

**VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF
TECHNOLOGY**
Department of Computer Engineering



Project Report on

**CLOUD-BASED CORE BANKING SOFTWARE
FOR SMALL FINANCIAL INSTITUTIONS**

In partial fulfillment of the Fourth Year, Bachelor of Engineering (B.E.) Degree in
Computer Engineering at the University of Mumbai Academic Year 2017-2018

Submitted by

Manjiri Bhat (D17 - A , Roll no -7)
Romit Bonkar (D17 - A , Roll no -10)
Monal Daswani (D17 - A , Roll no -21)
Narayan Mundhara (D17 - A , Roll no -51)

Project Mentor
Mrs. Anjali Yeole

(2017-18)

**VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF
TECHNOLOGY**
Department of Computer Engineering



Certificate

This is to certify that **Manjiri Bhat, Romit Bonkar, Monal Daswani, Narayan Mundhara** of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on "**Cloud Based Core Banking Software for Small Financial Institutions**" as a part of their coursework of PROJECT-II for Semester-VIII under the guidance of their mentor **Prof.Anjali Yeole** in the year 2017-2018 .

This thesis/dissertation/project report entitled (**Title**) by (**Author Name**) is approved for the degree of _____ (**Degree details**).

Programme Outcomes	Grade
PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date:

Project Guide:

This page should have the certificate of industrial project (If any)

Project Report Approval

For

B. E (Computer Engineering)

This thesis/dissertation/project report entitled (*Title*) by (*Author Name*) is approved for the degree of _____ (*Degree details*).

Internal Examiner

External Examiner

Head of the Department

Principal

Date:
Place:

Declaration

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

(Signature)

Manjiri Bhat - 07

(Name of student and Roll No.)

(Signature)

Monal Daswani - 21

(Name of student and Roll No.)

(Signature)

Romit Bonkar - 10

(Name of student and Roll No.)

(Signature)

Narayan Mundhara - 51

(Name of student and Roll No.)

Date:

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Assistant Professor **Mrs. Anjali Yeole** (Project Guide) for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We would also like to thank **Mr. Nilesh Bane** for giving us the idea and guidance for the implementation of the project.

We are deeply indebted to Head of the Computer Department **Dr. (Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair**, for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement at several times.

Computer Engineering Department
COURSE OUTCOMES FOR B.E PROJECT

Learners will be to,

Course Outcome	Description of the Course Outcome
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solution for the problem.
CO 4	Able to interpret the data and datasets to be utilized.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one's competency.
CO 11	Able to develop professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

Abstract

Credit societies are small institutions that work like banks for a restricted range of users. These credit societies, commonly known as ‘Patpedhi’, are existing for quite some time and they need a standard, common platform for maintaining their database and transactions. The objective is to create a website for a cloud based system for the credit societies so that they can have a centralized core banking application which can connect them with other small financial institutions. The project is developed with respect to agile methodology with continuous user inputs and changing requirements. The website is based on a multi-user multi-level hierarchy and is implemented using Laravel; currently the best framework; and PostgreSQL. The multi-level user hierarchy provides data abstraction and access control. The new modules for Credit society registration, user registration and customer application developed in the project will be integrated with the existing transaction and general functional modules and deployed on cloud for a fully working system. The system will be monthly billing and usage based. The software will provide high level security, mobility and will be affordable.

Table of Contents

CHAPTER 1: INTRODUCTION	11
1.1 Introduction	11
1.2 Motivation	11
1.3 Problem Definition	12
1.4 Relevance of the project	12
1.5 Methodology employed for development	12
CHAPTER 2: LITERATURE SURVEY	13
2.1 Literature survey and its various sources	13
2.2 Books / Articles referred / news paper referred	18
CHAPTER 3: REQUIREMENT GATHERING	20
3.1 Functional Requirement	20
3.2 Non Functional Requirements	20
3.3 Constraints	21
3.4 Hardware, Software , Technology and tools - Requirements	21
3.4.1 Hardware Requirements:	21
3.4.2 Software Requirements:	21
3.4.3 Techniques and Tools	22
3.5 Selection of the Hardware, Software , Technology and tools	22
CHAPTER 4: PROPOSED DESIGN	23
4.1 System design/Conceptual Design	23
4.2 Block diagram of the system	24
4.3 Modular diagram of the system	25
4.4 Detailed Design (DFD - level 0,1,2, State Transition Diagram, ER Diagram, etc...)	27
4.4.a ER Diagram	27
4.4.b Flowchart	29
4.4.c Data Flow Diagram	30
4.4.d Project Scheduling & Tracking using Timeline / Gantt Chart:	33
CHAPTER 5: IMPLEMENTATION DETAILS	34
5.1. Evaluation of the developed system	34
CHAPTER 6: TESTING	35
6.1 SQL Injection :	35
6.2 Broken Access	36
6.3 Unauthorized User Access	37
6.4 XSS Attack	38
6.5 Validations:	39
CHAPTER 7: RESULT ANALYSIS	42
7.1 Parameters considered for development	42
7.2 Screenshots of User Interface (UI)	42

CHAPTER 8: CONCLUSION	51
8.1 Limitations	51
8.2 Conclusion	51
8.3 Future Scope	51
References	52
Appendix	54
6.1 Table of figures:	54
6.2 Papers Published	57
Paper 1 (semester 7)	57
Paper 2 (Semester 8)	60

CHAPTER 1: INTRODUCTION

1.1 Introduction

As the world is moving towards digitization, a plethora of web-based solutions in the different aspects of life are emerging. Retail, finance, entertainment, etc. are all going digital. With digitization, comes the need for development of a platform for these services. Particularly in banking sector, a complex platform for the intricacies of the banking world is necessary. A new trend of using cloud-computing in banking sector to reduce development costs is emerging.

Different solutions in the form of PaaS (platform as a service), SaaS (Software as a service) and IaaS (Infrastructure as a service) which are different models of cloud-based technology are being developed.

Credit societies are basically small institutes established by a group of people which work like banks for a limited number of people. The number of members can range from 20 to 2000. Still it is a small number to actually call it a bank but larger than a single account. Lending money, buying shares in the societies, forming groups to save money, depositing money for safekeeping are some of the characteristics of a Credit society, also called as “Patpedhi”. These credit societies also need a digital platform for maintaining their databases and transactions. We aim to fulfil this need.

This application is based on the concept of ‘**software as a service**’, in which software and associated data are centrally hosted on the cloud(a remotely located server).One of the biggest constraints of small financial institutes to implement automated banking services is high investment involved in software development. One of the biggest selling point of ‘Cloud-based credit banking software’ is that it has the potential to reduce IT support costs by outsourcing hardware and software maintenance and support to the SaaS provider. Provision for network-based access to , and management of , banking software.It is based on a single-instance, multi-tenant(one to many) architecture including pricing,partnering and management characteristics.

1.2 Motivation

The motive for this project is to help small credit societies who are geographically dispersed.The objective of this project is to create a cloud for these credit societies so that they can have a centralized core banking application which can connect them with other small financial institutes. Currently, the software which are available are expensive and small institutes are not able to afford such high investments. This software will offer an affordable alternative with additional features of high level security, low installation costs, mobility etc. to such institutes.

1.3 Problem Definition

Credit societies which are small financial institutes can rarely bear the high investment cost and risk of project failure associated with it. Even if they are able to afford that cost, it keeps them isolated from other credit societies as there is no centralized core banking application which can connect them with other small financial institute. This project is an attempt to remove such drawback of system and to create a cloud for credit societies. We need to develop new modules which can be integrated with some of the existing modules to form the fully functional system.

1.4 Relevance of the project

The Final system will be affordable by all small financial institutions with low upfront costs and low installation costs. The user need not be well-educated or qualified, he just needs to have a brief introduction to the system. The users are granted access and permissions to different facilities depending on the user hierarchy. The system will act as a core banking application.

1.5 Methodology employed for development

Agile method is used for project development. In agile methodology there is a continuous development and simultaneous testing process going on in iterations in the software development life cycle. The user requirements were met from meeting to meeting and additional requirements were determined in each meeting. The meetings were conducted every month to determine the acceptability and changes in the project.

This is a cloud based system. It refers to applications, services or resources made available to users on demand via the Internet from provider's servers. cloud-based computing is used as a way to increase capacity, enhance functionality or add additional services on demand without having to commit to potentially expensive infrastructure costs or increase / train existing in-house support staff.

CHAPTER 2: LITERATURE SURVEY

2.1 Literature survey and its various sources

1. Cloud Computing for Banking and Investment Services [6]

Abstract:

“The more the things change the more they stay the same.” That can be true in some aspects of life, but not in the banking industry. At a global level, the scale of change in this industry since 2008 has been unprecedented. Control is now in the hands of the customer, rather than the bank. Satisfying customer demands has become more complex as they are looking for convenience and control over their banking services. . Technology has become a costly hurdle for financial institutions and developing customized solutions or investing in advanced banking platforms has either been unfeasible or they resulted to too many failures. In today’s business and economic climate, financial institutions are looking out for better solutions to cope up with their customer needs as well as get a competitive edge over other banks. Cloud computing is touted as a way to deliver software in the future, can help banking sector react to this new customer-driven environment with innovation in business models, operations and IT. It can offer unconventional business models for delivering innovative client experiences, better data security, high service quality, enhanced IT efficiency and reduced operating cost. Cloud computing is a type of disruptive innovation that is likely to change the demographics of banking industry. Using cloud computing, banks can create a flexible and agile banking environment that can quickly respond to new business needs. Though cloud banking can face few problems like security, data confidentiality and quality of services, they can be sorted by building a good application model. This paper aims to give an overview of cloud computing and proposes a business model to eliminate the negative perception related to cloud based technologies. It also elaborates the defects of current banking systems, and explores the innovative applications of cloud computing in banks.

Inference:

Banking softwares using cloud computing are a good option to reduce initial costs required for creating individual systems for each institute. Its ability of pay as you go i.e. billing according to usage makes cloud banking a desirable option. Cloud computing provides shared resources and a common infrastructure which can be accessed anytime and from anywhere. Current adoption of cloud computing in banking industry is mostly using private clouds for individual institutions. Mobility and scalability are the advantages whereas security , data segregation and privileged user access are some of the challenges in cloud banking industry.

2. Cloud Computing in Banking Services^[7]

Abstract:

The banking industry is facing several changes. Control is now in the hands of the customer, rather than the bank. Customers are driving new business models. Technology changes the traditional business transformation. Banks need to react to this new customer-driven environment with innovation in business models, operations and IT. For banks, the value proposition for cloud computing affects the entire business. Cloud technology offers a new model for delivering innovative client experiences, effective collaboration, improved speed to market and increased IT efficiency. Cloud computing provides a platform for optimizing financial services operations while creating and delivering the kind of innovative services that differentiate and propel your business forward. It is agility that will be the lifeblood of successful financial enterprises going forward, and cloud computing is one way of gaining that agility. Cloud services deliver revolutionary performance that empower the banking industry to automate and manage their processes.

Inference:

More people are moving to web-based services for instant access to their accounts for transactions. Cloud-based banking provides a competitive advantage to institutes over other parallel non-cloud banking and finance institutes. Data recovery in case of disaster is easier. Cloud can be implemented in 4 different ways: public, private, community, hybrid. Cloud-computing can be used to develop SaaS(Software as a service) for thin-clients as well as thick-clients.

3. Data Security, Privacy, Availability and Integrity in Cloud Computing:

Issues and Current Solutions^[8]

Abstract:

Cloud computing changed the world around us. Now people are moving their data to the cloud since data is getting bigger and needs to be accessible from many devices. Therefore, storing the data on the cloud becomes a norm. However, there are many issues that counter data stored in the cloud starting from virtual machine which is the mean to share resources in cloud and ending on cloud storage itself issues. In this paper, we present those issues that are preventing people from adopting the cloud and give a survey on solutions that have been done to minimize risks of these issues. For example, the data stored in the cloud needs to be confidential, preserving integrity and available. Moreover, sharing the data stored in the cloud among many users is still an issue since the cloud service provider is untrustworthy to manage authentication and authorization. In this

paper, we list issues related to data stored in cloud storage and solutions to those issues which differ from other papers which focus on cloud as general.

Inference:

Characteristics of cloud computing are on-demand self-service, broad network access, resource pooling, Rapid elasticity, measured service. Traditional security architecture is broken because the customer does not own the infrastructure any more. Outsourcing data on cloud leads to loss of control over its storage. The threats to data from an outsider or an insider endangers the data integrity, data confidentiality and data availability . Therefore maintaining privacy and security of data becomes vital for cloud-computing services especially where money is involved. User authentication (usually 2-level) can be used to maintain data integrity, encryption for data confidentiality and data distribution for availability.

4. International Journal OF Engineering Sciences & Management Research
[http://© International Journal of Engineering Sciences & Management Research \[47\] IMPLEMENTATION OF WEB APPLICATION USING LARAVEL FRAMEWORK \[9\]](http://© International Journal of Engineering Sciences & Management Research [47] IMPLEMENTATION OF WEB APPLICATION USING LARAVEL FRAMEWORK [9])

Abstract:

This paper contains design and implementation process of a complex Laravel application. Laravel is most advance and elegant PHP framework & it make the development process simple. It processes non-business logic on its own that allows programmers to focus on business logic of an application. It has three interconnected parts called Model, View and Controller. Model consist of data and business logic. A view uses Laravel's blade templating engine which works seamlessly with HTML for representation of information. Controller classes are used are used to organize the behavior of request handling logic. Related HTTP request handling logic can be grouped into controller class. With Eloquent ORM active record implementation is done for working with database. Each table in MySQL database has a corresponding model which is used to perform CRUD operations on it. Third party packages can be integrated easily into Laravel application by writing few artisan commands. These packages save tremendous amount of development time. Most of these packages are well tested, stable and available for free of cost.

Inference:

Laravel is an excellent framework because it can generate HTML, do web serving and provides database management. Controllers, migration, user authentication are some of the excellent services provided in this framework. Search controller, user controller are some of the inbuilt

controllers available in laravel which make creating website easier. CSRF tokens can be used to avoid cross-site scripting. Authentication is provided by auth middleware in laravel. Laravel is a good choice for complex website design as it has many inbuilt functionalities.

5. Implementation of a Cloud in Banking Sector [10]

Abstract:

A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers. Cloud Computing has emerged as a new era in IT. A number of banks are now adopting cloud technologies to fulfill their varied purposes. Cloud technology offers business models for delivering innovative client experiences, effective collaboration, upgraded speed to market and enhanced IT efficiency. Using cloud computing banks can create a flexible and agile banking environment that can quickly respond to new business needs. In this research paper, I mainly focus on, how cloud computing can be used in the banking industry, various business models associated with it and the problems faced by the banking industry in adopting this technology.

Inference:

Different models in cloud-computing are Software as a service (SaaS), Platform as a service (PaaS), Infrastructure as a service (IaaS). SaaS is a good option for small finance institutes as they do not need to maintain platform or infrastructure for the system. They can just subscribe to use the software with databases. This will reduce the technical team costs substantially. Hybrid cloud models are better because they provide combined facilities of different cloud models.

