

1. Introduction

1.1 Objective & Scope of the Project

A bank is a financial institution licensed by a government. Its primary activities include providing financial services to customers while enriching its investors. Many financial services were allowed over time. Banks are important players in financial markets and offer financial services such as investment funds.

Under English common law a banker is defined as a person who carries on the business of banking, which is specified as

- Conducting current accounts for his customers
- Paying cheques and others drawn on him
- Collecting cheques for his customers

With years, banks are also adding services to their customers. The Indian banking industry is passing through a phase of customers market. The customers have more choices in choosing their banks. A competition has been established within the banks operating in India.

With stiff competition and advancement of technology, the service provided by banks has become more easy and convenient. The past days are witness to an hour wait before withdrawing cash from accounts or a cheque from north of the country being cleared in one month in the south.

1.2 Background Details

Existing System:

Existing system is running as semi automated system, the following problems we are faced through existing system.

Data Duplication:

The same data gets repeated over and over since the workers find it hard to keep track of the documents, information and transactions

Lack of security:

Since data is stored in filing cabinets it is freely available to anyone. if information falls into the wrong hands it can be used against the company and customers and can blackmail them.

Common errors:

When entering data customers might have accidentally switched details and data since it is hand written.

Inconsistency of data:

There will be unavailability for future use, since data might get misplaced during manual filing. So data won't be preserved properly for future use.

Repetition of work:

If there are any changes to be made, the data will have to be entered again. At times the worker would forget to make the changes or forget that they had already altered it and might redo it again, its again time consuming.

Too much paper work:

Since everything and every detail are written down manually in paper there will be too much paper work!

Space consuming:

Since the data and paper is stored in filing cabinets it consumes too much place, as the amount of work done on paper increases the filing cabinets too increases.

Slow retrieval of data:

The information of customers and details are stored in different parts of the site and so takes a long time to retrieve the data. It takes a long time to find the information about a relevant person. In case of a Delivery, the delivery will be held back. This results in a sharp drop in sales, unhappy customers and a bad impression on the banks.

Proposed System

While most of us have heard about online banking services, more than a majority of us have probably not even tried it out yet. It could possibly be because we are more comfortable working with real people; paper and money instead of its virtual counterpart, as performing transactions over the Internet can be very impersonal. Whatever may be the reason; there are a number of advantages and disadvantages to online banking services.

First let's start off with the advantages of online banking. First and foremost, online banking is very, very, very convenient. It will allow you to pay your bills and make transactions anytime during the day and the week. The bank will never close because you can access it through your laptop or computer. So, no matter in which country you are anywhere in the world, you can go online and handle your finances.

Secondly, online banking is very fast, effective and efficient. Over the Internet, you can make transactions that are typically executed and performed at a much faster pace than at ATM's. Online banking services also give you the option of handling several different bank accounts from one site itself.

1.3 Definition of the Problem

The basic purpose of this project is to change the existing manual system of recording the information in records into computerized so that it is very easy and efficient to maintain the records and facilitates the user for future references and forecasting. However, with the advent of the Internet, it is possible to enable generating results and HSBS

2. System Requirements Specification

2.1 Functional Requirements

This application consists following modules

STUDY OF THE SYSTEM

In the flexibility of uses the interface has been developed a graphics concepts in mind, associated through a browser interface. The GUI's at the top level has been categorized as follows

1. Administrative User Interface Design
2. The Operational and Generic User Interface Design

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The Interface helps the administration with all the transactional states like data insertion, data deletion, and data updating along with executive data search capabilities.

The operational and generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.

NUMBER OF MODULES

The system after careful analysis has been identified to be presented with the following modules:

The Modules involved are

1. Bank Administrator
2. Bank Manager
3. Customers
4. Search
5. Reports
6. Authentication

Bank Administrator

Administrator is a super user treated as owner of this site. He can have all the privileges. The admin services are some reusable components for the MAS and common services like user authentication, user service role mapping and user management.

Administration can keep track the following tasks:

Admin can Add new Managers to the branch and he can edit the Managers.

- Add Branches: Administrator adds new branches to the bank and admin can change and update the branch details which are already exist.
- Add New Loans: Administrator can add new types of loans which are required for the situation and he can fix the maximum amount for that particular loan type.
- Add Areas: Administrator adds new areas. The bank supposed to start a new branch in other country or in the same country he can add country and location where he wants to start a branch.

Bank Manager

The working employee on a Bank with some special powers is called Bank Manager

. They can perform different functionality on the particular bank branch.

- Manager can receive the new account request and he can view the old accounts under his bank branch.

Manager can manage the bank users request for new loans, request for new card (Credit/Debit) and requests for new check books. Manager can view the transaction details like username, transaction date and amount Deposited etc..

Search

Various types of search like Account details, Transaction details, card details and loan details and these search is maintained by the user.

Reports

Different kind of reports is generated by the system. Add Managers. Admin can Add new Managers to the branch and he can edit the Managers.

- Add Branches

Administrator adds new branches to the bank and admin can change and update the branch details which are already exist.

- Add New Loans

Administrator can add new types of loans which are required for the situation and he can fix the maximum amount for that particular loan type.

- Add Areas

Administrator adds new areas. The bank supposed to start a new branch in other country or in the same country he can add country and location where he wants to start a branch.

Bank Manager

The working employee on a Bank with some special powers is called Bank Manager. They can perform different functionality on the particular bank branch.

- Manager can receive the new account request and he can view the old accounts under his bank branch.
- Manager can manage the bank users request for new loans, request for new card(Credit/Debit) and requests for new check books.
- Manager can view the transaction details like username, transaction date and amount Deposited etc..

Users

User can change his personal information and he can change the nominee details of his account. User can send request for new loans, new cards like credit and debit cards, new check book and he can request for reactivation of his account. User can transfer the amount to another account. User can pay the taxes and he can pay various types of bills.

Search

Various types of search like Account details, Transaction details, card details and loan details and these search is maintained by the user.

Reports

Different kind of reports is generated by the system.

Inputs and outputs

The major inputs and outputs and major functions of the system are follows:

Inputs:

- Admin/Manager enters user id and password for login.
- User enters his user id and password for login.
- New user gives his completed personnel, address and phone details for registration.
- Admin gives different kind of user information for search the user data.
- Administrator giving information to generate various kinds of reports.

Outputs:

- Admin can have his own home page.
- Users enter their own home page.
- The new user's data will be stored in the centralized database.
- Admin get the search details of different criteria.
- Different kind of reports is generated by administrator.

Input Design:

Input design is a part of overall system design. The main objective during the input design as given below:

- To produce cost-effective method of input
- To achieve the highest possible level of accuracy. To ensure that the input is acceptable and understood by the user.

Input States:

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion

- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

Input Types:

It is necessary to determine the various types of input. Inputs can be categorized as follows:

- External Inputs which are prime inputs for the system.
- Internal Inputs, which are user communications with the systems.
- Operational, which are computer department's communications to the system?
- Interactive, which are inputs entered during a dialogue.

Input Media:

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to:

- Type of Input
- Flexibility of Format
- Speed
- Accuracy
- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use
- Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As input data is to be directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

Output Design:

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization,.
- Internal Outputs whose destination is within organization and they are the User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly with User Interface.

Output Definition:

The outputs should be defined in terms of the following points:

- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

2.2 Non functional Requirements

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility
- Economical Feasibility

Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hardware requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment

and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

Economic Feasibility

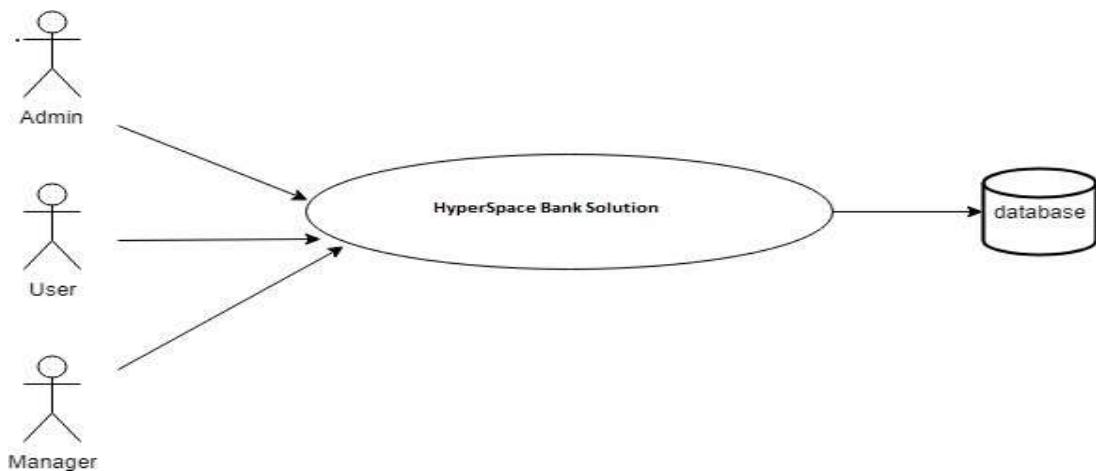
A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

3. Requirement Analysis Document

3.1 System Models

3.1.1 Identifying Actors

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap and synergy with the disciplines of systems analysis, systems architecture and systems engineering.



3.1.2 Identifying Scenarios

There are different scenarios used basing on the working procedure of the login user, the main scenario of the user have to register the customer, update the details about the customer, updating the IFSC numbers, updating online details and accessing the database.

Scenario is nothing but it is an instance of a use case it explains the single feature of a system.

Scenario for Registering Details:

Scenario name:	Registering details
Use case name:	Hyperspace Bank Solutions
Participating actors:	Admin, User
Entry condition:	Admin, User has to Registered their Profile Details
Flow of events:	<ul style="list-style-type: none">• If the details are valid account will be created.• Admin,User has to Login and view Profiles.

3.2 Use case model

3.2.1 Identifying Use Cases

Use case Diagrams represent the functionality of the system from a customer point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer ...etc., or another system like central database.

Use Case for Add User, and User details:

Use case:	Add user
Participating actors:	Admin
Precondition:	Admin must login as an authorized person.
Flow of events :	<ul style="list-style-type: none">• First Admin has to login as an authorized person.• Admin has to register the User, and User details.• After the completion of registration admin has to log out from the system.

Use Case for Remove User and User Details:

Use case:	Remove User, account and Details
Participating actors:	Admin
Precondition:	Admin must login as an authorized person.
Flow of events :	<ul style="list-style-type: none">• Admin has to remove the particular user and user Details.• After the completion of Deletion user has to log out from the system.

Use Case for View User details:

Use case:	View Applicants
Participating actors:	Admin
Precondition:	Admin must login as an authorized person.
Flow of events :	<ul style="list-style-type: none">• Admin has to view the applicant details for user.• Admin can able to View and modify the all the details of user• After the completion of registration Admin has to log out from the system.

