

        }

    }

}

return v1;

}

public static ResultApriori check(Vector v, int s) throws IOException, FileNotFoundException

{

    ResultApriori r = new ResultApriori();

    FileInputStream apfileOut = new FileInputStream(new File("C:\\\\Users\\\\vibhati\njoshi\\\\Documents\\\\vibhati\\\\Database\\\\Transaction\\\\finaltransaction.xls"));

    HSSFWorkbook apout2 = new HSSFWorkbook(apfileOut);

    HSSFSheet apsheet2 = apout2.getSheetAt(0);

    HSSFRow aprow2 = null;

    HSSFCell apcell2 = null;

    String store[];
```

```
String temp;

int counter[] = new int[v.size()];

System.out.println("Store: ");

for(int i=0;i<v.size();i++)

{

    temp = v.get(i).toString();

    store = temp.split("-");

    counter[i]=Transaction.search(store);

}

System.out.println("Count: ");

for(int i=0;i<counter.length;i++)

{

    System.out.print(counter[i] + " ");

    if(counter[i]>=s)

    {

        r.pass.addElement(v.get(i));

    }

    else

    {

        r.fail.addElement(v.get(i));

    }

}

System.out.println();

return r;
```

```
}
```

```
}
```

```
//converting output excel file to PDF
```

```
package CRM;
```

```
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import org.apache.poi.hssf.usermodel.HSSFSheet;  
import org.apache.poi.hssf.usermodel.HSSFWorkbook;  
import com.lowagie.text.Document;  
import com.lowagie.text.DocumentException;  
import com.lowagie.text.Paragraph;  
import com.lowagie.text.Phrase;  
import com.lowagie.text.pdf.PdfPCell;  
import com.lowagie.text.pdf.PdfPTable;  
import com.lowagie.text.pdf.PdfWriter;  
import java.io.FileOutputStream;  
import org.apache.poi.hssf.usermodel.HSSFCell;  
import org.apache.poi.hssf.usermodel.HSSFRow;  
import org.apache.poi.ss.usermodel.Cell;  
import org.apache.poi.ss.usermodel.Row;
```

```
public class Conversion {  
  
    public static String kconvert(String s) throws IOException, FileNotFoundException,  
    DocumentException, NullPointerException  
{  
    FileInputStream file = new FileInputStream(new File(s));  
  
    String s1 = s.replaceAll(".xls", ".pdf");  
  
    int index = s.indexOf(".xls");  
  
    System.out.println("index : " + index);  
  
    System.out.println("in conversion" + s1);  
  
    HSSFWorkbook wbin = new HSSFWorkbook(file);  
  
    //wbin.save("abc.pdf", SaveFormat.PDF);  
  
    HSSFSheet sheet = wbin.getSheetAt(0);  
  
    HSSFCell cell;  
  
    HSSFRow row;  
  
    Document doc = new Document();  
  
    PdfWriter.getInstance(doc, new FileOutputStream(s1));  
  
    doc.open();  
  
    System.out.println("pdf run");  
  
    //PdfPTable table = new PdfPTable(5);  
  
    PdfPCell tablecell;  
  
    int clusters = 4;  
  
    for(int i = 0; i < clusters; i++)  
    {  
        int st = 5 * i;
```

```
doc.add(new Paragraph("Cluster " + (i+1) + "\n"));

PdfPTable table = new PdfPTable(5);

there:for(int j = 0;j<sheet.getLastRowNum();j++)

{

    for(int k=st;k<(st+5);k++)

    {

        cell = sheet.getRow(j).getCell(k,Row.RETURN_BLANK_AS_NULL);

        if(cell==null)

            continue there;

        else

        {

            switch(cell.getCellType())

            {

                case Cell.CELL_TYPE_NUMERIC:

                    int num = (int)cell.getNumericCellValue();

                    String set = (" " + num + " ");

                    tablecell = new PdfPCell(new Phrase(set));

                    table.addCell(tablecell);

                    break;

                case Cell.CELL_TYPE_STRING:

                    tablecell = new PdfPCell(new Phrase(cell.getStringCellValue()));