6. Creation of Digital Libraries in Indian Languages Using Unicode [11]

Abstract:

Unicode is 16-bit code for character representation in a computer. Unicode is designed by Unicode consortium. It represents almost all the world script including extinct many of extinct scripts like Bramhi and Kharosthi. ISCII is another code developed for to represent the Indian characters in computer but there are problems with character representation using ISCII. It is found that Unicode can solve the problem. This paper suggests measures for creation of Digital library in Indic languages and the problem associated with Unicode.

Inference:

ASCII cannot be used to represent more than one script, therefore to store the data in Devanagari script Unicode is necessary. For unicode to work in website, the browsers should be unicode enabled. Unicode is the way to store Indian script data in SQL databases. UTF-8 and UTF-16 are the 2 versions of unicode available.

7.Database in Cloud Computing - Database-as-a Service (DBaaS) with its Challenges [12]

Abstract:

The Cloud computing (CC) has been widely recognized as the next generation's computing infrastructure [6]. CC offers many advantages by allowing users to use infrastructure like servers, networks, and storages, platforms containing middleware services and operating systems, and softwares for application programs. Cloud computing is an emerging style of IT delivery that intends to make the Internet the ultimate home of all computing resources. Cloud computing has emerged as a popular model in computing world to support processing large volumetric data using clusters of commodity computers. The delivery of Services by cloud Service Providers in terms of DBaaS is important as the cloud environment gives access to centralized shared hardware, software and other information. This paper introduces a new transactional "database-as-a-service" (DBaaS) called Database/Relational Cloud. A DBaaS promises to move much of the operational burden of provisioning, configuration, scaling, performance tuning, backup, privacy, and access control from the database users to the service operator, offering lower overall costs to users. Database as a service has several major issues and concerns, such as data Scalability, Elasticity, Availability, security, expectations, etc, issues. Proposed solutions include risk management, better contractual agreements, database encryption, and authenticity techniques. By bettering these situation the DBaaS service in cloud computing is effective to manage today's vast growing datasets.

Inference:

We can provide Databases as a service (DbaaS) in SaaS. The main challenges are scalability of the database in case of huge amounts of data. With DBaaS the need for database administrator (DBA) and DBMS developer at the institute end is eliminated. But the institution or organisation loses control over how the data is stored and where it is stored. It can only determine what is to be stored and who can access it.

8. Utilization of Laravel Framework for Development of Web Based Recruitment Tool [13]

Abstract:

Current era is of Internet. Nowadays majority of business processes are done online. Companies prefer to post vacancies on their website and accept resumes of eligible candidates. Millions of people are posting their resumes on such websites. This task of selecting and verifying resumes is laborious, time consuming but the accuracy of data is in question. Therefore it was thought of computerizing this activity. In this system, we are proposing a model or tool that will extract the needed information of eligible candidates. Proposed model uses searching and sorting algorithm. This algorithm efficiently and accurately searches for and sorts the qualifying candidates. Hence, proposed system gives accurate output and HR can either select or reject a candidate with respect to some given query terms. This tool is very useful for companies/organizations for candidate recruitment and further management of other processes. It includes Human Resource management activities and Interviewer activities and thus helps them to efficiently handle their recruitment and related activities.

Inference:

Currently, laravel is the most popular framework for developing a web design. Laravel has a provision for communicating with different databases. Eloquent in laravel offers pagination. Security, encryption and validation are some of the features of laravel which are necessary for a banking software.

2.2 Books / Articles referred / news paper referred

1.Present credit banking system software implemented on individual system

A current desktop system for credit societies exists with local databases. We studied the user interfaces and functionalities provided by this system.

2.Laravel 5.4 official documentation

The official documentation for Laravel 5.3 provides a valuable insight into the working of Laravel, its features and syntax. It will prove helpful in development of the website using laravel framework.

3.Security for Cloud Computing Ten Steps to Ensure Success Version 2.0

Cloud Standards Customer Council has published some standards to be maintained for security and privacy in cloud computing. This documentation provides stepwise explanation for creating a successful secure environment for cloud-computing. The landscape , methodology and assessment of security in cloud is described briefly.

CHAPTER 3: REQUIREMENT GATHERING

3.1 Functional Requirement

- User Hierarchy:

5-Level Multilevel and Multilateral user hierarchy has been maintained, with each level having different access and grants to various facilities.

- Income and expenditure details:

For the employees to get information about the monthly income of all the family members of the applicant and monthly expenditure of the applicant.

- Expense Summary Generation:

To generate a report based on expenditure and income details about savings.

- Log In / Sign-up:

For the institute's employees

- Photo upload:

Scanned signature and photo upload for customer verification.

- Credit Society Registration:

Registering new credit societies along with branches for each of them. Add branch, update branch details, society details, etc.

- User registration at different levels:

Create new users at different levels of hierarchy: Admin level, State level, City level, Branch level, Clerk and Accountant level. Registration control. Only higher level users can create new users at a lower level. An admin can create an admin as it is the highest level

3.2 Non Functional Requirements

- **Portability**

The Web Application must be portable over different computing platforms.

- **Scalability**

The ability to extend the system should be achieved with minimum efforts.

- **Maintainability**

Ease in maintenance of the product.

- **Reliability**

The system must be kept in a safe and reliable functioning condition.

- **Security**

The system must be secure from any type of cyber attack.

- **Transparency**

Transparency is to be maintained with respect to the actions performed by different users on the same level as well as different level users of the system.

- **Durability**

Assuring that most objects are stable over time and don't need further changes.

3.3 Constraints

- Suitable only for low volume of data entry.
- As it is SaaS, there are limitations on personalization and tailoring. i.e. The software cannot be customized differently for each user.
- Lack of net connection or interruption in net connection will result in unavailability of the web application.
- User should have general understanding of computers and using net services. He should know basic banking and finance terminologies and phenomena.

3.4 Hardware, Software , Technology and tools - Requirements

3.4.1 Hardware Requirements:

- Thin Client with :
 - a. Minimum 2GB RAM
 - b. Active Net Connection

3.4.2 Software Requirements:

Client Side :

- Thin-Client Responsive Web Browser
- Virtual Keyboard (for Devanagari)

Server side :

- PHP >=7.0.0
- OpenSSL PHP Extension
- XML PHP Extension
- Composer
- pgadmin / SQL Developer (PostgreSQL interfaces)

3.4.3 Techniques and Tools

- Laravel 5.3 framework
- PgAdmin
- Sublime Text
- Bootstrap
- PostgreSQL
- Technologies are:PHP-7 , CSS , HTML , SQL

3.5 Selection of the Hardware, Software , Technology and tools**PostgreSQL:**

It is object RDBMS. It supports NOSQL and JSON. It is Better for Joins and strict about data validation. It also supports multi dimensional array and user defined types. is a management tool for PostgreSQL, easy to use and easy to fire basic SQL queries.

Pgadmin: It is a management tool for PostgreSQL, easy to use and easy to fire basic SQL queries.

Laravel:

Laravel is a strong PHP framework. It makes the best use of HTML and builds the efficient websites and applications. Laravel eliminates the use of complex code. In fact, it makes the entire development very easy.

CHAPTER 4: PROPOSED DESIGN

4.1 System design/Conceptual Design

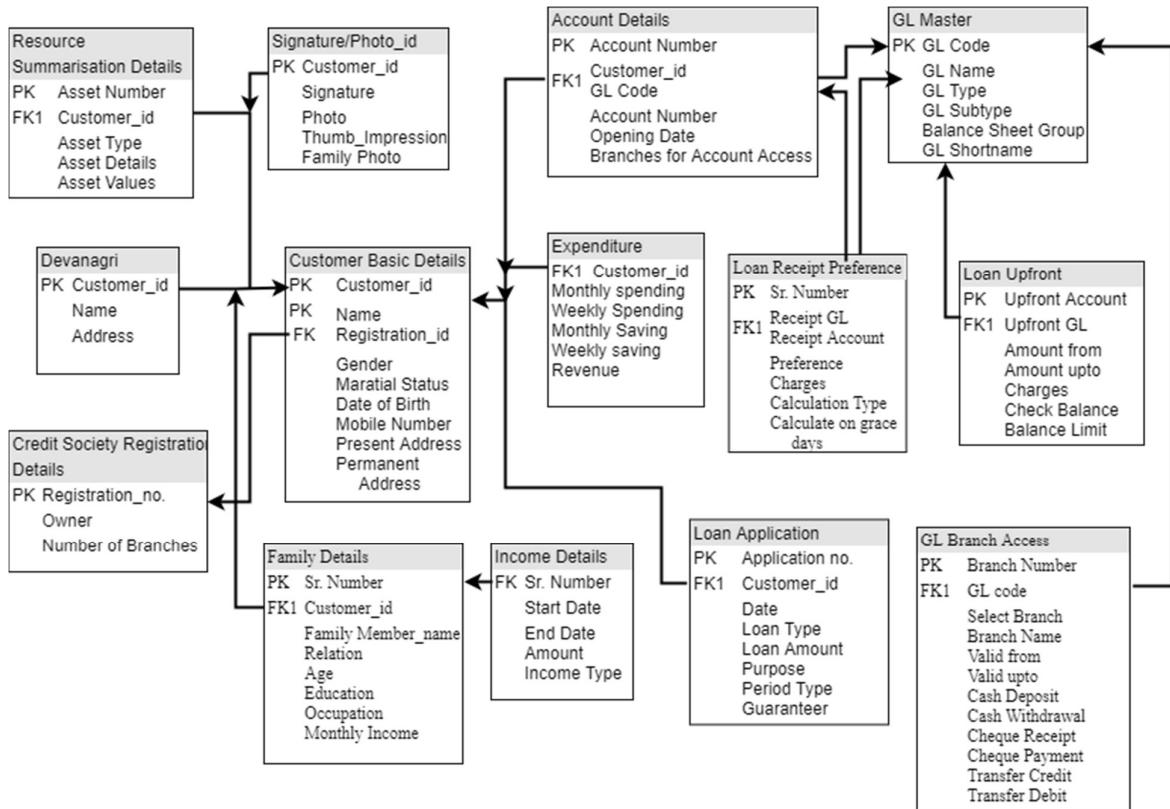


Fig. 4.1 System Architecture

4.2 Block diagram of the system

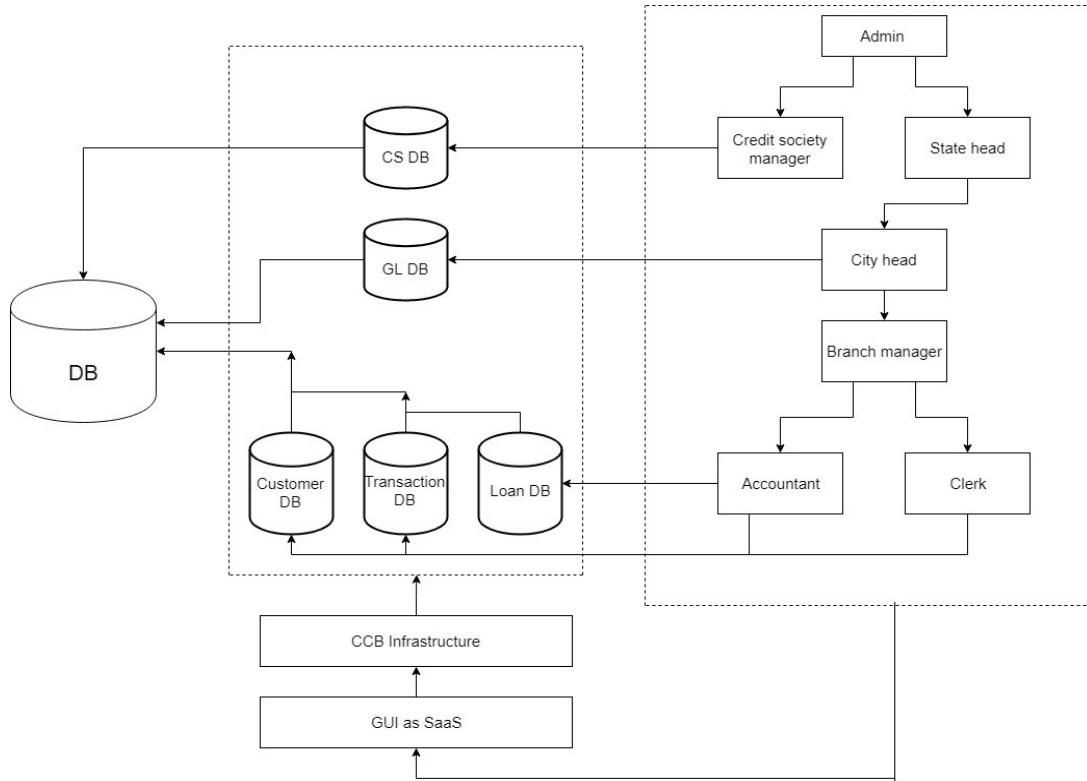


Fig. 4.2 Block Diagram of MahCredit

- **Users:** The administrators are level 0 users who have the highest privileges. The owners or managers of the individual credit societies are level 2 users which avail the services provided by MahCredit. The accountants and clerks are the level 4 users. There is also state level (level 1), district level and city level abstraction for the given user hierarchy. Accountants and clerk are the users on the lowest level of users who will take care of the day to day transactions and updates. The user hierarchy is multi-level as well as multilateral.
- **GUI for SaaS:** This is only part of the system in which user can enter features according to their requirements. GUI for SaaS is the graphical user interface provided by the system to its users. It is the topmost layer of the cloud-based system. It is necessary for user interaction. All the forms and tables are displayed or edited using the graphical user interface.
- **CCB infrastructure:** It is basically the connection between the database and the graphical user interface. Data processing, validation, authentication and conclusions are drawn in this layer of the architecture. This layer is vital to the proper working and functionality of the system.
- **DB:** DB stands for databases created for different entities of the system. The first step when a user avails the services of MahCredit is his registration in the CS DB i.e. the credit society database.