3.2.2 Use case Diagrams

Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use case focuses on the behavior of the system from an external point. The actors outside the boundary of the system. A use case contains all the events that can occur between an actor and a set of scenarios that explain the interactions as sequence of happenings.

System use case Diagram:

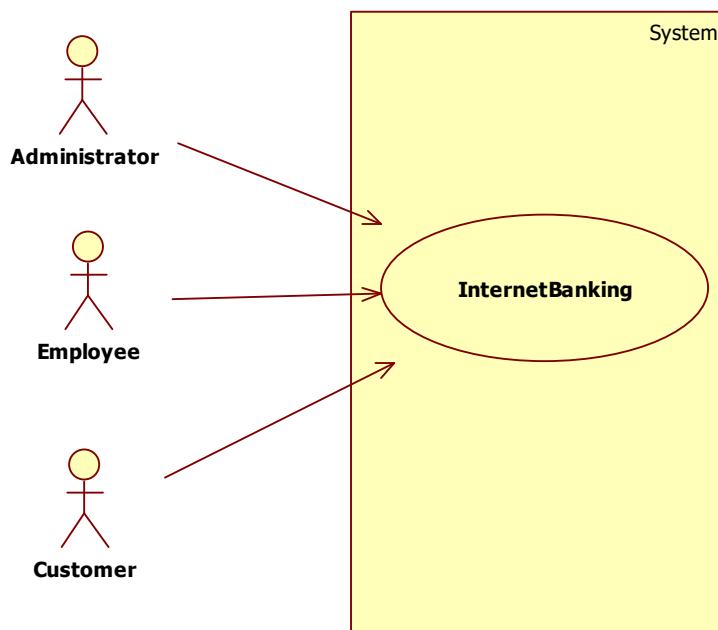


Fig 3.2.2(a) Use case diagram

Administrator Use case Diagram:

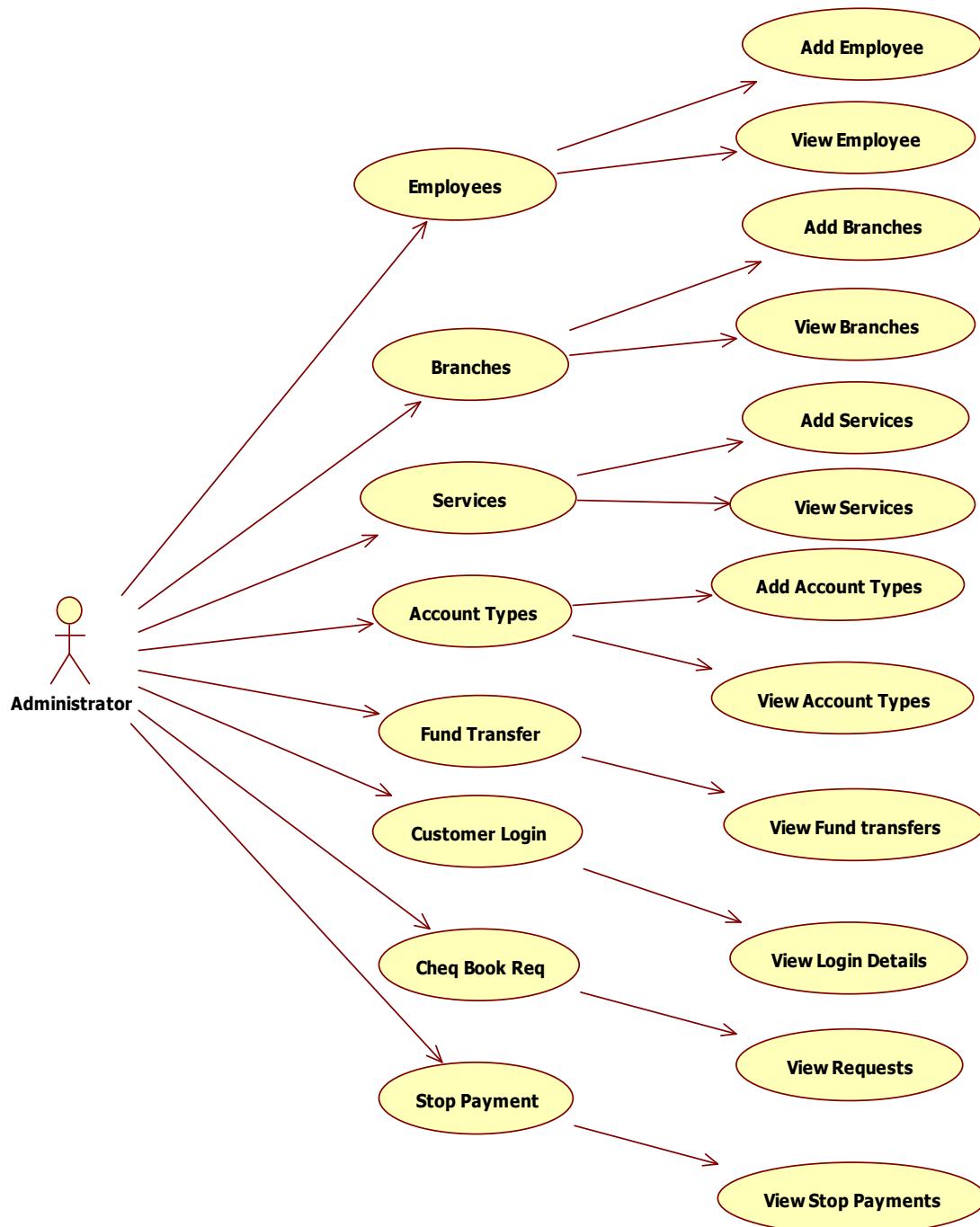


Fig 3.2.2(a) Usecase diagram for Admin

Employee Use case Diagram



Fig 3.2.2(b) Usecase diagram for employee

Customer Use Case Diagram:

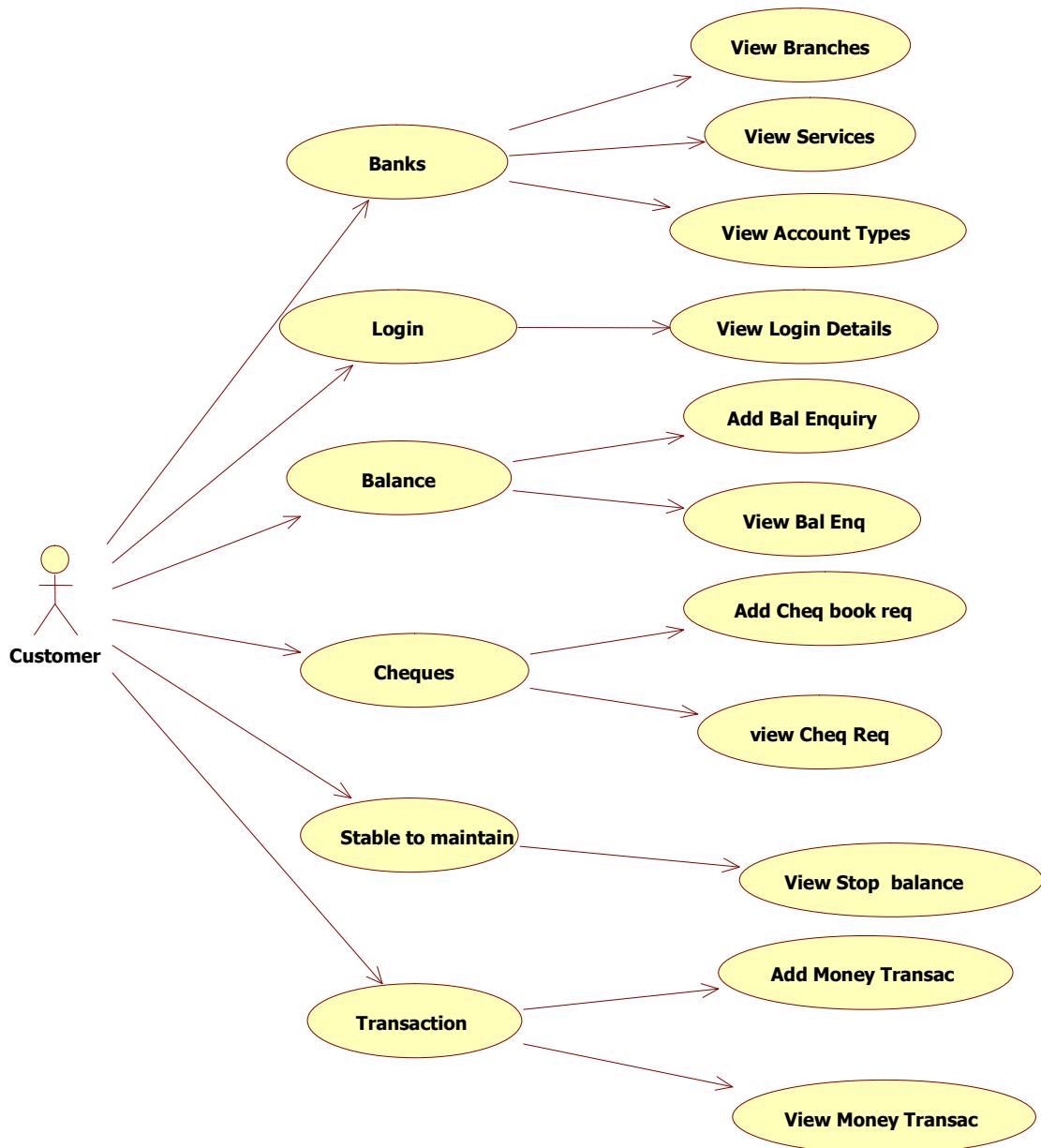


Fig 3.2.2(c) Usecase diagram for customer

3.3 Object Model

3.3.1 Data Dictionary

After carefully understanding the requirements of the client the the entire data storage requirements are divided into tables. The below tables are normalized to avoid any anomalies during the course of data entry.