            }
        }
    }
}
```

```
        table.addCell(tablecell);

        break;

    }

}

}

}

doc.add(table);

}

doc.close();

file.close();

return s1;

}

public static String aconvert(String transaction) throws IOException, FileNotFoundException, DocumentException, NullPointerException

{

    FileInputStream file = new FileInputStream(new File(transaction));

    String pdf = transaction.replaceAll(".xls", ".pdf");

    HSSFWorkbook wbin = new HSSFWorkbook(file);

    HSSFSheet sheet = wbin.getSheetAt(0);

    HSSFCell cell;

    HSSFRow row;
```

```
Document doc = new Document();

PdfWriter.getInstance(doc, new FileOutputStream(pdf));

doc.open();

PdfPTable table = new PdfPTable(7);

PdfPCell tablecell;

doc.add(new Paragraph("Transaction Database: \n"));

there: for(int i=2;i<sheet.getLastRowNum();i++)

{



//PdfPTable table = new PdfPTable(6);

here: for(int j=0;j<7;j++)

{

cell = sheet.getRow(i).getCell(j,Row.RETURN_BLANK_AS_NULL);

if (cell==null)

{

table.addCell("");

continue here;

}

else

{
```

```
switch(cell.getCellType())
{
    case Cell.CELL_TYPE_NUMERIC:
        int num = (int)cell.getNumericCellValue();
        String set = (" " + num + " ");
        tablecell = new PdfPCell(new Phrase(set));
        table.addCell(tablecell);
        break;

    case Cell.CELL_TYPE_STRING:
        tablecell = new PdfPCell(new Phrase(cell.getStringCellValue()));
        table.addCell(tablecell);
        break;

    /* case Cell.CELL_TYPE_BLANK:
        table.addCell("");
        break; */

}

//doc.add(table);
}
```

```
    doc.add(table);

    doc.close();

    file.close();

    return pdf;
}

}
```

Bibliography

- [1] M. W. W. M. Sonja Grabner-Kraeuter, Gernot Moedritscher, “Performance monitoring of crm initiatives,” in *in Proceedings: IEEE conference on System Sciences*, 2007.
- [2] F. Buttle, “Customer relationship management: Concepts and technologies,” *Oxford, Elsevier Butterworth Heinemann, ISBN 978-1- 85617-522-7*, 2009.
- [3] E. Janny.C.Hoekstra, “The lifetime value concept in customer-based marketing, journal of market focused management [m],” in *proc. 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR)*, vol. 3, pp. 257–274, 1999.
- [4] R. Agrawal, “Fast algorithms for mining association rules in large databases,” vol. OE-12, pp. 207–216, 1993.
- [5] L. S.-g. L. J. C. Han-joon, Kim TaeHee, “Automated data warehousing for rule-based crm systems,” *14th Australasian Database Conference (ADC-2003)*, 2003.
- [6] W. Y. X. W. E. W. B. Chen, “The research of improved apriori algorithm for mining association rules,” in *Communication Technology, 2008. ICCT 2008 11th IEEE International Conference on*, pp. 513–516, Nov 2008.
- [7] O. Z. Ismail, R. and Bakar, “Associative prediction model and clustering for product forecast data,” in *IEEE Intelligent Systems Design and Applications(ISDA),10th International Conference*, 2010.
- [8] M. R. G. Seyed Mohammad Seyed Hosseini, Anahita Maleki, “Cluster analysis using data mining approach to develop crm methodology to assess the customer loyalty,” in *Expert Systems with Applications*, vol. 37, pp. 5259–5264, 2010.
- [9] Z. Y. Xu and C. Zhang, “An optimized apriori algorithm for mining association rules,” vol. 29(19), pp. 83–84, 2003.
- [10] W. J. Luo Ke, “Apriori algorithm based on the improved [j],” in *Computer Engineering and application*, vol. 20, pp. 20–22, 2001.