4.3 Modular diagram of the system

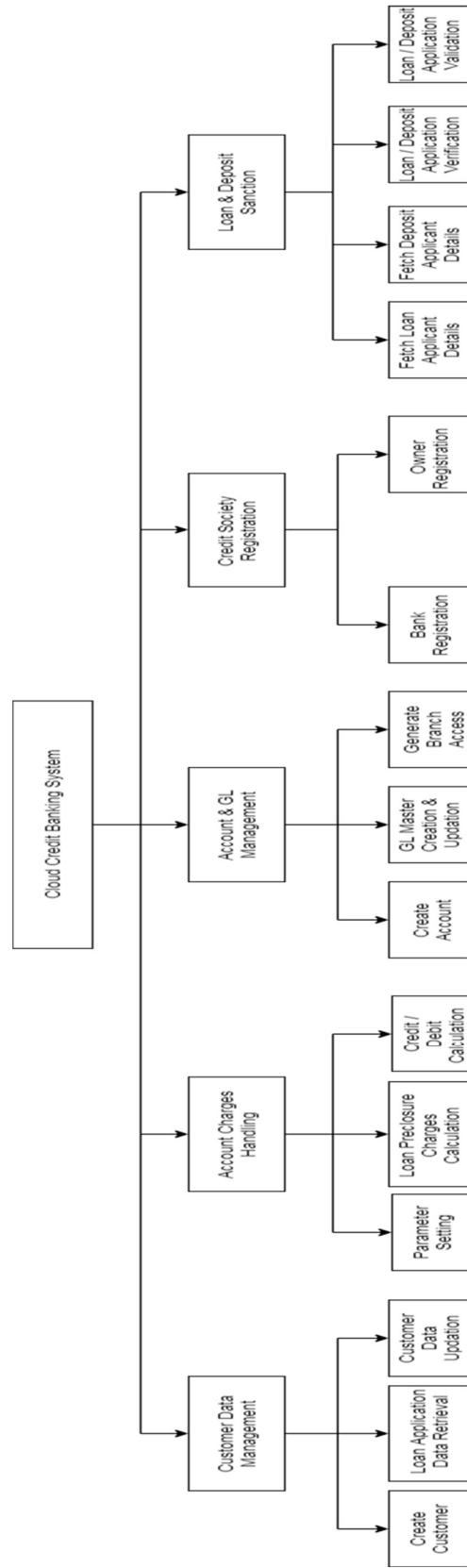


Fig. 4.3 Modular Diagram Of MahCredit

The modules under this system are :

- **Customer Data Management** : Creating customer database and updating it on timely basis is done under this module. Basically individual customer application, family details, income details, expense details are taken care of by customer data management
- **Account Charges Handling:** Calculation of credit, debit, account balance, pending loan amount is included in this module.
- **Account and GL management:** Creating GL master which is the core data file for all transactions, an account to access GL master, maintaining and periodically updating the GL master database and activities is done.
- **Credit Society Registration:** The bank and the owner of the credit society get registered with the system and their database is created. Different users related to this credit society like city head(if not present), state head(if not present), clerk are created in this module.
- **Loan and Deposit Sanction:** This module includes loan application details, deposit application details, loan application verification and validation, etc.