	bid	bname	ifscno	bpassword	village	mondal	district	state	country
<input type="checkbox"/>	3	sbh-kondamallepally	SBHY0028683	123456	kondamallepally	devarakonda	nalgonda	ap	India
<input type="checkbox"/>	4	Devarakonda	SBHY0013129	shankar	devarakonda	devarakonda	nalgonda	ap	India
<input type="checkbox"/>	5	sbh-nalgonda	SBHY0041585	523952	nalgonda	nalgonda	nalgonda	ap	India
<input type="checkbox"/>	6	b1		122159	b1	b1	b1	b1	India
<input type="checkbox"/>	7	hdfc	SBHY0090666	861406	hyd	hyd	hyd	hyd	India
<input type="checkbox"/>	8	sbi	SBHY0066454	516045	hyd	hyd	hyd	njk	jhk
<input type="checkbox"/>	9	gig	SBHY0049023	617252	bj	bj	hj	hj	fdg
<input type="checkbox"/>	10	gjhh	SBHY0011993	321918	hhhh	hhh	hh	h	hh
<input type="checkbox"/>	11	jhbj	SBHY0045557	247266	kjkj	jnjkh	jkjhk	jkhh	jkh
<input type="checkbox"/>	12	hello	SBHY0021004	756834	hjh	jh	jh	jh	h
<input type="checkbox"/>	13	sbihsa	SBHY0018872	855133	hsa	hsa	hsa	hsa	India
<input type="checkbox"/>	14	sr nagar	SBHY0021769	716091	sr nagar	ameerpet	hyderabad	telangana	india

	sno	first_name	last_name	account_number	transfer_amount	received_amount	total_amount	year	month	happen_date	unique_id
<input type="checkbox"/>	1	Madan	Sapkota	1002003302	200	0	4800	2010	2	2010-02-10 03:37:48	4489
<input type="checkbox"/>	2	Jagadish	Kumar	1002003301	0	200	10200	2010	2	2010-02-10 03:37:48	4490
<input type="checkbox"/>	3	Madan	Sapkota	1002003302	300	0	4500	2010	2	2010-02-10 03:38:01	4489
<input type="checkbox"/>	4	Jagadish	Kumar	1002003301	0	300	10500	2010	2	2010-02-10 03:38:01	4490
<input type="checkbox"/>	5	Jagadish	Kumar	1002003301	700	0	9800	2010	2	2010-02-10 04:08:18	4490
<input type="checkbox"/>	6	Madan	Sapkota	1002003302	0	700	5200	2010	2	2010-02-10 04:08:18	4489
<input type="checkbox"/>	7	Jagadish	Kumar	1002003301	300	0	9500	2010	2	2010-02-10 04:08:38	4490
<input type="checkbox"/>	8	Madan	Sapkota	1002003302	0	300	5500	2010	2	2010-02-10 04:08:38	4489
<input type="checkbox"/>	9	Madan	Sapkota	1002003301	0	0	5500	2010	2	2010-02-10 04:40:32	4489
<input type="checkbox"/>	10	Madan	Sapkota	1002003301	0	0	0	2010	2	2010-02-10 04:40:32	4489
<input type="checkbox"/>	11	something	something	1002003301	500	0	9500	2010	2	2010-02-10 07:19:43	4493
<input type="checkbox"/>	12	Madan	Sapkota	1002003305	0	500	6000	2010	2	2010-02-10 07:19:43	4489
<input type="checkbox"/>	13	Madan	Sapkota	1002003302	4586	0	5924	2010	2	2010-02-11 00:33:21	4489
<input type="checkbox"/>	14	Jagadish	Kumar	1002003301	0	4586	14086	2010	2	2010-02-11 00:33:21	4490
<input type="checkbox"/>	15	New	Account	1002003304	1500	0	3500	2014	3	2014-03-13 12:24:56	4494
<input type="checkbox"/>	17	New	Account	1002003304	1500	0	3500	2014	3	2014-03-13 12:31:40	4494
<input type="checkbox"/>	32	shankar	shiva	1002003302	300	0	9000	2014	3	2014-03-29 12:35:46	4492
<input type="checkbox"/>	19	raju	kumar23	1002003301	0	50	50	2014	3	2014-03-17 17:03:47	4490
<input type="checkbox"/>	20	test	test	1002003302	10	0	40	2014	3	2014-03-17 17:04:18	4489

+ Options

	<input type="button" value="← T →"/>		mid	managername	bankname	village	mondal	district	state	country
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	venkatapapaiah	9	nalgonda	nalgonda	ap	India
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	shankar	12	hyd	hyd	ap	India
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4	babu	3	hyd	hyd	ap	India
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	5	joy	0	hsa	hsa	hsa	India
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	6	reo	0	hsa	hh	h	India
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	7	hh	0	h	h	h	in
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	8	k	0	k	k	k	India
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	9	aA	0	a	a	a	India
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	10	hello	0	hyd	hyd	hyd	india
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	11	abc	0	kj	jj	j	j
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12	babu	0	hg	h	h	India
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	13	naresh	14	sr nagar	ameerpet	hyderabad	telangana
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	14	anjali	0	akvd	akvd	west godavari	andhrapradesh

+ Options

	<input type="button" value="← T →"/>		sno	first_name	last_name	account_number	password	email	mobile	adhar_no	country	state	address	pincode	t	
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	reo	1002003301	123456	reo@gmail.com	r	r	r	r	r	0		
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	r	1002003302	193075	r@gmail.com	r	r	r	r	r	0		
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	rr	1002003303	234154	rr@gmail.com	r	r	r	r	r	0		
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4	a	1002003304	698449	a@a.com	a	a	a	a	a	0		
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	13	b	1002003305	804710	b@b.com	b	b	b	b	b	0		
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	14	c	1002003306	969465	c@c.com	c	c	c	c	c	0		
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	15	d	1002003307	320892	d@d.com	d	d	d	d	d	0		
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	16	e	1002003308	755169	e@e.com	e	e	e	e	e	0		
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	17	f	1002003309	519969	f@f.com	f	f	f	f	f	0		
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	18	g	1002003310	258112	g@g.com	g	g	g	g	g	0		
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	19	h	1002003311	279727	h@h.com	h	h	h	h	h	0		
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	20	prashanthi	nadimpalli	1002003312	830729	prashanthinadimpalli777@gmail.com	9494335277	784512784512	india	Andhrapradesh	blim	534202	
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	21	shanthi	varma	1002003313	452202	prashanthi7488@gmail.com	9394233337	239365216688	india	Andhrapradesh	blimavaram	534202	

+ Options

	<input type="button" value="← T →"/>		sno	account_number	quantity	request	response
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	1002003304	1	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	1002003301	1	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	1002003302	1	1

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

3.3.2 Class diagrams

The object model, represented in UML with class diagrams, describes the structure of system in terms of objects, attributes associations. The UML enables us to model the classes in a system and their relationships via class diagrams. In a class diagram, each class is modeled as a rectangle. The middle part contains the class's attributes. The bottom contains the classes operations. The UML defines object diagrams which are similar to class diagrams except that they model objects and links-links are relationships between objects. Object diagrams present a snapshot of the structure while the system is running this provides information about which objects are participating in the system at a specific point in time.

Class diagram for Hyperspace Bank Solutions

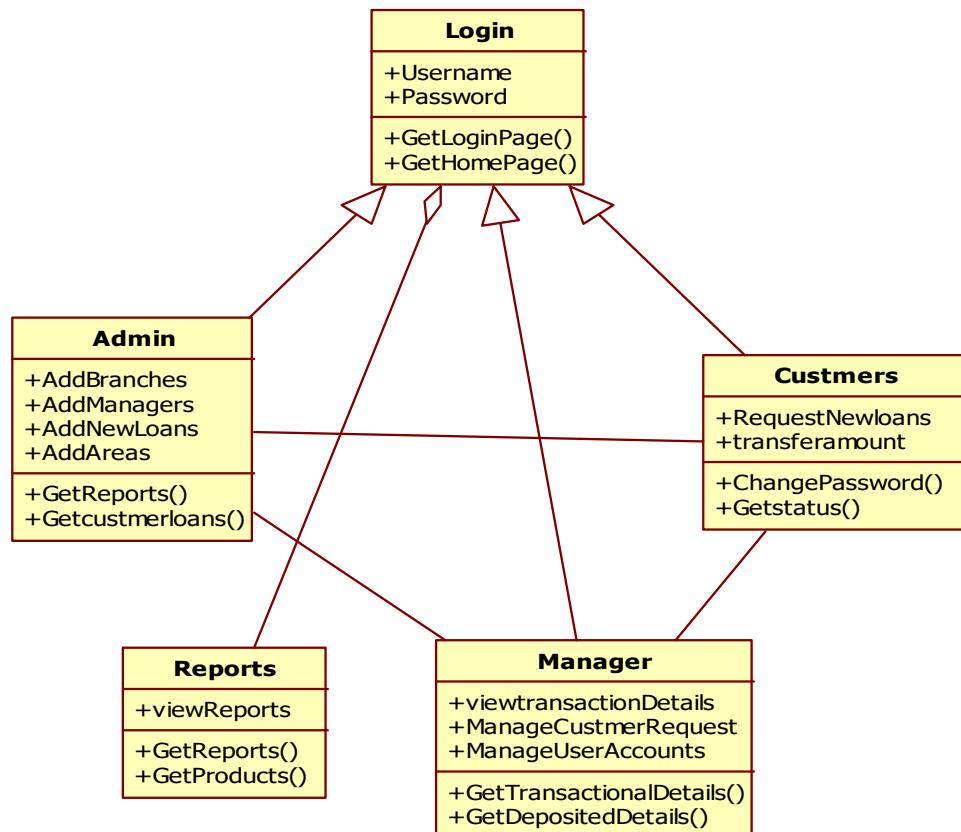


Fig 3.2.2(c) Class diagram for HSBS

3.4 Dynamic models

3.4.1 Sequence diagrams

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically.

Administrator Sequence Diagram:

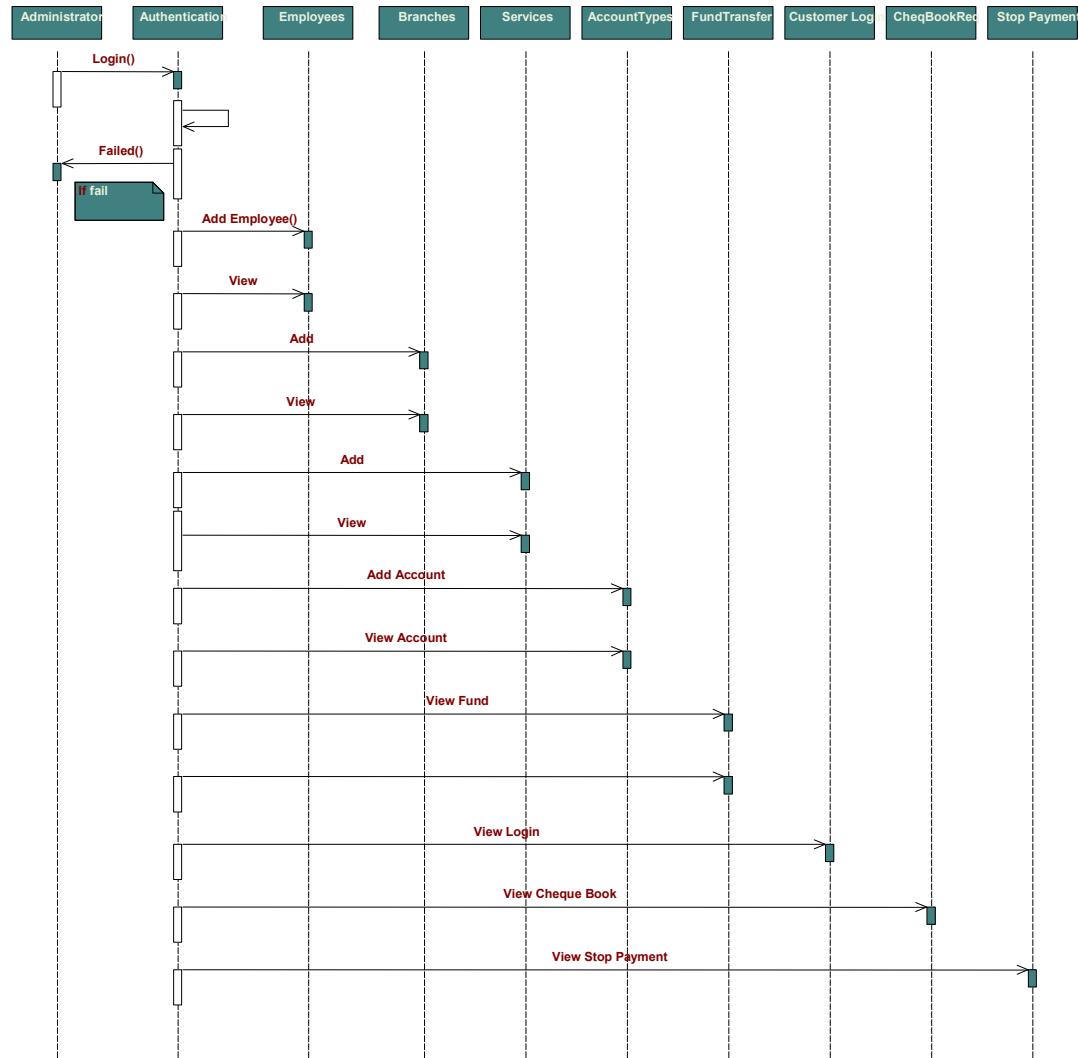


Fig 3.4.1(a) Sequence diagram for Admin

Employee Sequence Diagram

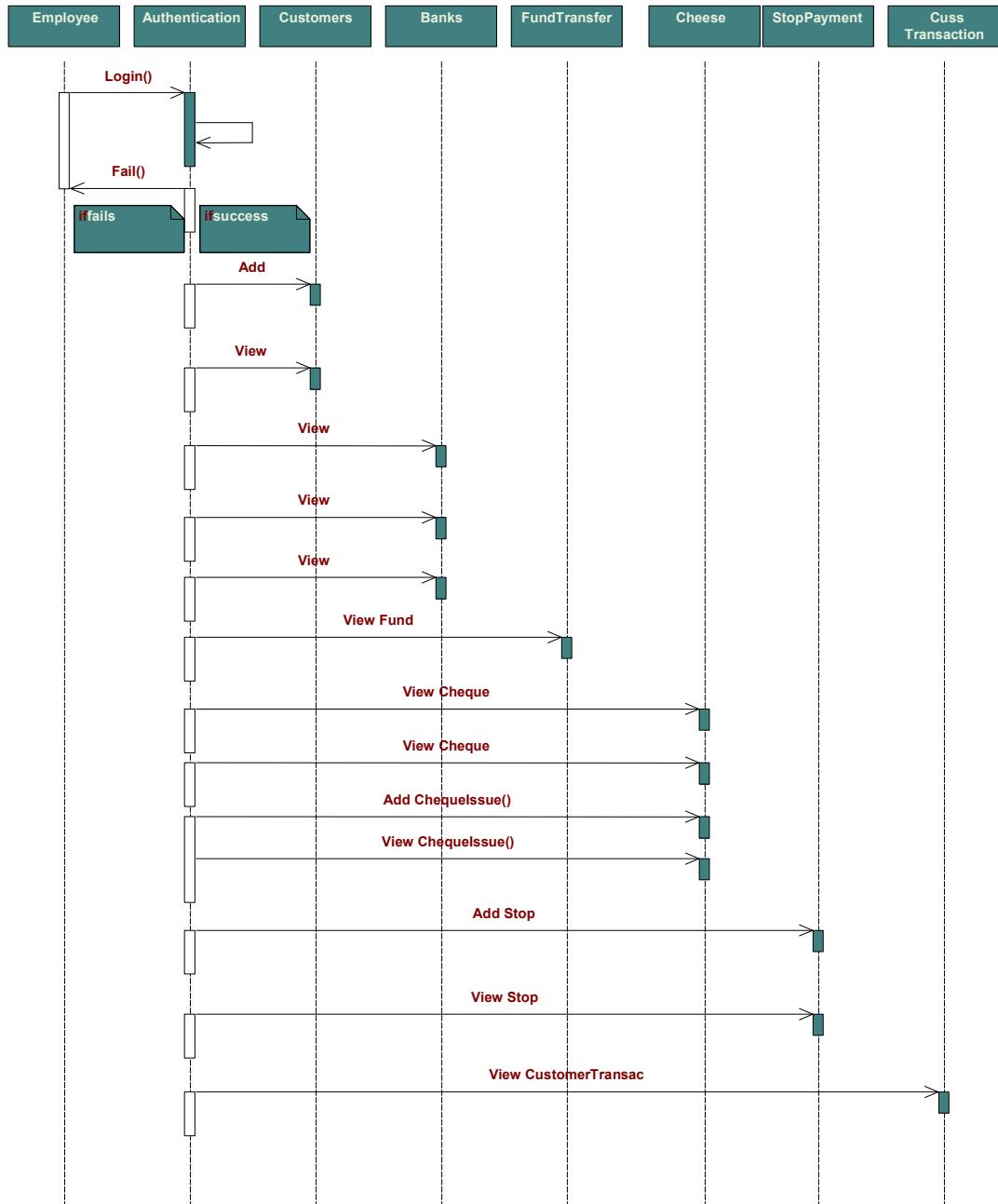


Fig 3.4.1(b) Sequence diagram for employee

Customer Sequence Diagram:

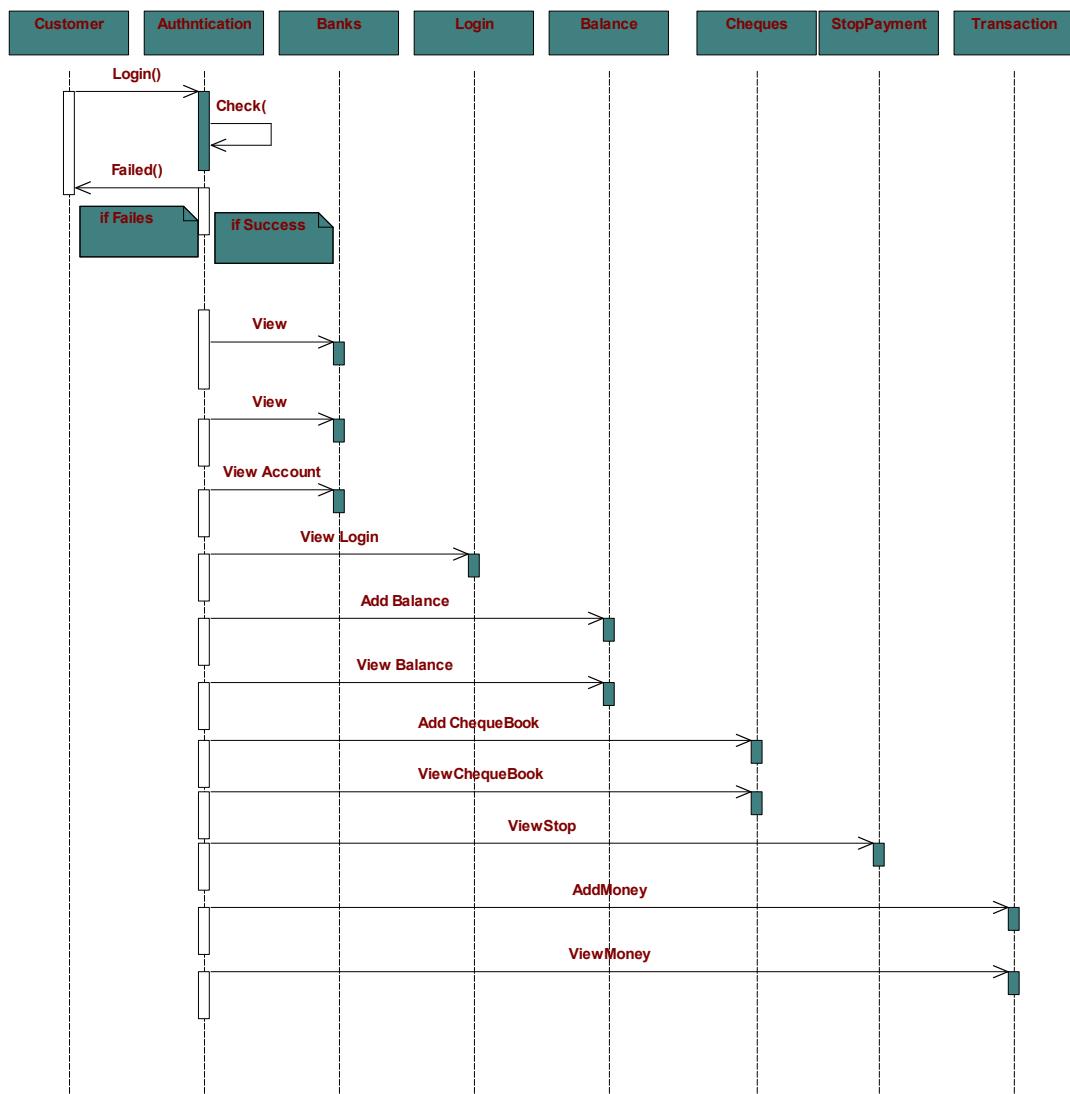


Fig 3.4.1(c) Sequence diagram for Customer

3.4.2 State chart diagram

A state chart diagram shows the behavior of classes in response to external stimulate. This diagram models the dynamic flow of control from state to state within a system.

Admin State chart Diagram

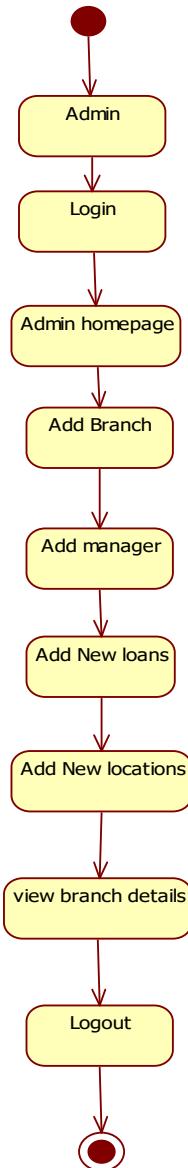


Fig 3.4.2(a) Statechart diagram for Admin

Manager State Chart Diagram:

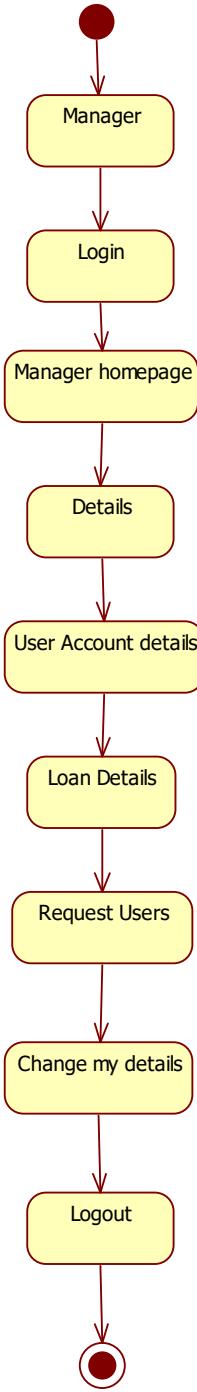


Fig 3.4.2(b) StateChart diagram for Manager

Customer State Chart Diagram:

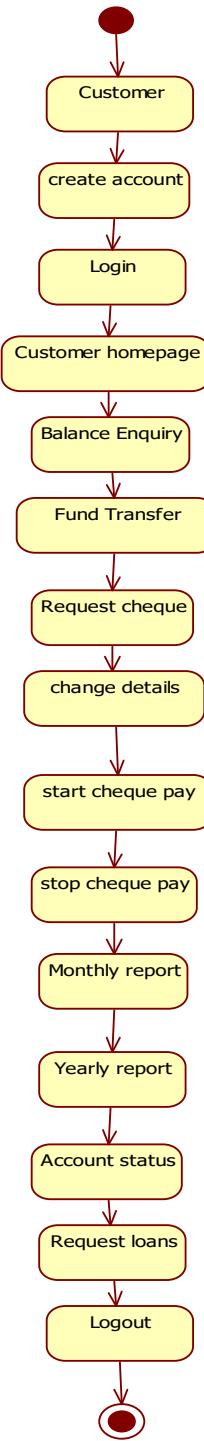


Fig 3.4.2(c) StateChart diagram for Customer

3.4.3 Activity diagram

An Activity diagram describes the behavior if the system in terms of activites

Activities are modeling elements that represents the execution of set of operations

Employee Activity Diagram:

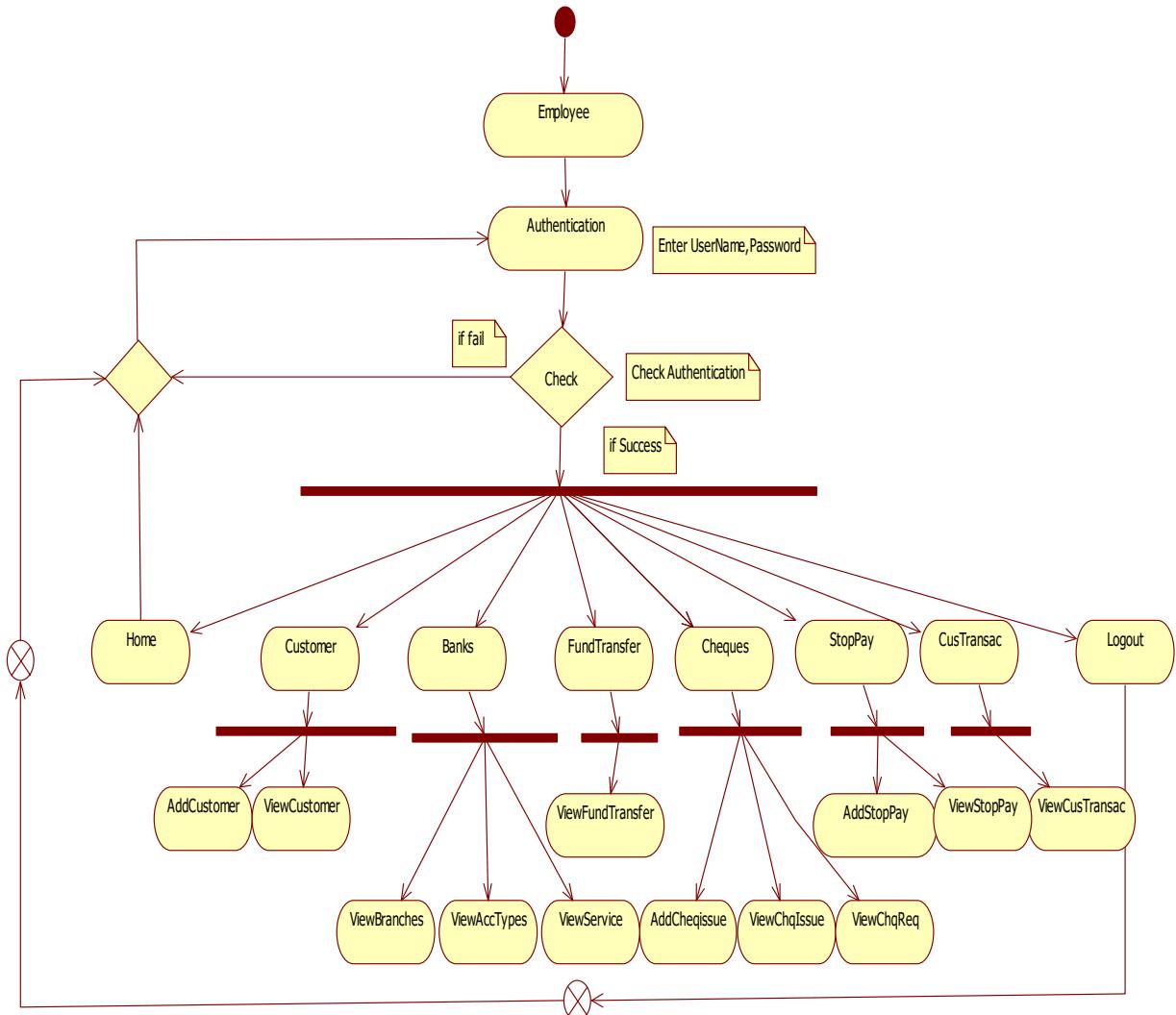


Fig 3.4.3(a) Activity diagram for Employee

Customer Activity Diagram:

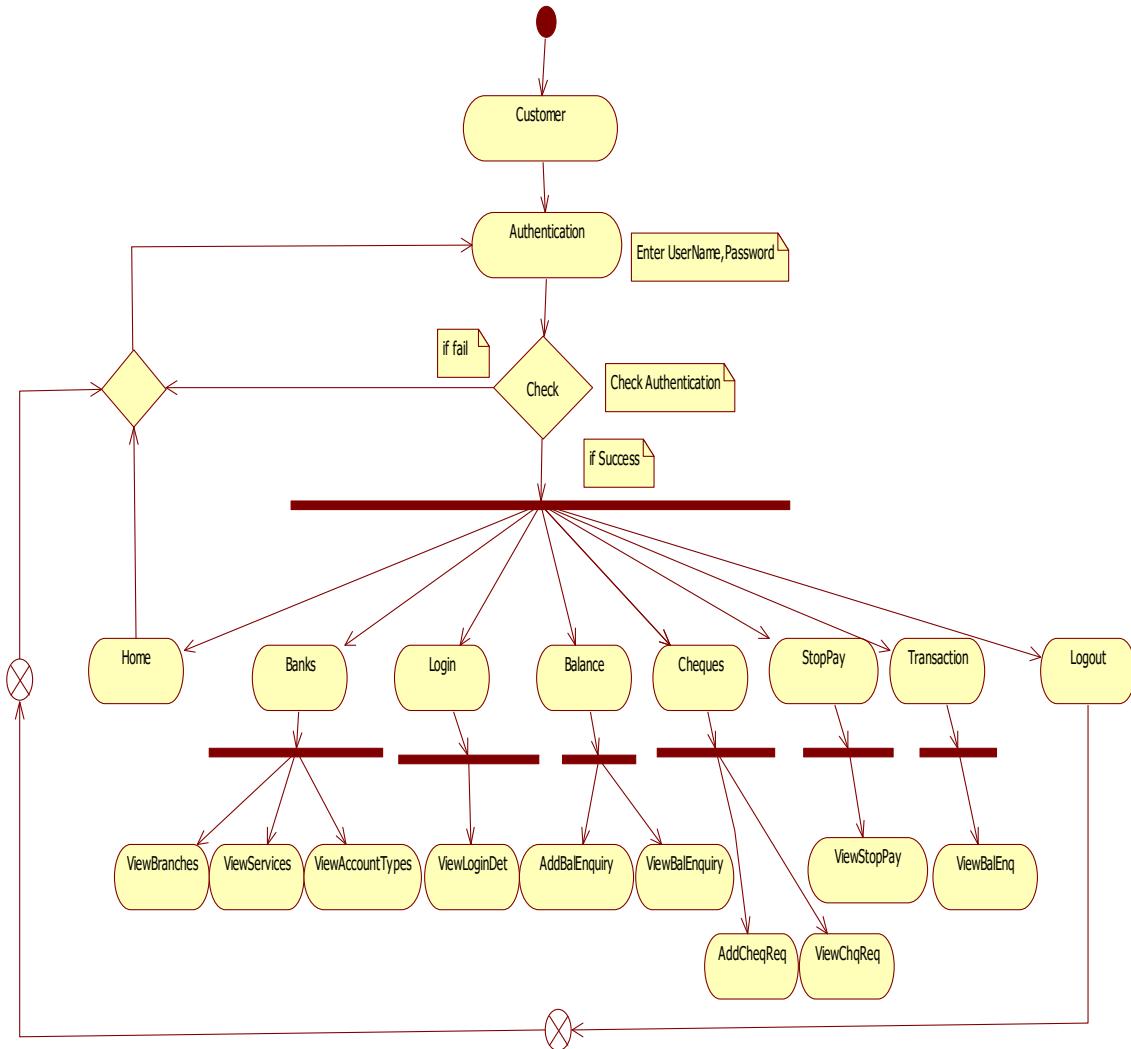


Fig 3.4.3(b) Activity diagram for customer

4. System Design Document

4.1 Subsystem Decomposition

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

4.2 Design Goals

- Reduced entry work
- Easy retrieval of information
- Reduced error due to human intervention
- User friendly screen to enter the data
- Portable and flexible for further enhancement
- Fast finding of information request

5. Object Design Document

5.1 Object Design Model

SDLC METHODOLOGIES

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models. The steps for Spiral Model can be generalized as follows. The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.

- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
- A second prototype is evolved by a fourfold procedure:
 1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
 2. Defining the requirements of the second prototype.
 3. Planning an designing the second prototype.
 4. Constructing and testing the second prototype.
- At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involved development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer’s judgment, result in a less-than-satisfactory final product.

- The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
- The final system is constructed, based on the refined prototype.

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

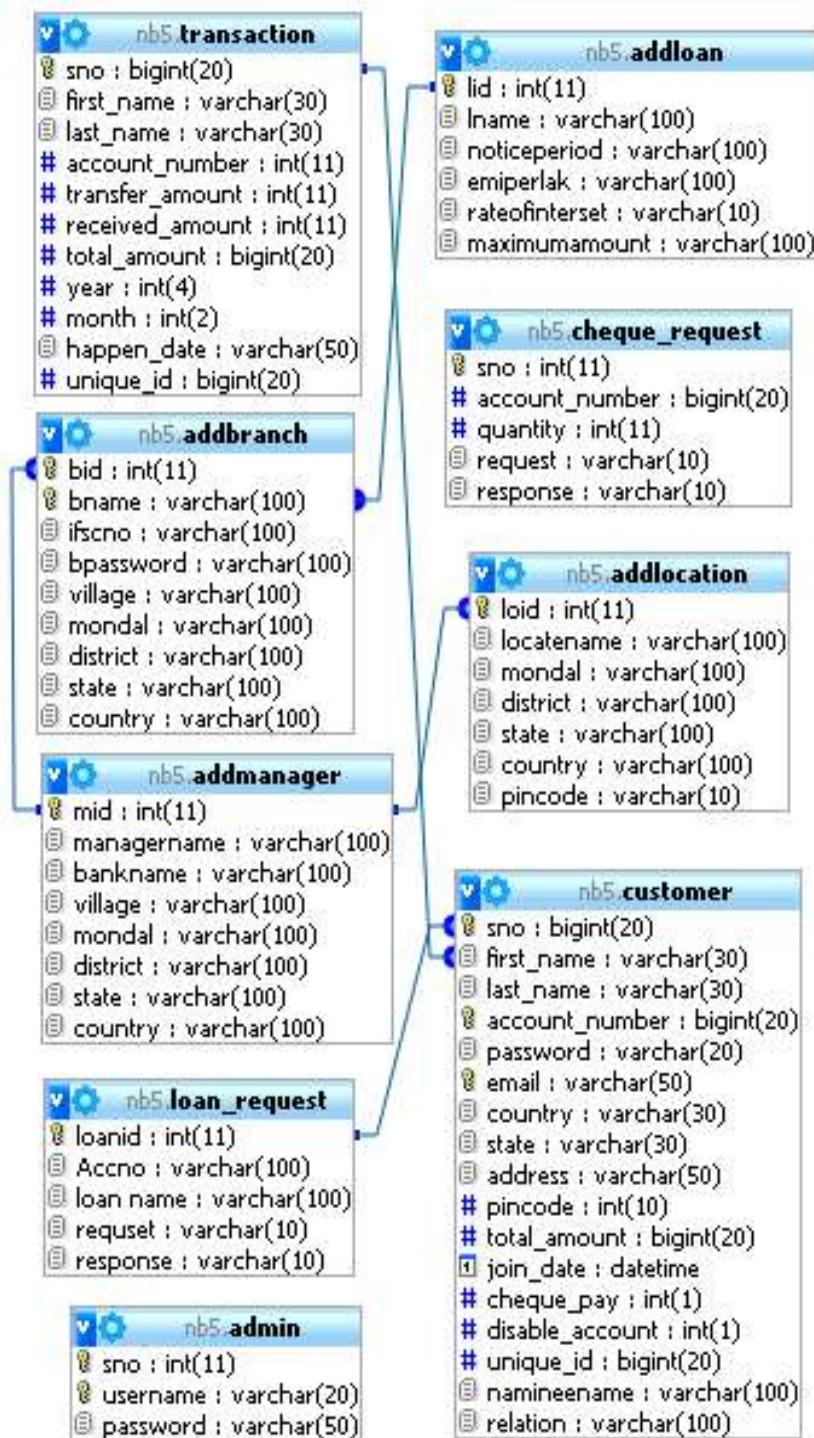
Stages in SDLC

- ◆ Requirement Gathering
- ◆ Analysis
- ◆ Designing
- ◆ Coding
- ◆ Testing

5.2 E – R DIAGRAMS

- The relation upon the system is structure through a conceptual ER-Diagram, which not only specifics the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.
- The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the date modeling activity the attributes of each data object noted is the ERD can be described resign a data object descriptions.
- The set of primary components that are identified by the ERD are
 - ◆ Data object
 - ◆ Relationships
 - ◆ Attributes
 - ◆ Various types of indicators.

The primary purpose of the ERD is to represent data objects and their relationships.



5.3. DATA FLOW DIAGRAMS

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this lead to the modular design. A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

6. Front end and Backend Tools

Software Requirements

Operating System	: Windows XP 7,8 ,Later
Database Server	: Mysql
Client	: Java Script
Tools	: Xampp
User Interface	: HTML,CSS ,Bootstrap with Ajax
Code Behind	: PHP
Web server	: Apache

HARDWARE REQUIREMENTS:

Processor	: Dual Core
RAM	: 1GB Ra
Hard Disk	: PC with 20GB

6.1 Details about Front End

Introduction to Php

PHP is an "HTML-embedded scripting language" primarily used for dynamic Web applications. The first part of this definition means that PHP code can be interspersed with HTML, making it simple to generate dynamic pieces of Web pages on the fly. As a scripting language, PHP code requires the presence of the PHP processor. PHP code is normally run in plain-text scripts that will only run on PHP-enabled computers (conversely programming languages can create standalone binary executable files, a.k.a. programs). PHP takes most of its syntax from C, Java, and Perl. It is an open source technology and runs on most operating systems and with most Web servers. PHP was written in the C programming language by RasmusLerdorf in 1994 for use in monitoring his online resume and related personal information. For this reason, PHP originally stood for "Personal Home Page". Lerdorf combined PHP with his own Form Interpreter, releasing the combination publicly as PHP/FI (generally referred to as PHP 2.0) on June 8, 1995. Two programmers, ZeevSuraski and AndiGutmans, rebuilt PHP's core, releasing the updated result as PHP/FI 2 in 1997. The acronym was formally changed to PHP: HyperText Preprocessor, at this time. (This is an example of a recursive acronym: where the acronym itself is in its own definition.) In 1998, PHP 3 was released, which was the first widely used version. PHP 4 was released in May 2000, with a new core, known as the Zend Engine 1.0. PHP 4 featured improved speed and reliability over PHP 3. In terms of features, PHP 4 added references, the Boolean type, COM support on Windows, output buffering, many new array functions, expanded object-oriented programming, inclusion of the PCRE library, and more. Maintenance releases of PHP 4 are still available, primarily for security updates. PHP 5 was released in July 2004, with the updated Zend Engine 2.0. Among the many new features in PHP 5 are:

improved object-oriented programming

embedded SQLite

support for new MySQL features (see the image at right)

exception handling using a try..catch structure

integrated SOAP support (see the image at right)

the Filter library (in PHP 5.1)

better XML tools

and much, much more. PHP 6 has been in development since October of 2006. The most significant change will be native support for Unicode. Unpopular, deprecated features such as Magic Quotes, register_globals, safe_mode, and the HTTP_*_VARS variables will disappear in PHP 6. Although PHP is still primarily used for server-side generation of Web pages, it can also be used to perform command-line scripting or to create graphical applications with the help of GTK+.

Originally started in 1994 as a replacement for various Perl scripts used to maintain his Personal Web Page (thus the acronym PHP) by the Danish/Greenlandish programmer RasmusLerdorf, the project has since grown into an open source community effort. Initial uses of PHP were limited to basic tasks such as counting how many visitors a web site had received, the introduction of PHP/FI (The FI stands for Form Interpreter) added additional functionality including implementation for the C programming language.

In addition to the inclusion of C support, PHP/FI also introduced native support for database communications. These two features have become the bedrock for future versions of PHP and together allowed the relatively swift and easy construction of dynamic web sites. While sites created with PHP at that time may be considered simple by modern standards were still leaps and bounds more impressive than static content and certainly helped to pave the way for the internet to flourish and grow as a medium.

In 1995 Lerdorf made the project public in an effort to improve the PHP code base in both reliability and scope. This release would eventually be known as PHP 2. At the time Perl was still the preeminent language for performing the tasks that PHP was seeking to perform and

6.2 Details about BackEnd

MySQL

What is a Database?

A database is a structure that comes in two flavors: a flat database and a relational database. A relational database is much more oriented to the human mind and is often

preferred over the gabble-de-gook flat database that are just stored on hard drives like a text file. MySQL is a relational database.

MySQL is a powerful database. It's very good and free of charge. Many developers in the world selected mysql and php for developing their website.

History of MySQL:

We started out with the intention of using the MySQL database system to connect to our tables using our own fast low-level (ISAM) routines. However, after some testing, we came to the conclusion that mSQL was not fast enough or flexible enough for our needs. This resulted in a new SQL interface to our database but with almost the same API interface as MySQL. This API was designed to allow third-party code that was written for use with mSQL to be ported easily for use with MySQL.

MySQL is named after co-founder Monty Widenius's daughter, My. The name of the MySQL Dolphin (our logo) is "Sakila," which was chosen from a huge list of names suggested by users in our "Name the Dolphin" contest. The winning name was submitted by Ambrose Twebaze, an Open Source software developer from Swaziland, Africa. According to Ambrose, the feminine name Sakila has its roots in SiSwati, the local language of Swaziland. Sakila is also the name of a town in Arusha, Tanzania, near Ambrose's country of origin, Uganda.

The history of MySQL dates back to 1995. For those of you that don't know what MySQL is, MySQL is a database program. Best MySQL Web Hosting has written this short article on the history of MySQL. We hope you learn from this article. As mentioned before, MySQL was started in 1995 under the name of MySQL AB. MySQL was founded by three guys, Michael Widenius, David Axmark, and Allan Larsson. The company was eventually sold in 2008 to Sun Microsystems at a cost of one billion dollars. MySQL has offices in Sweden and California.

Throughout MySQL's history, there have been numerous highlights that have made the computer industry better and what it is today. Those highlights are the following:

1998: In 1998, Windows decided to release their version for the Windows 95 and NT programs.

2001: MySQL released version 3.23.

2003: MySQL released version 4.0. Version 4.0 featured the union clause. The union clause allowed two data groups to be merged into one group.

2004: MySQL released version 4.1. Version 4.1 featured R-trees, B-trees, subqueries, and prepared statements. R-trees are data structures that are used for spatial access methods. B-trees are also data structures but their main purpose is to keep data sorted for easier access.