4.4 Detailed Design (DFD - level 0,1,2, State Transition Diagram, ER Diagram, etc...)

4.4.a ER Diagram

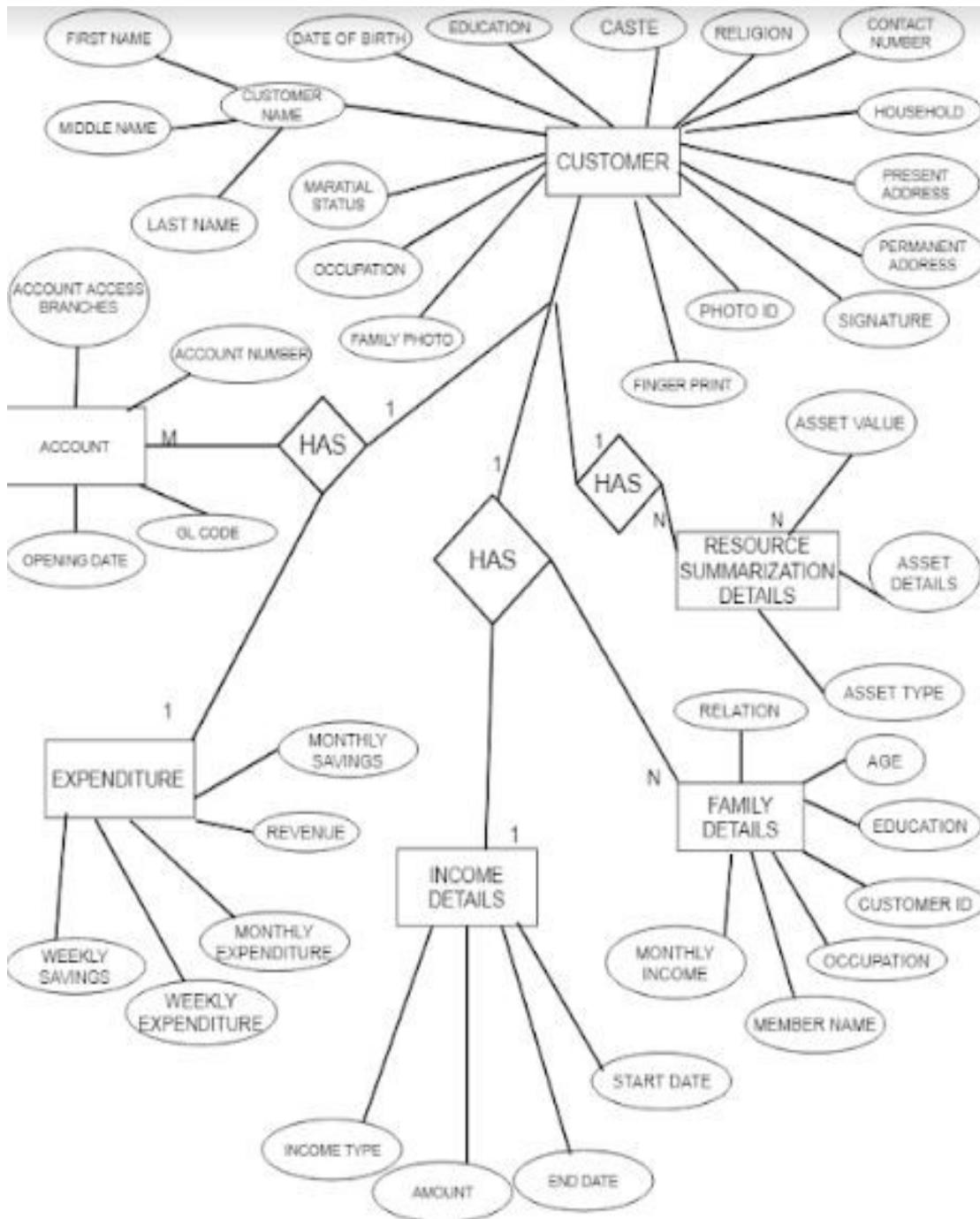


Fig. 4.4.a(i) ER Diagram-1

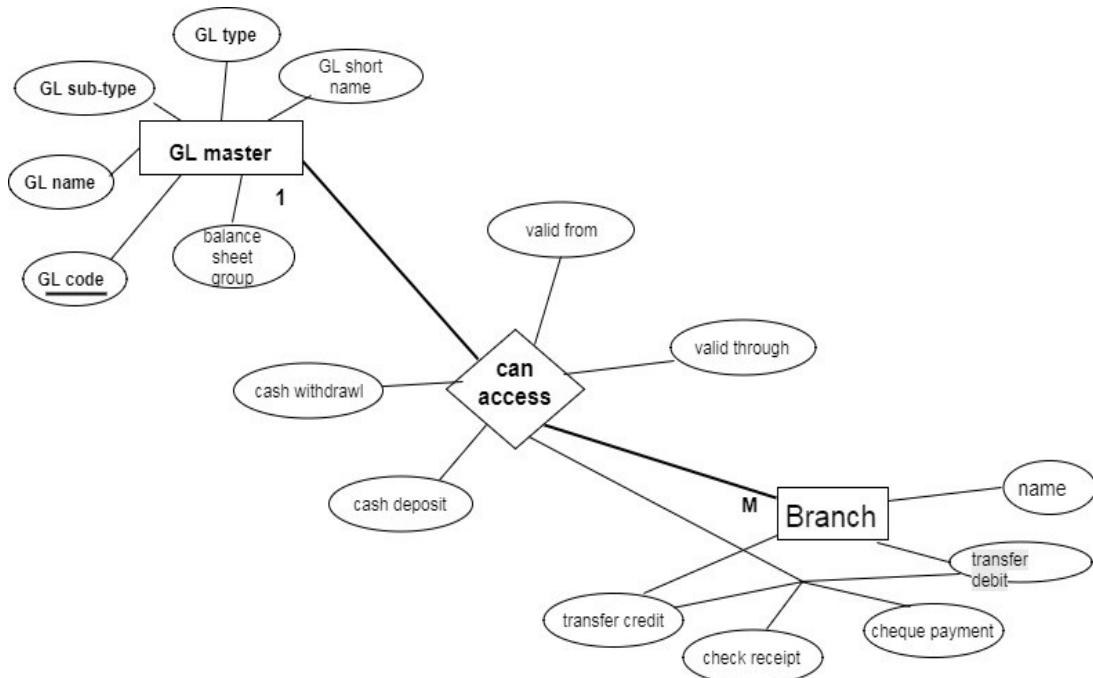


Fig. 4.4.a(ii) ER Diagram-2

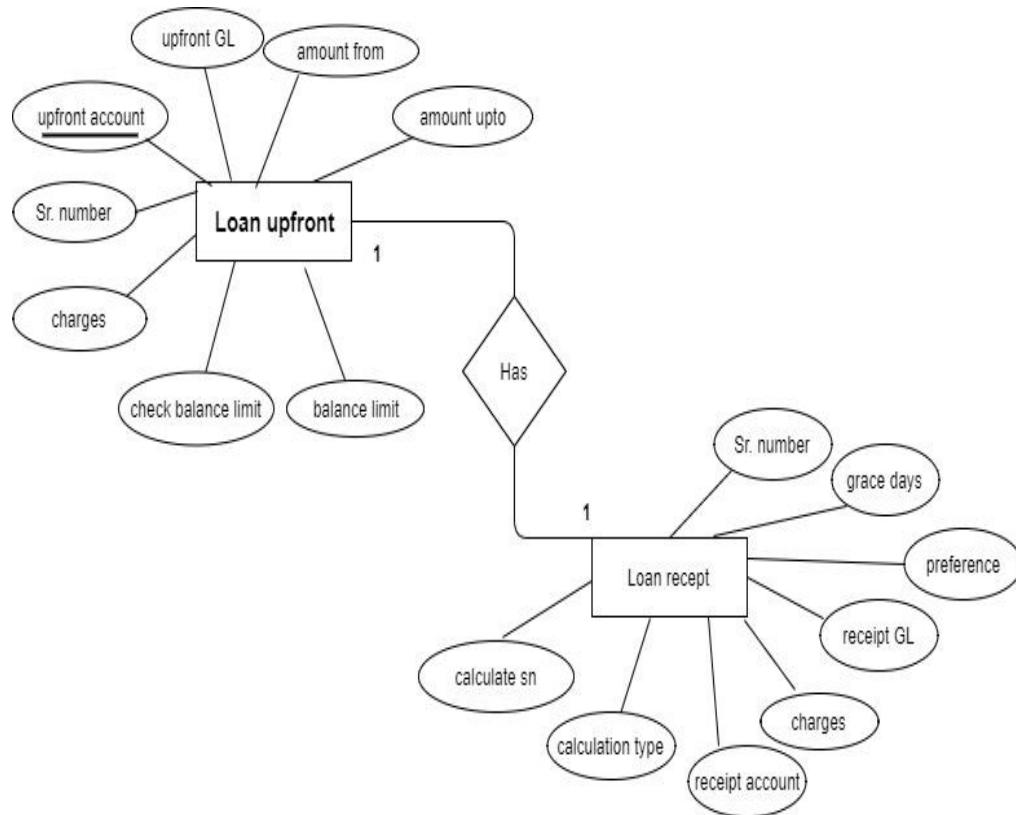


Fig. 4.4.a(iii) ER Diagram-3

4.4.b Flowchart

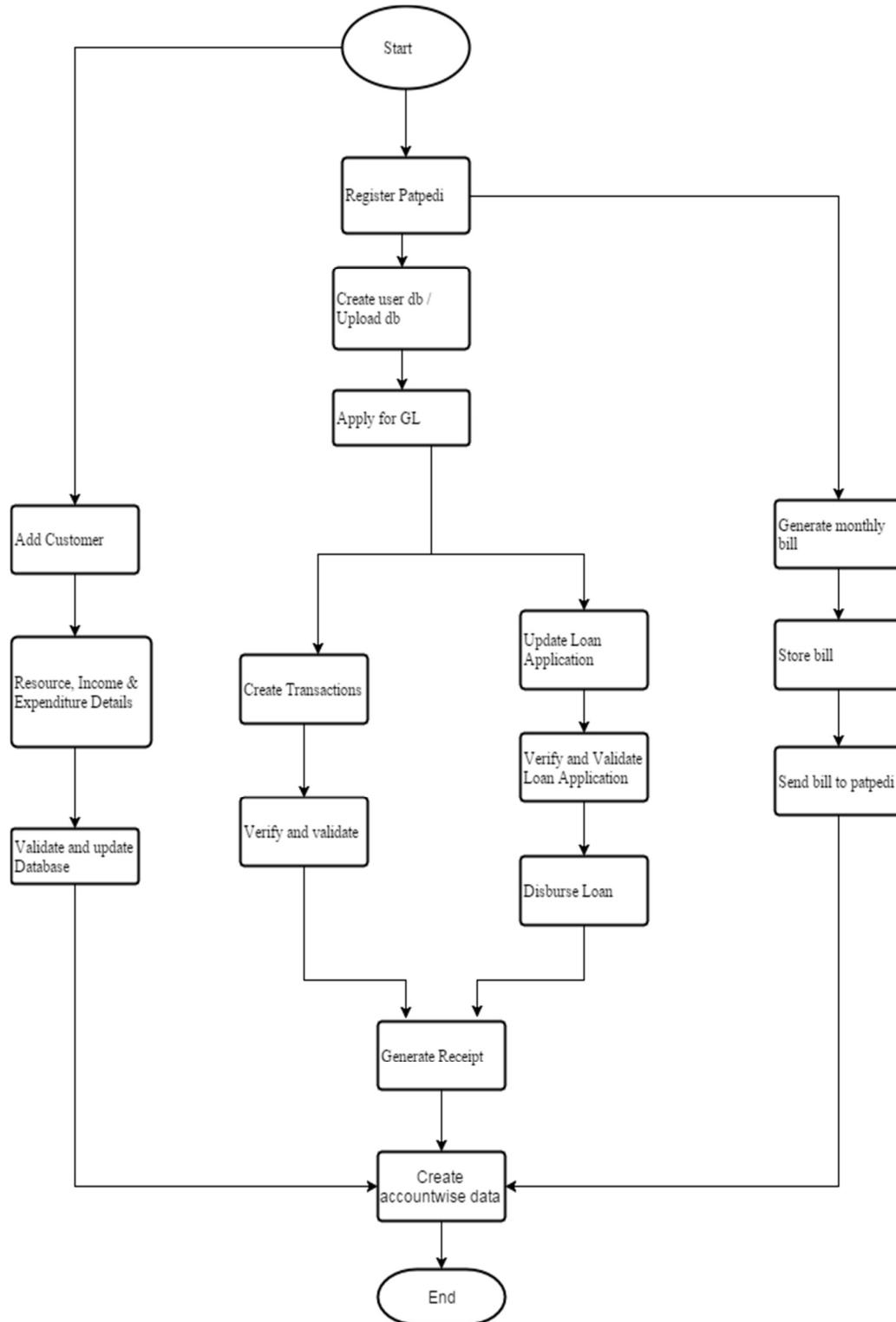


Fig 4.4.b Flowchart of MahCredit

4.4.c Data Flow Diagram

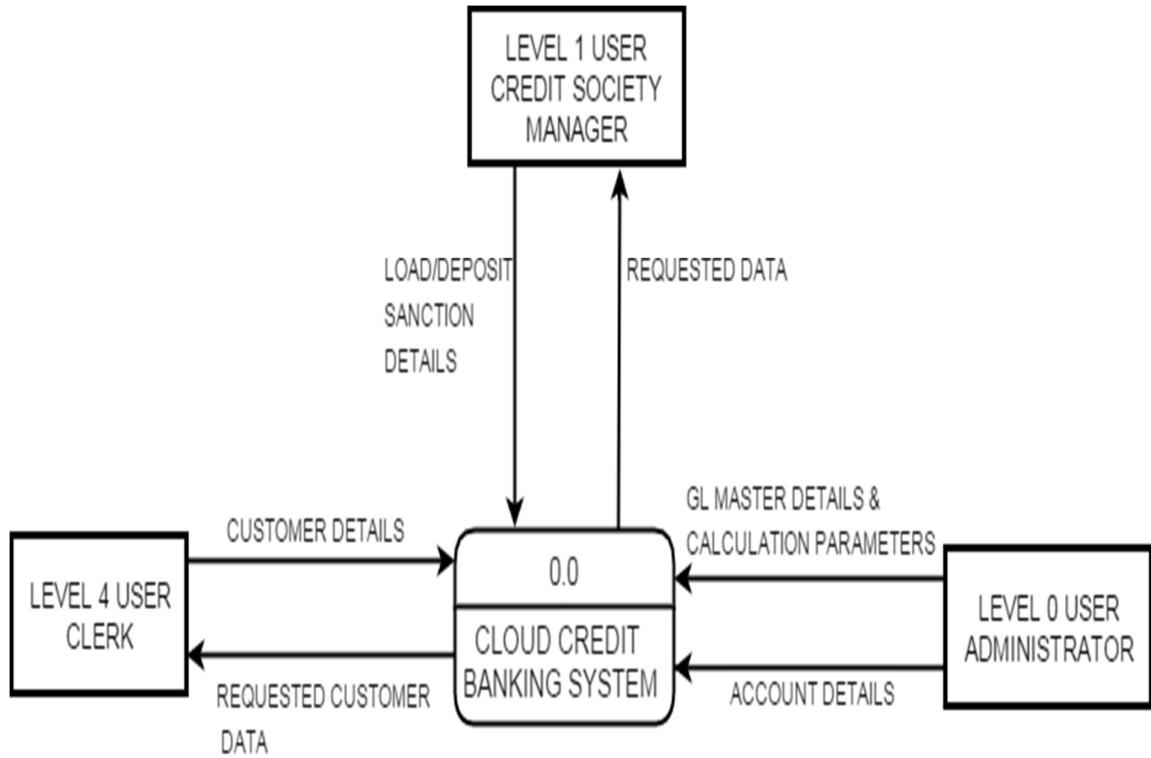


Fig. 4.4.c(i) Level-0 Data Flow Diagram

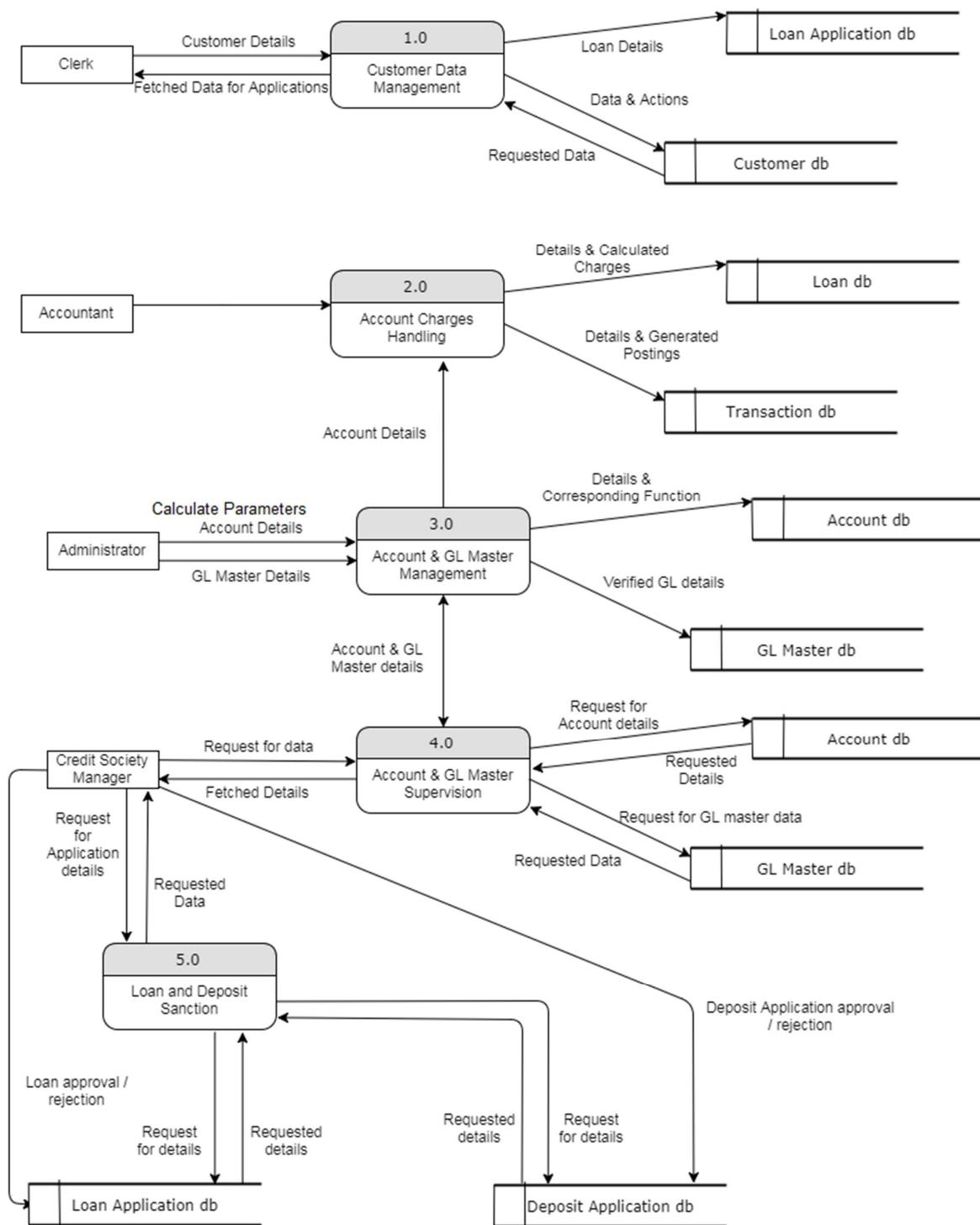


Fig. 4.4.c(ii) Level-1 Data Flow Diagram

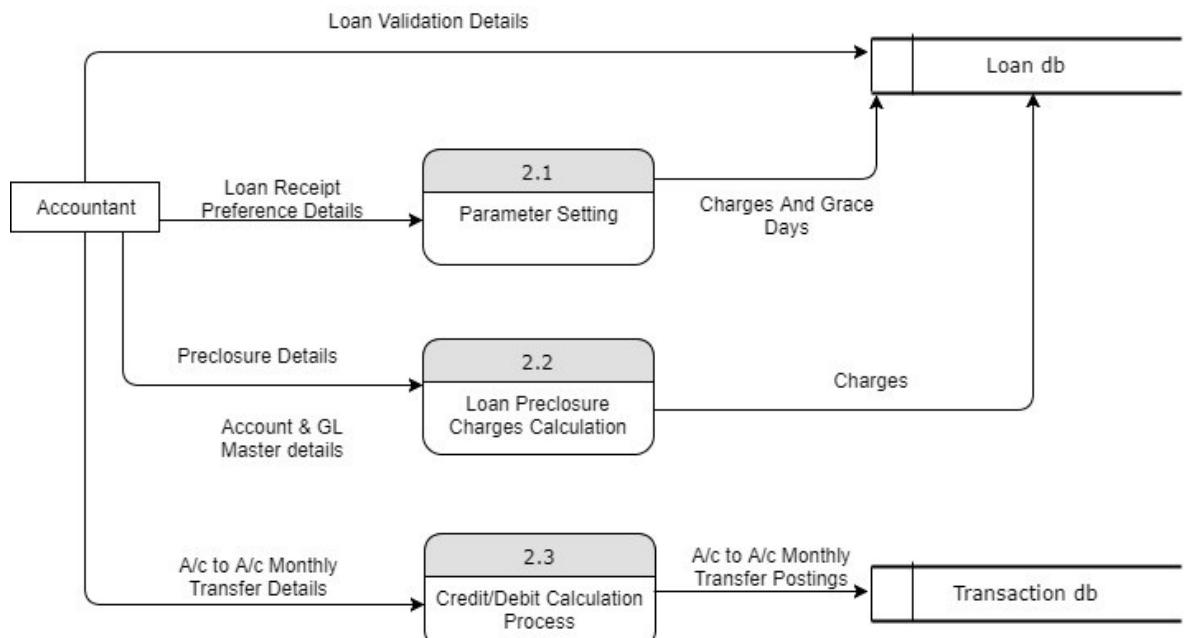


Fig. 4.4.c(iii) Level-2 Data Flow Diagram 1

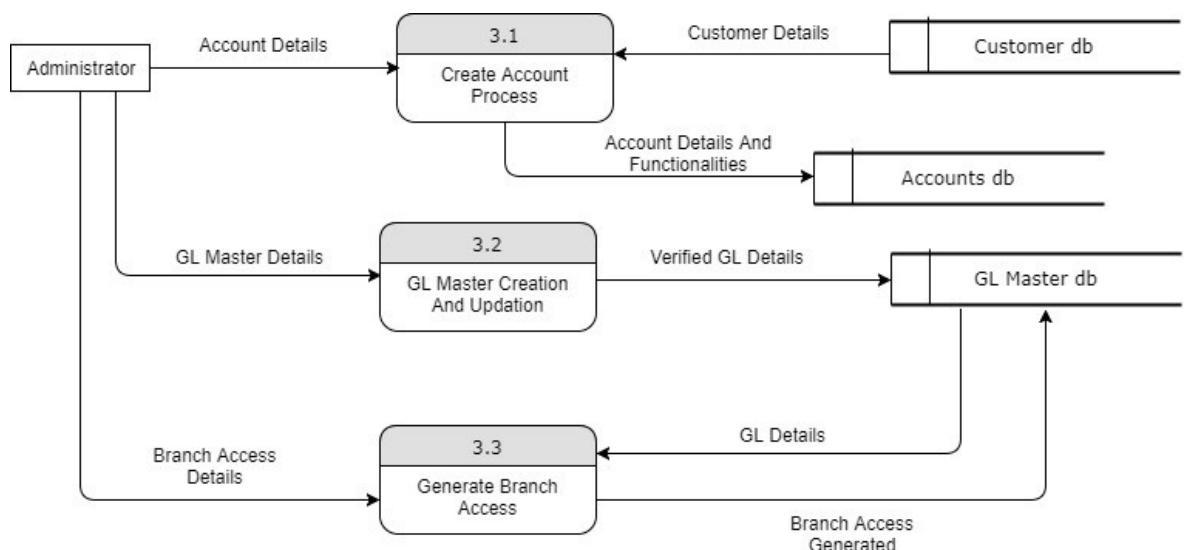


Fig. 4.4.c(iv) Level-2 Data Flow Diagram 2

4.4.d Project Scheduling & Tracking using Timeline / Gantt Chart:

Task Name	Start Date	End Date	Duration	Predecessors	% Complet Status
1 Problem Definition	06/13/17	06/14/17	2d		100% Completed
3 Requirement Gathering and Analysis	06/13/17	07/05/17	15d	1	100% Completed
4 Milestone : SRS Approval	07/05/17	07/05/17	0	2	100% Completed
5 Determine Project Team	06/15/17	06/15/17	1d	1	100% Completed
6 Project Commencement Meeting	06/16/17	06/16/17	1d	4	100% Completed
7 Research On Project Domain	06/19/17	07/14/17	20d	5	100% Completed
8 Design System Architecture	07/17/17	08/01/17	12d	6	100% Completed
9 Analyse and Design Use Case	08/02/17	08/17/17	12d	7, 3	100% Completed
10 Second Meeting	07/14/17	07/14/17	0	6	100% Completed
11 Design DFD	08/18/17	09/07/17	15d	8, 9	100% Completed
12 Milestone : DFD Approval	09/07/17	09/07/17	0	10	100% Completed
13 Design ER Diagram	08/18/17	09/04/17	12d	8, 9	100% Completed
14 Milestone : ER Approval	09/04/17	09/04/17	0	12	100% Completed
15 Develop Project Plan	09/08/17	09/22/17	11d	11, 13	100% Completed
16 Design Class and Database Models	09/08/17	09/26/17	13d	11, 13	100% Completed
Milestone : Approval of design model and document (Project Review 1)	09/26/17	09/26/17	0	14, 15	100% Completed
18 Design Modular Diagram	09/27/17	10/13/17	13d	16	
19 Design Login Module	10/16/17	10/18/17	3d	17	100% Completed
Knowledge Acquisition of Technologies being used	07/06/17	09/29/17	62d	3	100% Completed
21 Prototype Login Interface	10/19/17	10/25/17	5d	18, 19	100% Completed
21 Prototype Login Interface	10/19/17	10/25/17	5d	18, 19	100% Completed
22 Create Project Report	10/26/17	10/27/17	2d	20, 17	100% Completed
23 Milestone: Project Review 2	10/27/17	10/27/17	0	21	100% Completed
24 Coding	09/27/17	03/13/18	120d	16	100% Completed
25 Develop test cases	10/30/17	12/08/17	30d	21	100% Completed
26 Testing	12/11/17	03/23/18	75d	24	100% Completed
27 Milestone : Review and approve testing	03/23/18	03/23/18	0	25	100% Completed
28 User Testing	03/26/18	03/27/18	2d	24, 26	100% Completed
29 Milestone : User acceptance test	03/27/18	03/27/18	0	27	100% Completed
30 Project Management	03/28/18	04/13/18	13d	28	90% In Progress
31 Project Status Meetings	03/28/18	04/12/18	12d	29SS	85% In Progress
32 Milestone : Plan	04/13/18	04/13/18	0	29	80% In Progress
33 Updates files/records	04/16/18	04/18/18	3d	31	90% In Progress
34 Gain Formal Acceptance	04/19/18	04/19/18	1d	32	75% In Progress
35 Archive File/Documents	04/20/18	04/23/18	2d	33	50% In Progress

Fig. 4.4.d Timeline for MahCredit

CHAPTER 5: IMPLEMENTATION DETAILS

5.1. Evaluation of the developed system

- All the modules are running as expected with 100% accuracy. Expense summary calculation is correct.
- The system can be considered highly effective for limited database for a single society as all the results obtained coincide with the stated requirements by the industry.