2005: MySQL released version 5.0. Version 5.0 contained cursors, triggers, views, stored procedures, and XA transactions. Cursors are important because they assist in the processing of rows within database queries. Triggers come in two different versions; row and statement. Basically, triggers are codes that are the end result of certain actions within the database.

2008: MySQL releases version 5.1. Version 5.1 includes partitioning, event scheduler, row based replications, server log tables, and plugin API.

MySQL is not stopping with version 5.1. Just like they have done since 1995, they are continuing to improve MySQL. MySQL 6.0 is already in the works and it is going to contain referential integrity, additional unicodes, and a new storage engine.

Advantages:

1. Scalability and Flexibility

The MySQL database server provides the ultimate in scalability, sporting the capacity to handle deeply embedded applications with a footprint of only 1MB to running massive data warehouses holding terabytes of information. Platform flexibility is a stalwart feature of MySQL with all flavors of Linux, UNIX, and Windows being supported. And, of course, the open source nature of MySQL allows complete customization for those wanting to add unique requirements to the database server.

2. High Performance

A unique storage-engine architecture allows database professionals to configure the MySQL database server specifically for particular applications, with the end result being amazing performance results. Whether the intended application is a high-speed transactional processing system or a high-volume web site that services a billion queries a day, MySQL can meet the most demanding performance expectations of any system. With high-speed load utilities, distinctive memory caches, full text indexes, and other performance-enhancing mechanisms, MySQL offers all the right ammunition for today's critical business systems.

3. High Availability

Rock-solid reliability and constant availability are hallmarks of MySQL, with customers relying on MySQL to guarantee around-the-clock uptime. MySQL offers a variety of high-availability options from high-speed master/slave replication configurations, to specialized Cluster servers offering instant failover, to third party vendors offering unique high-availability solutions for the MySQL database server.

4. Robust Transactional Support

MySQL offers one of the most powerful transactional database engines on the market. Features include complete ACID (atomic, consistent, isolated, durable) transaction support, unlimited row-level locking, distributed transaction capability, and multi-version transaction support where readers never block writers and vice-versa. Full data integrity is also assured through server-enforced referential integrity, specialized transaction isolation levels, and instant deadlock detection.

5. Web and Data Warehouse Strengths

MySQL is the de-facto standard for high-traffic web sites because of its high-performance query engine, tremendously fast data insert capability, and strong support for specialized web functions like fast full text searches. These same strengths also apply to data warehousing environments where MySQL scales up into the terabyte range for either single servers or scale-out architectures. Other features like main memory tables, B-tree and hash indexes, and compressed archive tables that reduce storage requirements by up to eighty-percent make MySQL a strong standout for both web and business intelligence applications.

6. Strong Data Protection

Because guarding the data assets of corporations is the number one job of database professionals, MySQL offers exceptional security features that ensure absolute data protection. In terms of database authentication, MySQL provides powerful mechanisms for ensuring only authorized users have entry to the database server, with the ability to block users down to the client machine level being possible. SSH and SSL support are also provided to ensure safe and secure connections. A granular object privilege framework is present so that users only see the data they should, and powerful data encryption and decryption functions ensure that sensitive data is protected from unauthorized viewing. Finally, backup and recovery utilities provided through MySQL and third party software

vendors allow for complete logical and physical backup as well as full and point-in-time recovery.

7. Comprehensive Application Development

One of the reasons MySQL is the world's most popular open source database is that it provides comprehensive support for every application development need. Within the database, support can be found for stored procedures, triggers, functions, views, cursors, ANSI-standard SQL, and more. For embedded applications, plug-in libraries are available to embed MySQL database support into nearly any application. MySQL also provides connectors and drivers (ODBC, JDBC, etc.) that allow all forms of applications to make use of MySQL as a preferred data management server. It doesn't matter if it's PHP, Perl, Java, Visual Basic, or .NET

8. Management Easy

MySQL offers exceptional quick-start capability with the average time from software download to installation completion being less than fifteen minutes. This rule holds true whether the platform is Microsoft Windows, Linux, Macintosh, or UNIX. Once installed, self-management features like automatic space expansion, auto-restart, and dynamic configuration changes take much of the burden off already overworked database administrators. MySQL also provides a complete suite of graphical management and migration tools that allow a DBA to manage, troubleshoot, and control the operation of many MySQL servers from a single workstation. Many third party software vendor tools are also available for MySQL that handle tasks ranging from data design and ETL, to complete database administration, job management, and performance monitoring.

9. Open Source Freedom and 24 x 7 Support

Many corporations are hesitant to fully commit to open source software because they believe they can't get the type of support or professional service safety nets they currently rely on with proprietary software to ensure the overall success of their key applications. The questions of indemnification come up often as well. These worries can be put to rest with MySQL as complete around-the-clock support as well as indemnification is available through MySQL Network. MySQL is not a typical open source project as all the software is owned and supported by MySQLAB, and because of this, a unique cost and support model are

available that provides a unique combination of open source freedom and trusted software with support.

10. Lowest Total Cost of Ownership

By migrating current database-drive applications to MySQL, or using MySQL for new development projects, corporations are realizing cost savings that many times stretch into seven figures. Accomplished through the use of the MySQL database server and scale-out architectures that utilize low-cost commodity hardware, corporations are finding that they can achieve amazing levels of scalability and performance, all at a cost that is far less than those offered by proprietary and scale-up software vendors. In addition, the reliability and easy maintainability of MySQL means that database administrators don't waste time troubleshooting performance or downtime issues, but instead can concentrate on making a positive impact on higher level tasks that involve the business side of data

What is JavaScript?

- JavaScript was designed to add interactivity to HTML pages
- JavaScript is a scripting language
- A scripting language is a lightweight programming language
- JavaScript is usually embedded directly into HTML pages
- JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
- Everyone can use JavaScript without purchasing a license

Are Java and JavaScript the same?

NO!

Java and JavaScript are two completely different languages in both concept and design!Java (developed by Sun Microsystems) is a powerful and much more complex programming language - in the same category as C and C++.

What can a JavaScript do?

JavaScript gives HTML designers a programming tool – HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax! Almost anyone can put small "snippets" of code into their HTML pagesJavaScript can put dynamic

text into an HTML JavaScript statement like this: `document.write("<h1>" + name + "</h1>")` can write a variable text into an HTML page

JavaScript can react to events - A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element
JavaScript can read and write HTML elements - A JavaScript can read and change the content of an HTML element
JavaScript can be used to validate data - A JavaScript can be used to validate form data before it is submitted to a server.

The Real Name is ECMAScript

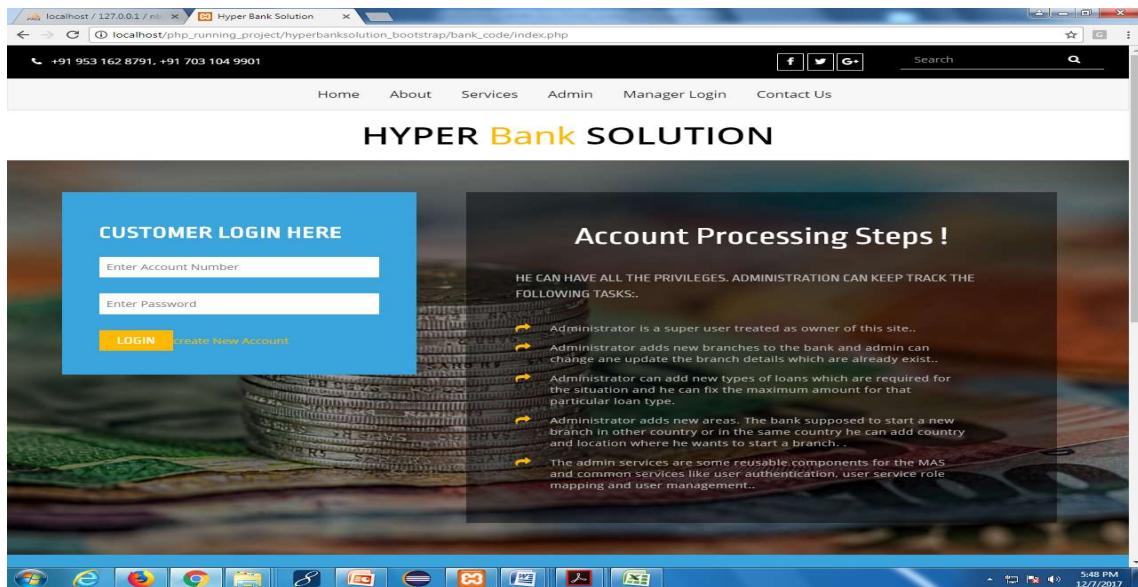
JavaScript's official name is ECMAScript. ECMAScript is developed and maintained by the ECMA organization. ECMA-262 is the official JavaScript standard.

The language was invented by Brendan Eich at Netscape (with Navigator 2.0), and has appeared in all Netscape and Microsoft browsers since 1996.

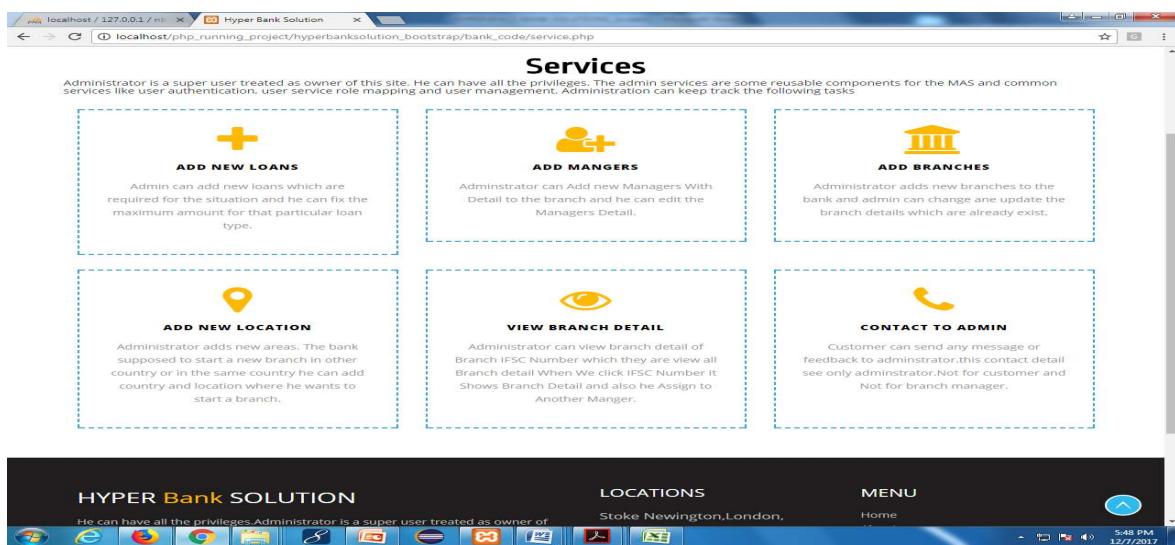
The development of ECMA-262 started in 1996, and the first edition of was adopted by the ECMA General Assembly in June 1997. The standard was approved as an international ISO (ISO/IEC 16262) standard in 1998. The development of the standard is still in progress.