CHAPTER 6: TESTING

6.1 SQL Injection :

MahCredit

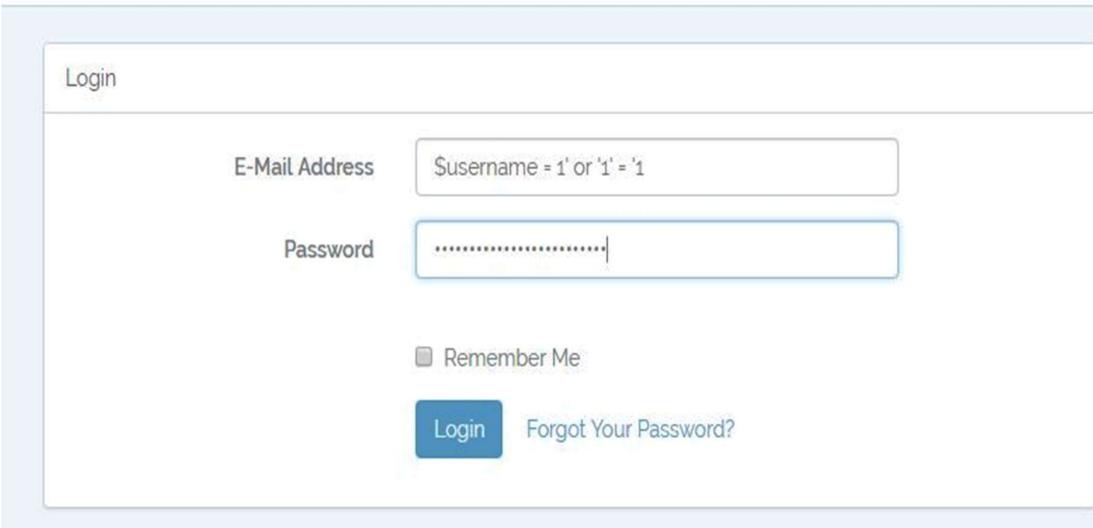
Login

E-Mail Address

Password

Remember Me

[Forgot Your Password?](#)



This screenshot shows a login interface for 'MahCredit'. The user has entered '\$username = '1' or '1' = '1'' into the E-Mail Address field. The password field contains several dots. Below the fields are 'Remember Me' checkboxes and 'Login' and 'Forgot Your Password?' buttons.

Fig. 6.1.1 Trying SQL Injection

MahCredit

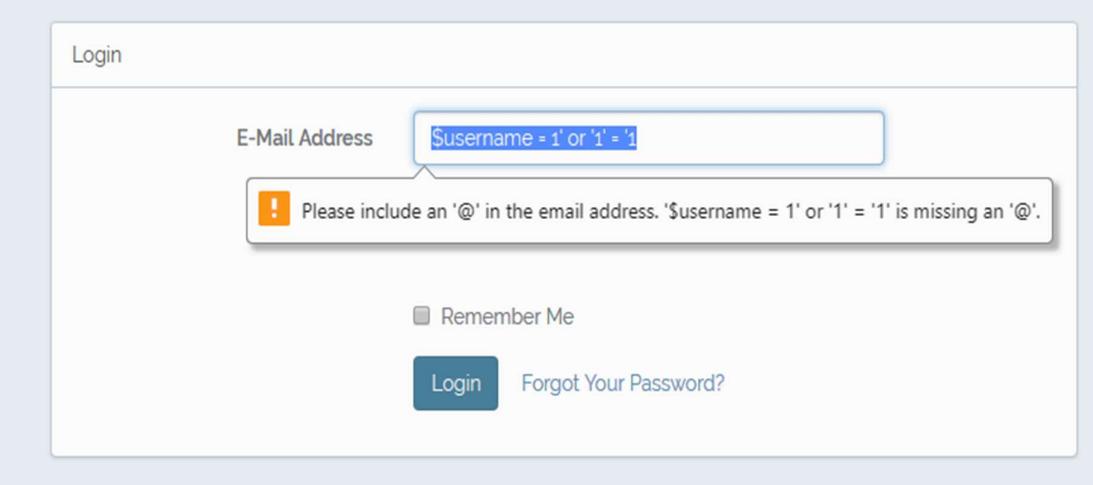
Login

E-Mail Address

 Please include an '@' in the email address. '\$username = '1' or '1' = '1' is missing an '@'.

Remember Me

[Forgot Your Password?](#)



This screenshot shows the same login interface as Fig. 6.1.1. However, the input field now contains '\$username = '1' or '1' = '1''. A tooltip-like message appears over the field, stating 'Please include an '@' in the email address. '\$username = '1' or '1' = '1' is missing an '@''. The rest of the interface remains the same.

Fig. 6.1.2 Failed SQL Injection

6.2 Broken Access

The screenshot shows a web browser with the URL `localhost:8000/customer/9` in the address bar. The page title is "MahCredit". On the left, there is a logo of three stylized human figures. The main content area has a header "Update Customer". It contains fields for "Customer Id" (value: 8), "First Name" (value: Mahindra), "Middle name" (value: Singh), "Gender" (radio buttons: Male selected, Female unselected), and "Marital Status" (value: Married). The "Customer Id" field is highlighted with a red border.

Fig. 6.2.1 Broken access attack

The screenshot shows a web browser with the URL `localhost:8000/customer/8` in the address bar. The page title is "MahCredit". On the left, there is a logo of three stylized human figures. A pink error message box at the top says "You are trying to access unauthorized url!". The main content area has a header "Update Customer". It contains fields for "Customer Id" (value: 8), "First Name" (value: Mahindra), "Middle name" (value: Singh), "Gender" (radio buttons: Male selected, Female unselected), "Marital Status" (value: Married), "Education" (dropdown: Graduate), "Occupation" (dropdown: Service), "Aadhar No." (value: 646464646464), and "No. of Dependents" (value: 4). The "Customer Id" field is highlighted with a red border.

Fig. 6.2.2 Unsuccessful Broken Access

6.3 Unauthorized User Access

The screenshot shows a web browser window with the URL `localhost:8000/customer` in the address bar. The page title is "MahCredit". On the left, there is a logo of three stylized figures and the text "MahCredit". The main content area has a light blue header with the word "Register". Below it is a form with five input fields: "Name", "E-Mail Address", "Personal Contact No.", "Password", and "Confirm Password". A blue "Register" button is located at the bottom right of the form. The entire interface is set against a white background.

Fig. 6.3.1 Unauthorized user access through URL

The screenshot shows a web browser window with the URL `localhost:8000/register` in the address bar. The page title is "MahCredit". On the left, there is a logo of three stylized figures and the text "MahCredit". The main content area has a light blue header with the word "Register". Below it is a form with five input fields: "Name", "E-Mail Address", "Personal Contact No.", "Password", and "Confirm Password". Above the form, a pink message box contains the text "You are trying to access unauthorized level of user!". A blue "Register" button is located at the bottom right of the form. The entire interface is set against a white background.

Fig. 6.3.2 Unsuccessful unauthorized user access

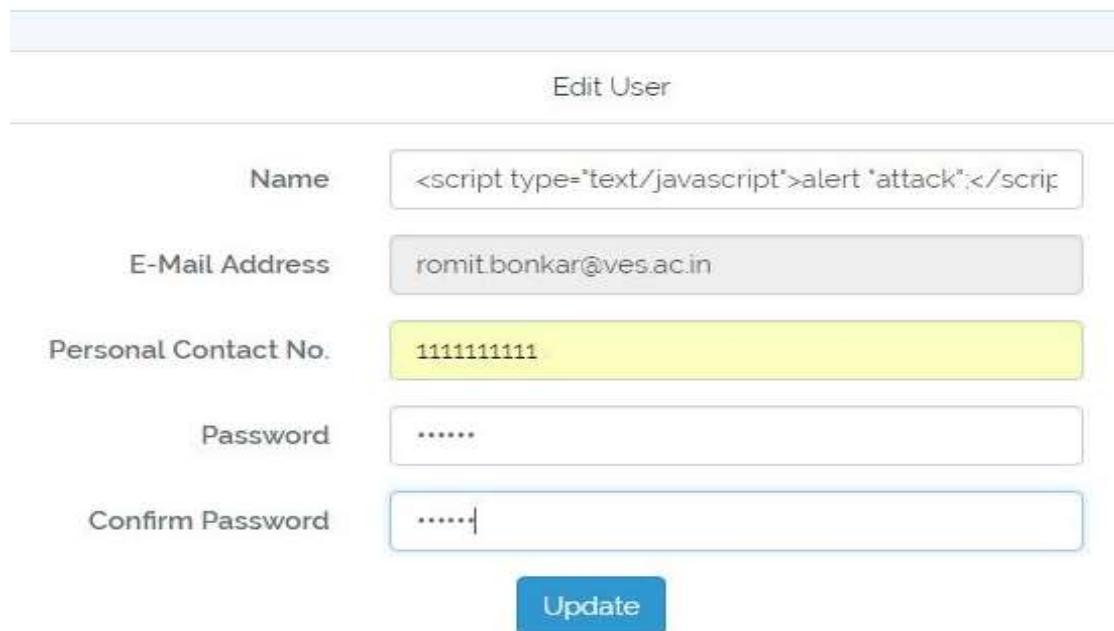
6.4 XSS Attack

MahCredit

Edit User

Name	<input type="text" value="<script type='text/javascript'>alert 'attack'</script>"/>
E-Mail Address	<input type="text" value="romit.bonkar@ves.ac.in"/>
Personal Contact No.	<input type="text" value="1111111111"/>
Password	<input type="password" value="....."/>
Confirm Password	<input type="password" value="....."/>

Update



This screenshot shows a user interface for editing a user profile. The title is 'MahCredit'. Below it is a heading 'Edit User'. There are five input fields: 'Name' containing '<script type="text/javascript">alert "attack"</script>', 'E-Mail Address' containing 'romit.bonkar@ves.ac.in', 'Personal Contact No.' containing '1111111111', 'Password' containing '.....', and 'Confirm Password' containing '.....'. A blue 'Update' button is at the bottom. The 'Name' field's value is highlighted with a yellow background.

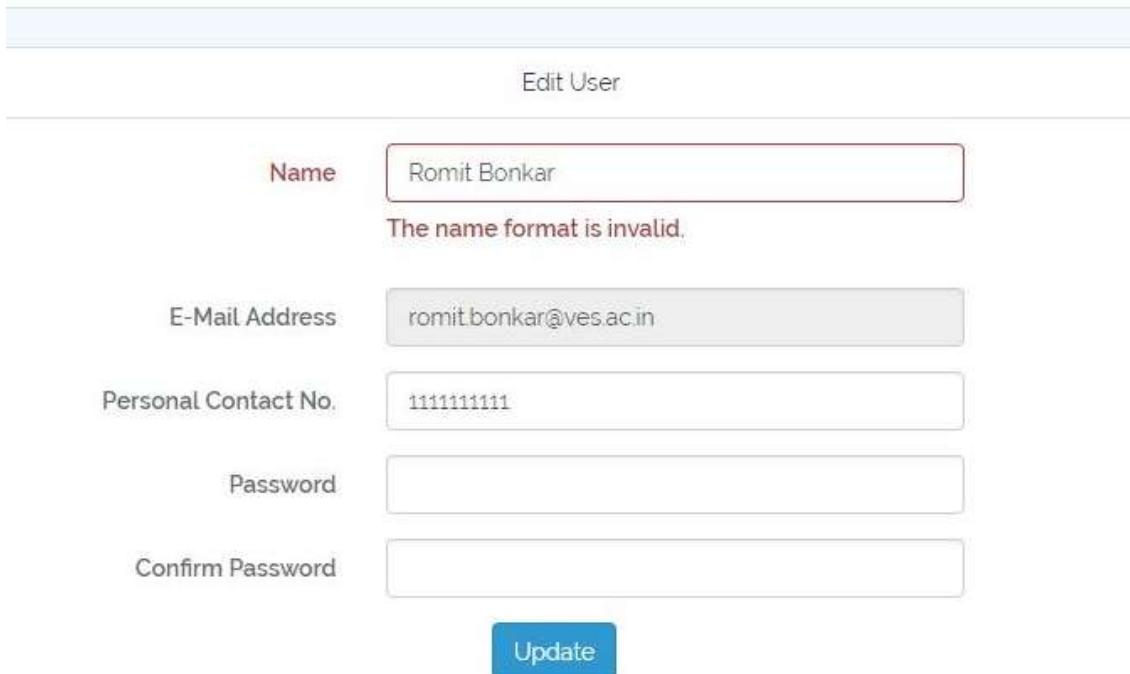
Fig. 6.4.1 XSS attack on name field

MahCredit

Edit User

Name	<input type="text" value="Romit Bonkar"/>
The name format is invalid.	
E-Mail Address	<input type="text" value="romit.bonkar@ves.ac.in"/>
Personal Contact No.	<input type="text" value="1111111111"/>
Password	<input type="password"/>
Confirm Password	<input type="password"/>

Update



This screenshot shows the same user interface as Fig. 6.4.1, but the 'Name' field now contains 'Romit Bonkar'. A red error message 'The name format is invalid.' is displayed below the field. The other fields and the 'Update' button are identical to the previous figure.

Fig. 6.4.2 Failed XSS attack

6.5 Validations:

MahCredit

Register

Name

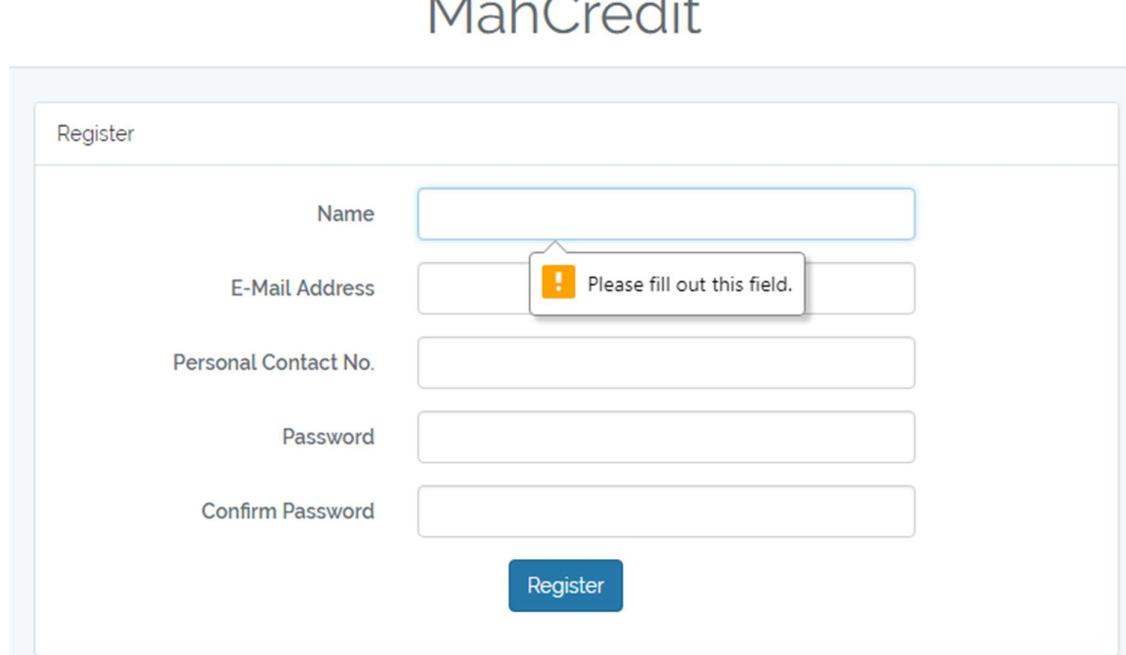
E-Mail Address Please fill out this field.

Personal Contact No.

Password

Confirm Password

Register



This screenshot shows a registration form titled 'MahCredit' with a 'Register' button. The first field, 'Name', is empty and highlighted with a light blue border. A validation message 'Please fill out this field.' is displayed in a callout bubble above it. The other fields (E-Mail Address, Personal Contact No., Password, Confirm Password) are also empty and have standard grey borders. The 'Register' button is at the bottom.

Fig. 6.5.1 Field Validation

MahCredit

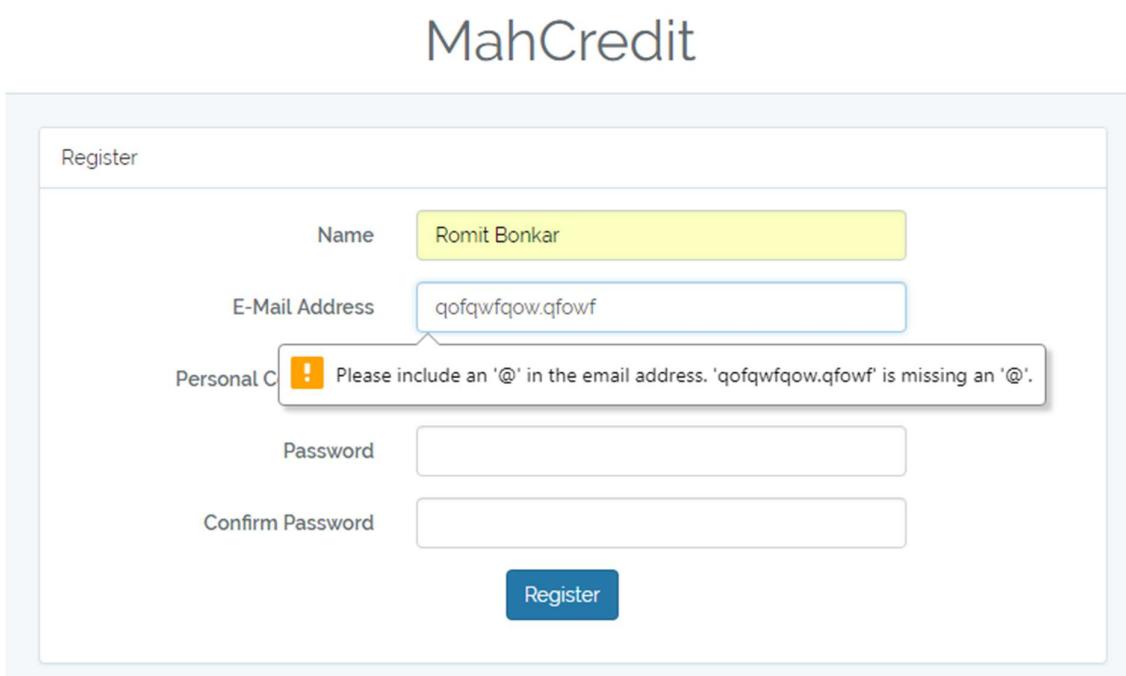
Register

Name

E-Mail Address

Personal C Please include an '@' in the email address. 'qofqwfqow.qfowf' is missing an '@'.

Register



This screenshot shows the same registration form as Fig. 6.5.1, but with an invalid email address entered in the 'E-Mail Address' field. The field is highlighted with a light blue border. A validation message 'Please include an '@' in the email address. 'qofqwfqow.qfowf' is missing an '@'.' is displayed in a callout bubble to the right of the field. The other fields and the 'Register' button are identical to the previous figure.

Fig. 6.5.2 Email Format Validation

MahCredit

Register

Name	<input type="text" value="Romit Bonkar"/>
E-Mail Address	<input type="text" value="romit.bonkar@ves.ac.in"/> The email has already been taken.
Personal Contact No.	<input type="text" value="1111111111"/>
Password	<input type="password"/>
Confirm Password	<input type="password"/>

Register

Fig. 6.5.3 Duplicate Email Notification

MahCredit

Register

Name	<input type="text" value="Romit Bonkar"/>
E-Mail Address	<input type="text" value="romit.bonkar1@ves.ac.in"/>
Personal Contact No.	<input type="text" value="alphabetic"/> The contact no must be a number.
Password	<input type="password"/>
Confirm Password	<input type="password"/>

Register

Fig. 6.5.4 Phone Number Validation

MahCredit

Register

Name	Romit Bonkar
E-Mail Address	romit.bonkar1@ves.ac.in
Personal Contact No.	2222222222
Password	<input type="password"/>
The password confirmation does not match.	
Confirm Password	<input type="password"/>
<input type="button" value="Register"/>	

Fig. 6.5.5 Password Validation

CHAPTER 7: RESULT ANALYSIS

7.1 Parameters considered for development

1. Password length for registration - minimum 6 characters
2. Image size for photo upload
3. Levels of users
4. End user understanding of the system

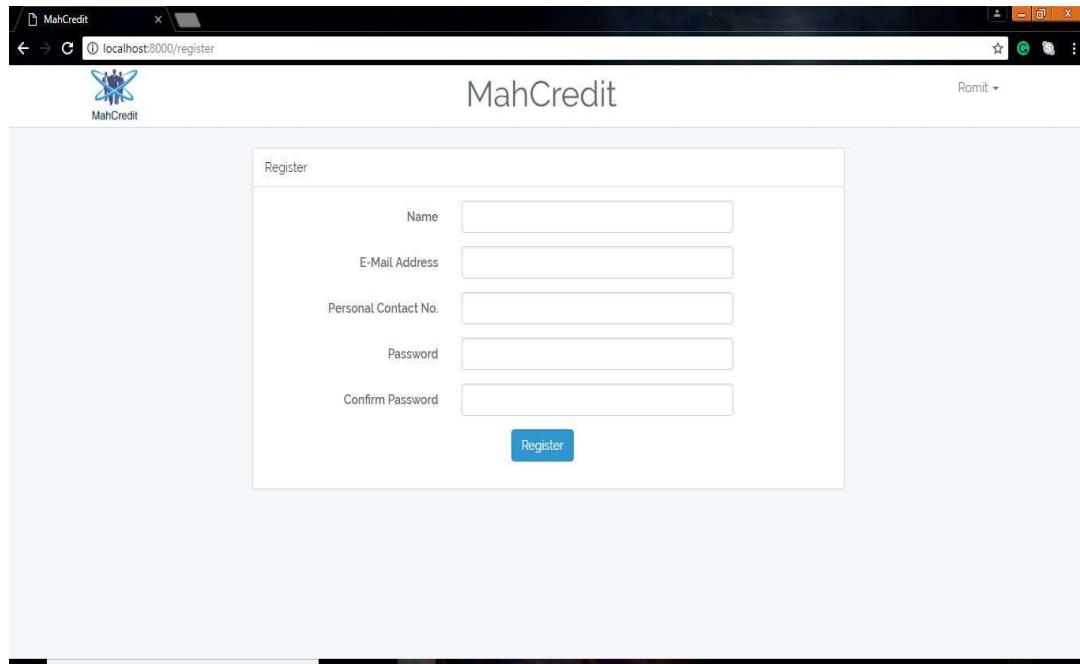
7.2 Screenshots of User Interface (UI)



Fig 7.2.1 Home Page

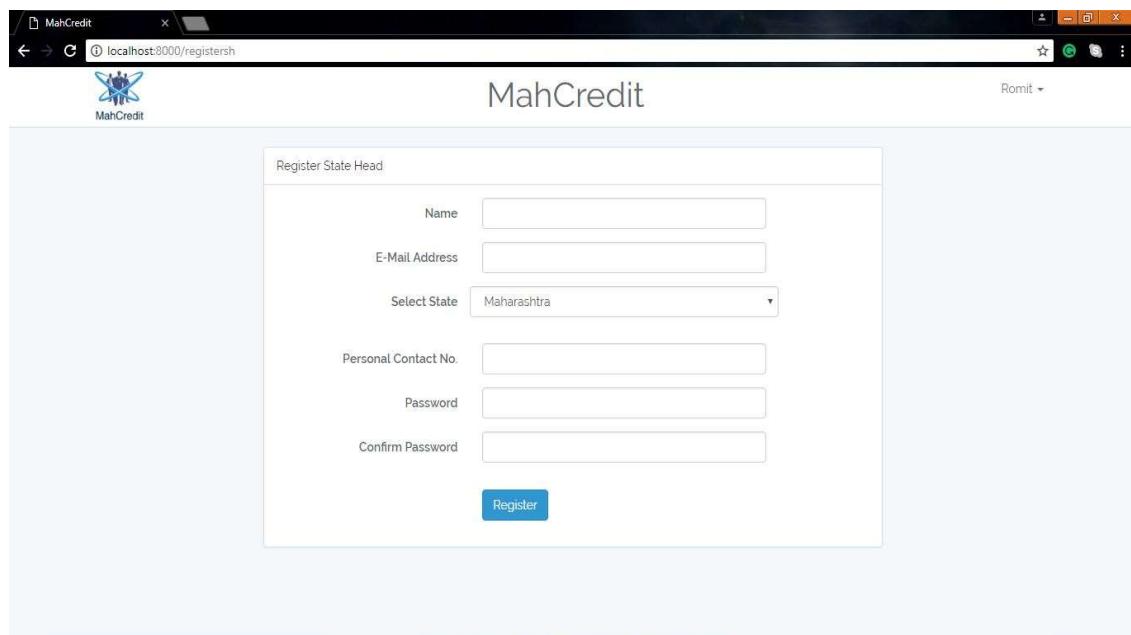
The image shows the login page of the MahCredit website. At the top left is the MahCredit logo, which includes a stylized blue and green butterfly icon and the word "MahCredit". The main title "MahCredit" is centered above a login form. The login form has a light gray header with the word "Login". Below the header are two input fields: "E-Mail Address" and "Password", each with its own input box. Underneath these fields is a "Remember Me" checkbox. At the bottom of the form are two buttons: a blue "Login" button and a link "Forgot Your Password?".

Fig 7.2.2 Login page



The screenshot shows a web browser window for 'MahCredit' at the URL 'localhost:8000/register'. The page title is 'MahCredit' and the sub-page title is 'Register'. It features a logo of a stylized blue and green emblem with the text 'MahCredit' below it. The form contains five input fields: 'Name' (text), 'E-Mail Address' (text), 'Personal Contact No.' (text), 'Password' (text), and 'Confirm Password' (text). A blue 'Register' button is located at the bottom right of the form area.

Fig 7.2.3 Admin Registration



The screenshot shows a web browser window for 'MahCredit' at the URL 'localhost:8000/registersth'. The page title is 'MahCredit' and the sub-page title is 'Register State Head'. It features a logo of a stylized blue and green emblem with the text 'MahCredit' below it. The form contains six input fields: 'Name' (text), 'E-Mail Address' (text), 'Select State' (dropdown menu showing 'Maharashtra'), 'Personal Contact No.' (text), 'Password' (text), and 'Confirm Password' (text). A blue 'Register' button is located at the bottom right of the form area.

Fig 7.2.4 State head Registration

The screenshot shows a web browser window titled 'MahCredit' with the URL 'localhost:8000/registerch'. The page features a logo for 'MahCredit' and a title 'MahCredit'. A user profile 'Romit' is visible in the top right corner. The main content is a form titled 'Register City Head' containing fields for Name, E-Mail Address, Select City (set to 'Thane'), Personal Contact No., Password, and Confirm Password, followed by a 'Register' button.

Fig 7.2.5 City head Registration

The screenshot shows a web browser window titled 'MahCredit' with the URL 'localhost:8000/registeremployees'. The page features a logo for 'MahCredit' and a title 'MahCredit'. A user profile 'Vegeta' is visible in the top right corner. The main content is a form titled 'Register Employee' containing fields for Name, E-Mail Address, Employee Type (set to 'Accountant'), Personal Contact No., Password, and Confirm Password, followed by a 'Register' button.

Fig 7.2.6 Employee Registration

The screenshot shows a web browser window for 'MahCredit' at the URL 'localhost:8000/registercsm'. The page title is 'MahCredit'. On the left, there is a logo for 'MahCredit' featuring a stylized blue and white design. The main content area is titled 'Credit Society Manager Details'. It contains a dropdown menu labeled 'Select Credit Society' with 'Ashayein' selected. Below it are five input fields: 'Name' (empty), 'E-Mail Address' (empty), 'Personal Contact No.' (empty), 'Password' (empty), and 'Confirm Password' (empty). A blue 'Register' button is located at the bottom of the form.

Fig 7.2.7 Credit Society registration

The screenshot shows a table of registered credit societies. The table has columns: Id, Name, Registration Number, No. of Branches, Contact Number, Edit, and Delete. There are four rows of data:

Id	Name	Registration Number	No. of Branches	Contact Number	Edit	Delete
19	Ashayein	123	0	2222222222		
22	Lamhein	432	0	8888888888		
20	Udaan	1234	0	2222222222		
21	XYZ	111	0	1111111111		

A blue 'Add Credit Society' button is located at the bottom of the table.

Fig 7.2.8 Listing Credit Societies

The screenshot shows a web browser window for 'MahCredit' at the URL 'localhost:8000/createtimewithbranch'. The title bar says 'MahCredit' and the address bar shows the URL. The main content area has a logo and the text 'MahCredit'. Below it is a 'Register' form with fields for Name, E-Mail Address, Select Branch (dropdown menu showing 'ai'), Personal Contact No., Password, and Confirm Password. A 'Register' button is at the bottom.

Name	<input type="text"/>
E-Mail Address	<input type="text"/>
Select Branch	<input type="text" value="ai"/>
Personal Contact No.	<input type="text"/>
Password	<input type="text"/>
Confirm Password	<input type="text"/>

Register

Fig 7.2.9 Branch Registration

The screenshot shows a web browser window for 'MahCredit' at the URL 'localhost:8000/createtimewithbranch'. The title bar says 'MahCredit' and the address bar shows the URL. The main content area has a logo and the text 'MahCredit'. Below it is a 'Register Customer' form with fields for First Name, Middle name, Last Name, Gender (Male), Marital Status (Single), Date of Birth (dd-mm-yyyy), Education (Matric), Occupation (Student), Caste (Hindu), Aadhar No., No. of Dependents, Religion (Hindu), Mobile Number, and Email ID. At the bottom, there are buttons for Mailing Address*, Present Address, Permanent Address, Present Address:, and Permanent Address: with Address input fields.

First Name*	<input type="text"/>	Middle name	<input type="text"/>	Last Name*	<input type="text"/>
Gender*	<input type="text" value="Male"/>	Marital Status*	<input type="text" value="Single"/>	Date of Birth*	<input type="text" value="dd-mm-yyyy"/>
Education	<input type="text" value="Matric"/>	Occupation*	<input type="text" value="Student"/>	Caste	<input type="text" value="Hindu"/>
Aadhar No.*	<input type="text"/>	No. of Dependents	<input type="text"/>	Religion*	<input type="text" value="Hindu"/>
Mobile Number*	<input type="text"/>	Email ID*	<input type="text"/>		

Mailing Address* Present Address Permanent Address

Present Address:
Address*

Permanent Address:
Address*

Fig 7.2.10 Customer Registration

The screenshot shows a web browser window with the title "MahCredit" at the top. Below the title is a logo consisting of a blue stylized 'M' with a crown-like shape above it, followed by the text "MahCredit". The main content area displays a table titled "Customer Listing". The table has columns: "Id", "Name", "Gender", "Edit", "Delete", and "Last Updated By". There are six rows of data:

Id	Name	Gender	Edit	Delete	Last Updated By
3	abcd abcd abcd	Male	<input checked="" type="checkbox"/>		Trunks
4	qwer qwer qwer	Male	<input checked="" type="checkbox"/>		Trunks
5	kasbcasc sascas csacasl	Male	<input checked="" type="checkbox"/>		Trunks
6	kavita ram patil	Female	<input checked="" type="checkbox"/>		Trunks
7	Ravi Chandra Ashwin	Male	<input checked="" type="checkbox"/>		Trunks
8	Mahindra Singh Dhoni	Male	<input checked="" type="checkbox"/>		Trunks

At the bottom of the table is a blue button labeled "Add Customer".

Fig 7.2.11 Customer Listing

The screenshot shows a web browser window with the title "MahCredit" at the top. Below the title is a logo consisting of a blue stylized 'M' with a crown-like shape above it, followed by the text "MahCredit". The main content area displays a table titled "Customer Resource Summary". The table has columns: "Asset No.", "Type", "Details", "Value", "Start Date", "End Date", "Verification", and "Delete". There is one row of data:

Asset No.	Type	Details	Value	Start Date	End Date	Verification	Delete
1	Home	566 sqft	3000000	2011-02-10	2024-02-10	Property papers	

Below the table is a section titled "Add Resource" with a form. The form has fields for "Type" (dropdown menu with "Car" selected), "Details" (text input), "Value" (text input), "Start Date" (date input), "End Date" (date input), and "Verification" (text input). At the bottom of the form is a blue button labeled "Add Resource".

Fig 7.2.12 Resource Summary Details

The screenshot shows a web browser window titled "MahCredit" with the URL "localhost:8000/showfamilyincome/8". The page displays a table of "Self-Employed Income" for four individuals. Below the table is a form for "Add Income".

Name	Type	Amount	Frequency	Start Date	End Date	Delete
Mahindra Singh Dhoni	Business	10000	Daily	2017-01-01	2018-03-01	
Sakshi Mahindra Dhoni	Business	10000	Quarterly	2017-01-01	2017-12-31	
Ziva Mahindra Dhoni						
Pan Singh Dhoni	Business	10000	Quarterly	2018-01-01	2018-03-01	

Add Income

Name	Type	Amount	Frequency	Start Date	End Date
Mahindra Singh Dhoni	Pension	<input type="text"/>	Weekly	<input type="text"/>	<input type="text"/>

Add Income

Fig 7.2.13 Income Details

The screenshot shows a web browser window titled "MahCredit" with the URL "localhost:8000/showfamily/8". The page displays a table of "Family Member" details for four individuals. Below the table is a form for "Add Family Member".

Id	Name	Relation	Education	Occupation	Date of Birth	Edit	Delete
6	Mahindra Singh Dhoni	Self	Graduate	Service	1984-06-12		
7	Sakshi Mahindra Dhoni	Wife	Post Graduate	Service	1986-07-17		
8	Ziva Mahindra Dhoni	Daughter	Illiterate	Student	2015-03-05		
9	Pan Singh Dhoni	Father	Matric	Unemployed	1954-08-20		

Add Family Member

<input type="text"/>	Mother	<input type="text"/>	Student	<input type="text"/>
----------------------	--------	----------------------	---------	----------------------

Fig 7.2.14 Family Details

The screenshot shows a web browser window for the MahCredit application. The URL in the address bar is `localhost:8000/showexpense/8`. The page title is "MahCredit". The main content area displays a table titled "Monthly Expense" with one row of data:

Name	Type	EMI	Start Date	End Date	Verification	Delete
Mahindra Singh Dhoni	Vehicle Loan	1000	2016-01-01	2018-06-30	Loan papers	

Below this table is a section titled "Add Expense" with input fields for Name, Type, EMI, Start Date, End Date, and Verification. The "Name" field contains "Mahindra Singh Dhoni", the "Type" field contains "Vehicle Loan", and the "EMI" field is empty. The "Start Date" and "End Date" fields are set to "dd-mm-yyyy". A blue "Add Expense" button is located below the input fields.

Fig 7.2.15 Expense Details

7.3 Reports generated / Tables obtained

The screenshot shows a web browser window for the MahCredit application. The URL in the address bar is `localhost:8000/showmss/8`. The page title is "MahCredit". The main content area displays three lines of summary information:

Average Income is Rs. 252500
Average Expense is Rs. 1000
Evaluation of Resources is Rs. 3000000

Fig 7.3.1 Expense Summary

	id [PK] integer	user_id integer	state_id integer	city_id integer	creditsoc_id integer	branch_id integer	level integer	parent_id integer	user_code character varying (191)
1	1	1	0	0	0	0	0	0	0 [null]
2	3	25	0	0	0	0	0	0	0 [null]
3	4	27	1	0	0	0	1	1	1 [null]
4	5	29	1	2	0	0	2	27	27 [null]
5	6	30	1	1	0	0	2	27	27 [null]
6	7	31	0	0	19	0	0	1	1 [null]
7	11	36	1	2	19	8	3	31	31 [null]
8	12	37	1	1	19	7	3	31	31 [null]
9	13	38	1	1	19	7	4	37	37 [null]
10	14	39	1	1	19	7	4	37	37 [null]
11	15	40	0	0	23	0	0	1	1 [null]
12	16	41	1	2	23	9	3	40	40 [null]

Fig 7.3.2 User Hierarchy in database

CHAPTER 8: CONCLUSION

8.1 Limitations

- This is a cloud based system, it is not necessary that you have an internet connection everytime and everywhere.
- The images and signature of customers are stored as byteA files, the images take time for retrieval.
- The other modules, Account and GL management and Account Charges Handling are still under development.
- It does not work for very large database and each credit society is allotted limited storage space.

8.2 Conclusion

This software will provide an affordable alternative to the small credit societies which cannot afford large upfront costs for customized applications. This system has the Indian credit society community as its target demographic. The system is developed using FOSS technologies to minimize the development costs. MahCredit is created using Laravel framework as it is the most popular framework with numerous functionalities. Laravel also provides smooth database connectivity and migration. We have chosen PostgreSQL because it is a FOSS and currently available data can be incorporated in it. When MahCredit is successfully implemented, it will provide an effective, easy-to-access solution for the credit societies (patpedhi) in India. This is basically a re-engineering project with added functionalities for betterment of the current services available.

8.3 Future Scope

- The income and expenditure details can be analysed to get the financial status of the person and thus take decision to sanction the loan.
- Currently, the system developed by us is a partial system which will be further developed by ‘Haplotech IT solutions LLP’ and deployed on cloud. The modules implemented by us can further be integrated with the other modules which are shown in the modular diagram to get the complete system.
- The current focus of the project is on the functionalities and not on presentation. The present system user interface can be modified and updated as per the industry requirements
- The database can further be optimised as per the requirement.
- There is a possibility of developing the system to accommodate national as well as

international level societies similar to Patpedhi as their basic functions will be similar to those implemented in our system.

- Incorporating options for different languages, currencies, countries, etc is also a distant scope.

References

- [1.] Present credit banking system software implemented on individual systems.
- [2.] Laravel 5.3 official documentation
- [3.] Basic concepts of credit, debit, loan disbursement , loan preclosure charges , general ledger (GL).
- [4.] <https://www.guru99.com/agile-scrum-extreme-testing.html>
- [5.] <https://drive.google.com/open?id=0Bx89CTEH7Np5enlwUTIxWHBwTzhoQmJGeWhfVFhQS1BWc0VR>
- [6.] Anurag Bejju, "Cloud Computing for Banking and Investment Services", ISSN: 2394-1553; Volume 1, Number 2; November, 2014 pp. 34-40
- [7.] Dr. Sheel Ghule, Ms. Rupali Chikhale, Mr. Kalpesh Parmar, "Cloud Computing in Banking Services", International Journal of Scientific and Research Publications, Volume 4, Issue 6, June 2014 1 ISSN 2250-3153
- [8.] Sultan Aldossary, William Allen,"Data Security, Privacy, Availability and Integrity in Cloud Computing: Issues and Current Solutions", International Journal of Advanced Computer Science and Applications, Vol. 7, No. 4, 2016
- [9.] Akash Jadhav , Rohit Jadhav & Shivganga Gavhane,"IMPLEMENTATION OF WEB APPLICATION USING LARAVEL FRAMEWORK" ,<http://> © International Journal of Engineering Sciences & Management Research, ISSN 2349-6193, Impact Factor: 2.805
- [10.] Chitrali Agre , "Implementation of a Cloud in Banking Sector", International Journal of Computer Science and Information Technology Research ISSN 2348-120X
- [11.] Dr. ARD Prasad,"Creation of Digital Libraries in Indian Languages Using Unicode", Workshop on Digital Libraries: Theory and Practice, March, 2003, DRTC, Bangalore
- [12.] Shweta Dinesh Bijwe , Prof. P. L. Ramteke, "Database in Cloud Computing Database-as-a Service (DBaaS) with its Challenges", ISSN 2320-088X IJCSMC, Vol. 4, Issue. 2, February 2015, pg.73 – 79
- [13.] Vishal V. Parkar , Prashant P. Shinde , Sanket C. Gadade , Prasad M. Shinde, "Utilization of Laravel Framework for Development of Web Based Recruitment Tool" IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727 PP 36-41

[14.] Copyright © 2015 Cloud Standards Customer Council, “Security for Cloud Computing Ten Steps to Ensure Success Version 2.0”

Appendix

6.1 Table of figures:

Figure Number	Name	Page Number
4.1	System Architecture	23
4.2	Block Diagram Of MahCredit	24
4.3	Modular Diagram of MahCredit	25
4.4.a(i)	ER Diagram-1	27
4.4.a(ii)	ER Diagram-2	28
4.4.a(iii)	ER Diagram-3	28
4.4.b	Flowchart of MahCredit	29
4.4.c(i)	Level-0 Data Flow Diagram	30
4.4.c(ii)	Level-1 Data Flow Diagram	31
4.4.c(iii)	Level-2 Data Flow Diagram 1	32
4.4.c(iv)	Level-2 Data Flow Diagram 2	32
4.4.d	Timeline for MahCredit	33
6.1.1	Trying SQL Injection	35
6.1.2	Failed SQL Injection	35
6.2.1	Broken access attack	36
6.2.2	Unsuccessful Broken Access	36
6.3.1	Unauthorized user access through URL	37
6.3.2	Unsuccessful unauthorized user access	37

6.4.1	XSS attack on name field	38
6.4.2	Failed XSS attack	38
6.5.1	Field Validation	39
6.5.2	Email Format Validation	39
6.5.3	Duplicate Email Notification	40
6.5.4	Phone Number Validation	40
6.5.5	Password Validation	41
7.2.1	Home Page	42
7.2.2	Login page	42
7.2.3	Admin Registration	43
7.2.4	State head Registration	43
7.2.5	City head Registration	44
7.2.6	Employee Registration	44
7.2.7	Credit Society registration	45
7.2.8	Listing Credit Societies	45
7.2.9	Branch Registration	46
7.2.10	Customer Registration	46
7.2.11	Customer Listing	47
7.2.12	Resource Summary Details	47
7.2.13	Income Details	48
7.2.14	Family Details	48
7.2.15	Expense Details	49
7.3.1	Expense Summary	49

6.2 Papers Published

Paper 1 (semester 7)

Cloud-based Core Banking Software For Small Finance Institutes

Manjiri Bhat, Romit Bonkar, Monal Daswani, Narayan Mundhara, Prof. Anjali Yeole

Department of Computer Science, Vivekanand Education Society's Institute Of Technology

Abstract: Credit societies are small institutions that work like banks for a limited number of people. These societies need a common platform for maintaining their databases and transaction. The objective is to create a cloud-based system for these credit societies so that they can have a centralized core banking application which can connect them with other small financial institutes. The project is developed with respect to the agile method. The software provides high-level security, is mobile and affordable.

Index Terms: Banking applications, Cloud-based system, cloud-computing, Credit Society, Indian-client base, SaaS

I. INTRODUCTION

As the world is moving towards

digitization, lethora of web-based solutions in the different aspects of life is emerging. Retail, finance, entertainment, etc. are all going digital. With digitization, comes the need for the development of a platform for these services. Particularly in the banking sector, a complex platform for the intricacies of the banking world is necessary. A new trend of using cloud-computing in the banking sector to reduce development costs is emerging.^[3]

Different solutions in the form of PaaS (platform as a service), SaaS (Software as a service) and IaaS (Infrastructure as a service) which are different models of cloud-based technology are being developed.

Credit societies are basically small institutes established by a group of people which work like banks for a limited number of people. The number of members can range from 20 to 2000. Still, it is a small number to actually call it a bank but larger than a single account. Lending money, buying shares in the societies, forming groups to save money, depositing money for safekeeping are some of the characteristics of a Credit society, also called as "Patpedi". These credit societies also need a digital platform for maintaining their databases and transactions. We aim to fulfill this need. credit societies also need a digital platform for maintaining their databases and transactions. We aim to fulfill this need.

This application is based on the concept of '**software as a service**', in which software and associated data are centrally hosted on the cloud(a remotely located server).One of the biggest constraints of small financial institutes to implement automated banking services is high investment involved in software development. One of the biggest selling point of 'Cloud-based credit banking software' is that it has the potential to reduce IT support costs by outsourcing hardware and software maintenance and support to the SaaS provider. Provision for network-based access to, and management of, banking software.It is based on a single-instance, multi-tenant(one to many) architectures including pricing, partnering and management characteristics.

II. OVERVIEW

Sometimes referred to as “on-demand software”^[4], SaaS is typically accessed by users from a thin-client via a web browser. SaaS has become a common delivery model for many business applications. The cloud database distribution and storage, as well as the system infrastructure is managed by the service providers and the users have access to the application and their database.

One of the biggest constraints on small finance institutes to get automated software is the large investment costs involved in product development and the following continual maintenance costs. The other hidden costs for individual project development are as shown in the Figure 1.

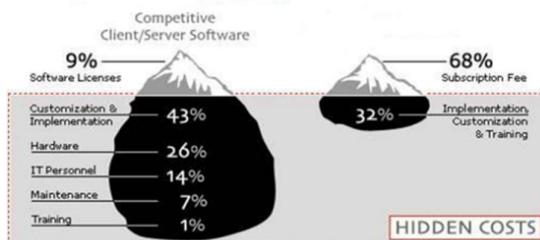


Figure 1. Hidden costs in existing system

We aim to overcome this drawback create a system which will be based on monthly billing and pay as per usage.

III. PROPOSED SYSTEM

“MahCredit” is a cloud-based credit banking software for small financial institutes. The proposed system architecture is shown in Figure 2.

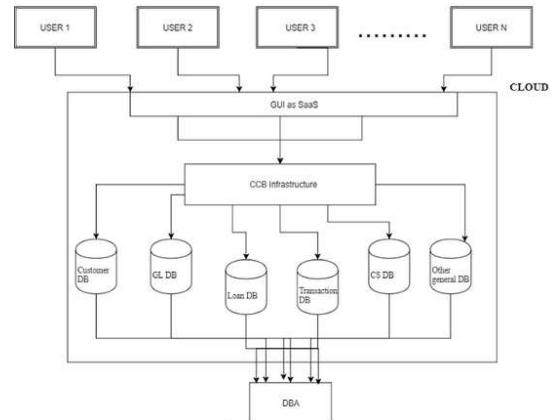


Figure 2. System architecture

A. *Users*: These are the owners or managers of the individual credit societies which avail the services provided by MahCredit.

B. *GUI for SaaS*: This is only part of the system in which user can enter features according to their requirements. GUI for SaaS is the graphical user interface provided by the system to its users. It is the topmost layer of the cloud-based system. It is necessary for user interaction. All the forms and tables are displayed or edited using the graphical user interface. It is the point of contact between the clients and the Credit Cloud banking system.

C. *CCB infrastructure*: It is basically the connection between the databases and the graphical user interface. Data processing, validation, authentication, inferencing is done in this layer of the architecture. This layer is vital to the proper working and functionality of the system.

D. *DB*: DB stands for databases created for different entities of the system. The first step when a user avails the services of MahCredit is his registration in the CS DB i.e. the credit society database.

The feature for Devanagari display will also be incorporated in the system to make it easier to enter the data for people who are not comfortable with English. Also, standards for security and privacy of data in the system will be maintained.

IV. METHODOLOGY

We aim to develop this system using Free and Open Source Softwares (Foss). For the development of the client-side infrastructure Laravel 5.3^[2], currently, the most popular framework, is chosen. The database design will be created using PostgreSQL.

ASCII cannot be used to represent more than one script, therefore to store the data in Devanagari script Unicode is necessary. For Unicode to work in a website, the browsers should be Unicode enabled. Unicode is the way to store Indian script data in SQL databases. UTF-8 and UTF-16 are the two versions of Unicode available.^[5] UTF-8 is required for the devanagari data storage for customer information in MahCredit.

V. CONCLUSION

This system has the Indian credit society community as its target demographic. It is developed using open source technologies which were absent in the previously studied system. MahCredit will be developed using Laravel framework as it is the most popular framework with a lot of functionalities. We have chosen PostgreSQL because it is a FOSS. MahCredit with a lot of functionalities. We have chosen PostgreSQL because it is a FOSS. MahCredit once implemented will provide an effective online solution for the credit societies in India.

REFERENCES

- [1.] Present credit banking system software implemented on individual systems.
- [2.] Laravel 5.3 official documentation
- [3.] Dr. Sheel Ghule, Ms. Rupali Chikhale, Mr. Kalpesh Parmar, "Cloud Computing in Banking Services", International Journal of Scientific and Research Publications, Volume 4, Issue 6, June 2014 1 ISSN 2250-3153
- [4.] Chitrali Agre,"Implementation of a Cloud in Banking Sector", International Journal of Computer Science and Information Technology Research ISSN 2348-120X
- [5.] Dr. ARD Prasad,"Creation of Digital Libraries in Indian Languages Using Unicode", Workshop on Digital Libraries: Theory and Practice, March 2003, DRTC, Bangalore

Authors

First Author- Manjiri Bhat, Engineering Final Year, Vivekanand Education Society's Institute Of Technology, manjiri.bhat@ves.ac.in

Second Author- Romit Bonkar, Engineering Final Year, Vivekanand Education Society's Institute Of Technology, romit.bonkar@ves.ac.in

Third Author- Monal Daswani, Engineering Final Year, Vivekanand Education Society's Institute Of Technology, monal.daswani@ves.ac.in

Forth Author- Narayan Mundhara, Engineering Final Year, Vivekanand Education Society's Institute Of Technology, narayan.mundhara@ves.ac.in

Correspondance Author- Anjali Yeole, Professor, Vivekanand Education Society's Institute Of Technology, anjali.yeole@ves.ac.in

Core Banking Software Based On Cloud For Credit Societies

Manjiri Bhat, Romit Bonkar, Monal Daswani, Narayan Mundhara,

Mrs. Anjali Yeole

(Department of Computer Science, Vivekanand Education Society's Institute Of Technology, India)

Abstract: Credit societies are small establishments that are similar to banks but for a restricted range of users. These societies need a standard platform for maintaining their data and transactions. The aim is to create a cloud-based system for the credit societies such that they will have a core banking application which can connect them with other small financial establishments. The development of the project is relevant to agile methodology. The software will offer high-level security and will be mobile and cheap.

Key words: core banking, Cloud-based system, cloud-computing, Credit Society, Indian-client base, SaaS

I. INTRODUCTION

As the world is moving towards digitization, an assorted set of web-based solutions in the different aspects of life are emerging. Retail, finance, entertainment, etc. are all going digital. With digitization, there is a necessity for the development of a platform for these services. Notably within the banking sector, a complex platform for the intricacies of the banking world is required. A recent trend of using cloud-computing in the banking sector to reduce development costs is emerging.^[3]

Various solutions in the form of PaaS (Platform as a service), SaaS (Software as a service) and IaaS (Infrastructure as a service), which are the three models of cloud-based technology, are being developed.

Credit societies are primarily small institutions established by a group of people which work like banks for a limited number of users. The number of members may be in the range of 20 and 2000. Still, it is a small range to actually form a bank but larger than just one account. Lending money, buying shares within the societies, forming groups to save money, depositing money for safekeeping are some of the features of a Credit society referred to as "Patpedhi". These establishments need a digital platform for maintaining their database and transactions. Our aim is to fulfil this need.

This application is based on the idea of '**Software as a service**', in which software and associated data are centrally hosted on the cloud (a remotely located server). One of the most important constraints of small financial institutions to implement automated banking services is; huge investment involved in software package development. One of the biggest selling point of 'Cloud-based credit banking software' is that it has the potential to reduce IT support costs by outsourcing hardware and software maintenance and support to the SaaS provider. It will provide network-based access to, and management of, the banking software. It is based on a single-instance multi-tenant (one to many) architecture with a multilevel and multilateral user hierarchy, including valuation, partnering and management characteristics.

II. OVERVIEW

Sometimes known as "on-demand software"^[4], SaaS is usually accessed by users from a thin-client via a web browser. SaaS has become a standard delivery model for several business applications. The cloud data distribution and storage, as well as the system infrastructure and user hierarchy are managed by the service providers and the users have access to the application and only their database.

One of the restrictions on small financial institutions to get a digital platform is the large initial cost involved in product development and the successive maintenance costs. The invisible investments required for individual project development are as shown in the Fig.1.

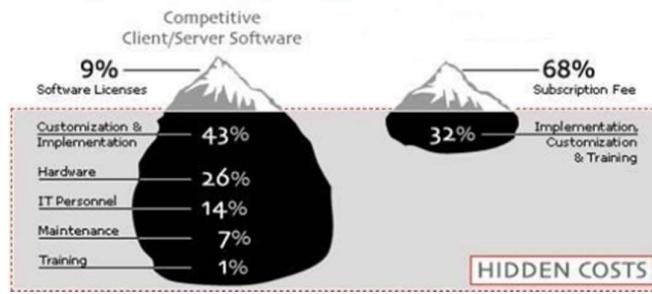


Fig.1. Hidden costs in existing system

Our target is to overcome this drawback by creating a system which will be based on monthly billing and pay as per usage.

III. PROPOSED SYSTEM

"MahCredit" is a cloud-based credit banking software for small financial institutions. The proposed system architecture is shown in Fig. 2.

The feature for Devanagari display will also be incorporated in the system to make it easier to enter the data for people who are not comfortable with English. Also, standards for security and privacy of data in the system will be maintained. The privacy and privileges of different users will be maintained by the user hierarchy. The users will have access to separate databases according to their level in the hierarchy. For example, state level users will have a more aggregated view of the transactions while the accountant and branch manager can view and edit the day-to-day transactions.

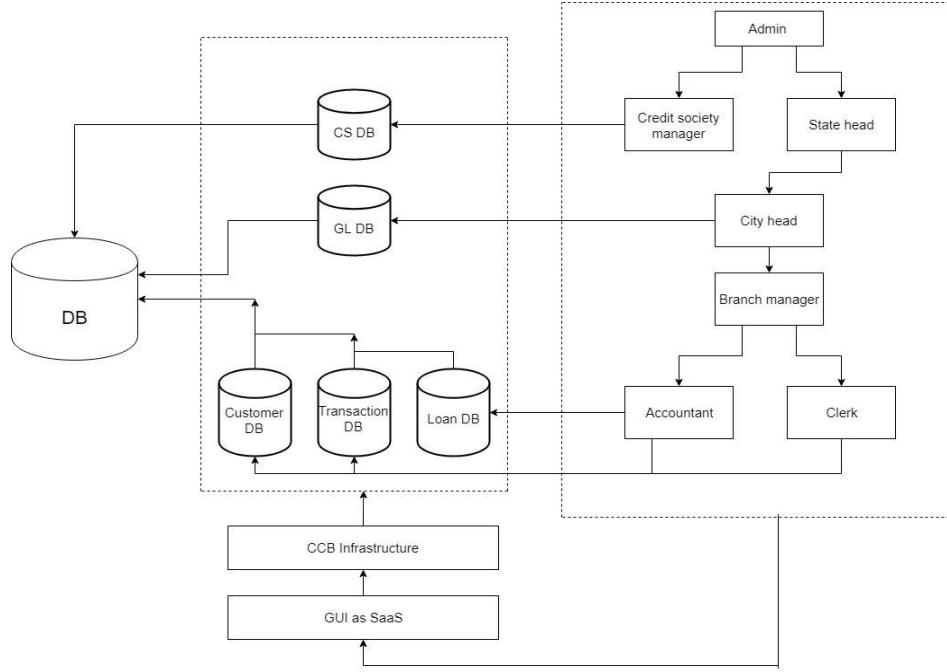


Fig.2. System architecture

A. *Users:* The administrators are level 0 users who have the highest privileges. The owners or managers of the individual credit societies are level 2 users which avail the services provided by MahCredit. The accountants and clerks are the level 4 users. There is also state level (level 1), district level and city level abstraction for the given user hierarchy. Accountants and clerk are the users on the lowest level of users who will take care of the day to day transactions and updates. The user hierarchy is multi-level as well as multilateral as shown in Fig.2

B. *GUI for SaaS:* This is only part of the system in which user can enter features according to their requirements. GUI for SaaS is the graphical user interface provided by the system to its users. It is the topmost layer of the cloud-based system. It is necessary for user interaction. All the forms and tables are displayed or edited using the graphical user interface.

C. *CCB infrastructure:* It is basically the connection between the database and the graphical user interface. Data processing, validation, authentication and conclusions are drawn in this layer of the architecture. This layer is vital to the proper working and functionality of the system.

D. *DB:* DB stands for databases created for different entities of the system. The first step when a user avails the services of MahCredit is his registration in the CS DB i.e. the credit society database.

IV. METHODOLOGY

We aim to develop this system using Free and Open Source Software (Foss). For the development of the client-side infrastructure Laravel 5.3^[2], currently the most popular framework, is chosen. The database will be created and maintained using PostgreSQL.

ASCII cannot be used to represent more than one script, therefore to store the data in Devanagari script Unicode is necessary. For Unicode to work in a website, the browsers should be Unicode enabled. Unicode is the way to store Indian script data in SQL databases. UTF-8 and UTF-16 are the two versions of Unicode

available.^[5] UTF-8 is required for the Devanagari data storage for customer information in MahCredit. The images in the database will be stored in ByteA format which is a standard in PostgreSQL. ByteA is equivalent to blob data representation. Maintaining consistency will also be important and complex as there is a lot of dependency and connectivity amongst different tables in the database. Laravel documentation is limited as it is a comparatively new framework, so it may present some difficulties in implementing certain features. We will be using Bootstrap for layout design as the focus is on functionalities and not on GUI design. We will be developing the system module-wise which will later be integrated to form the fully functional software.

V. CONCLUSION

This system has the Indian credit society community as its target demographic. The system will be developed using FOSS technologies to minimize the development costs. MahCredit will be created using Laravel framework as it is the most popular framework with numerous functionalities. Laravel also provides smooth database connectivity and migration. We have chosen PostgreSQL because it is a FOSS and currently available data can be incorporated in it. When MahCredit is successfully implemented, it will provide an effective, easy-to-access solution for the credit societies (patpedhi) in India.

REFERENCES

- [1.] Present credit banking system software implemented on individual systems.
- [2.] Laravel 5.3 official documentation
- [3.] Dr. Sheel Ghule, Ms. Rupali Chikhale, Mr. Kalpesh Parmar, "Cloud Computing in Banking Services", *International Journal of Scientific and Research Publications, Volume 4, Issue 6, June 2014 1 ISSN 2250-3153*
- [4.] Chitrali Agre, "Implementation of a Cloud in Banking Sector", *International Journal of Computer Science and Information Technology Research*, ISSN 2348-120X
- [5.] Dr. ARD Prasad, "Creation of Digital Libraries in Indian Languages Using Unicode", *Workshop on Digital Libraries: Theory and Practice, March 2003, DRTC, Bangalore*