7. Screens

Home



Services



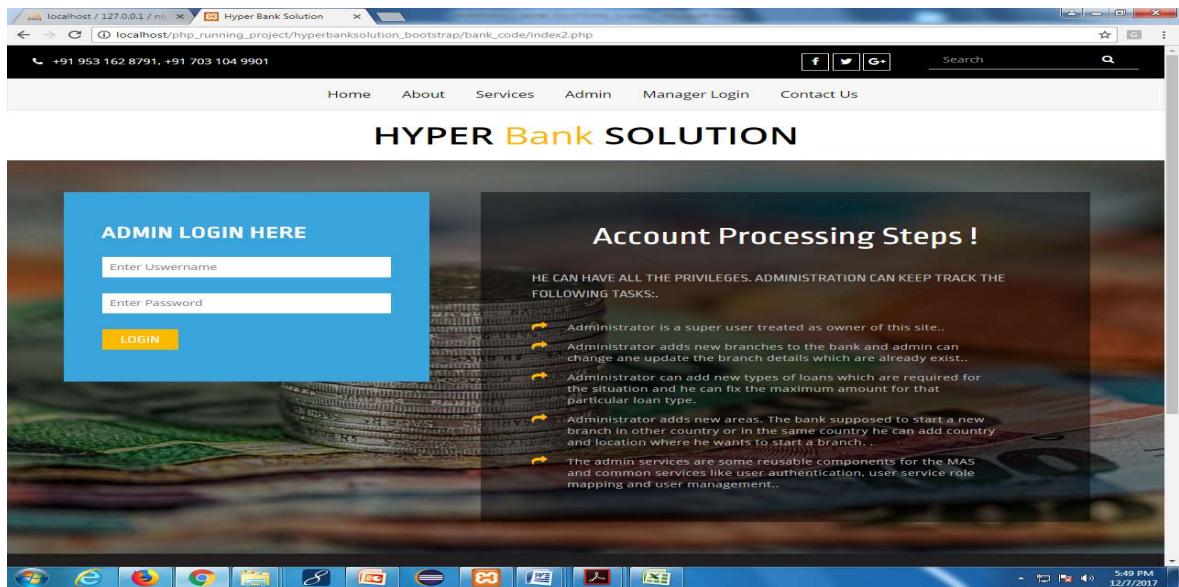
About US:

The screenshot shows a web browser window for 'Hyper Bank Solution' on a local host. The page title is 'About'. It features a large image of many Indian Rupee banknotes stacked together. To the right of the image is a block of text describing the working employee's role and manager's responsibilities. Below this is a navigation menu with links to Home, About, Services, Admin, Manager Login, and Contact Us. The footer contains standard links for locations and menu items.

Contact US:

The screenshot shows a web browser window for 'Hyper Bank Solution' on a local host. The page title is 'Contact Us'. On the left, there are four input fields for 'Enter Subject', 'Enter Your Name', 'Enter Your Email', and 'Enter Message', followed by a 'SUBMIT' button. To the right is a map of Paris, France, showing various neighborhoods and major roads. The footer contains standard links for locations and menu items.

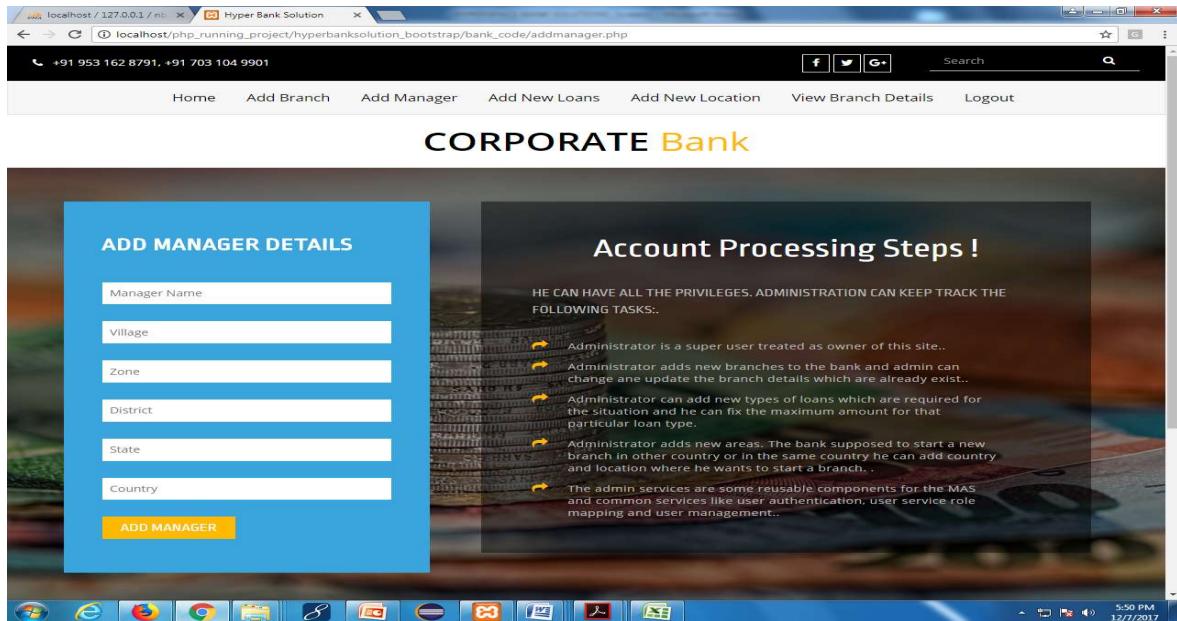
Admin Login:



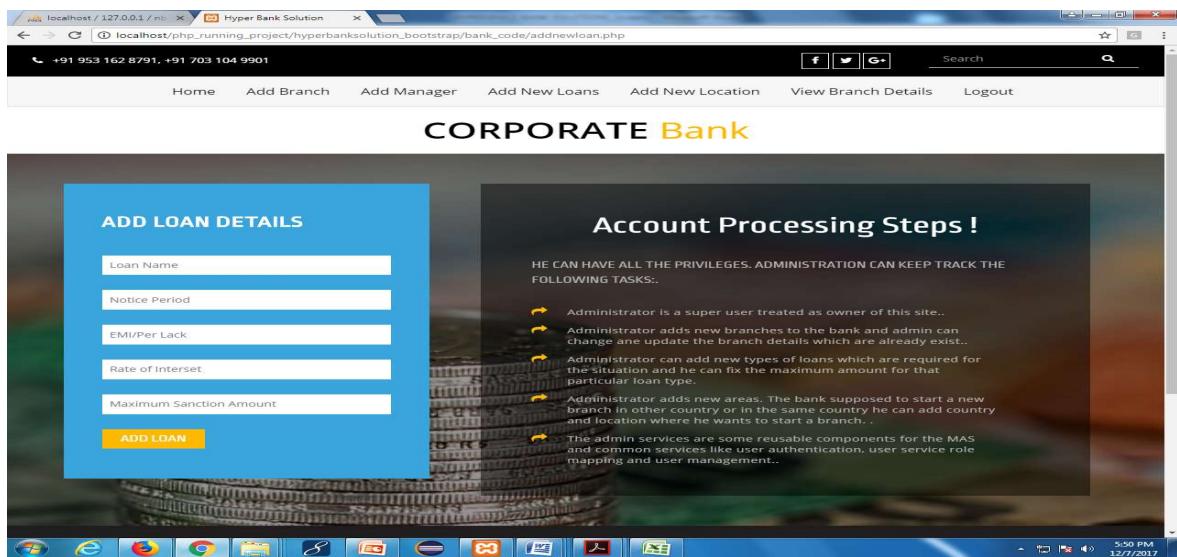
Add Branch:



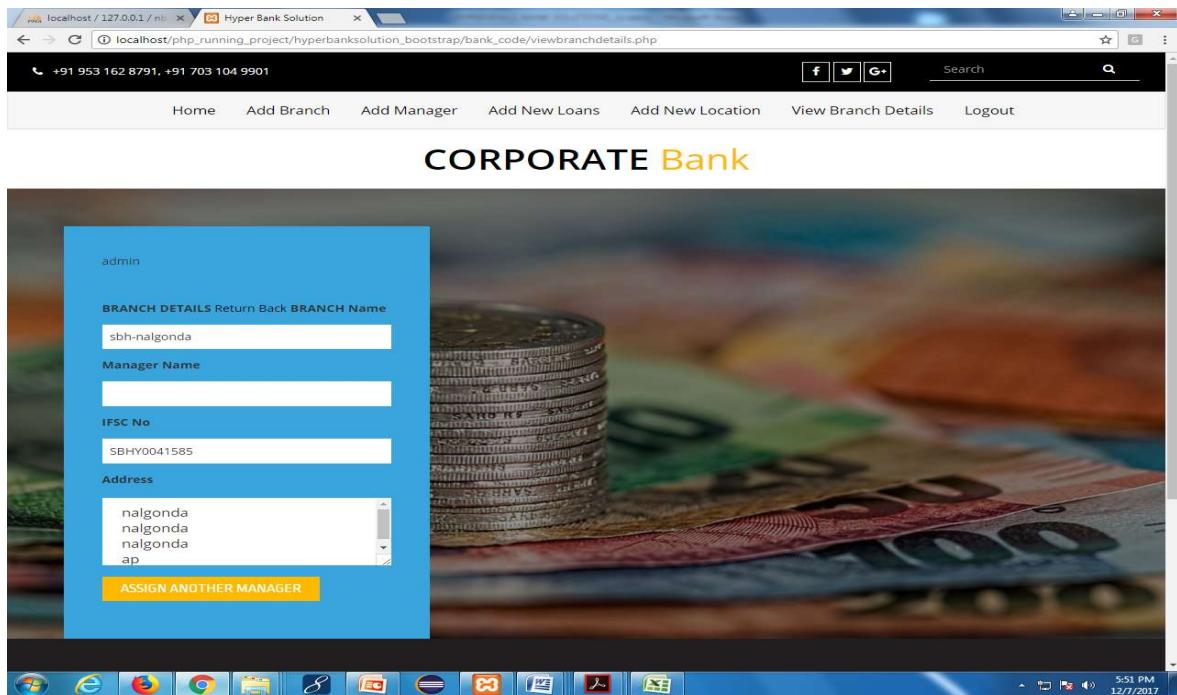
Add Manager:



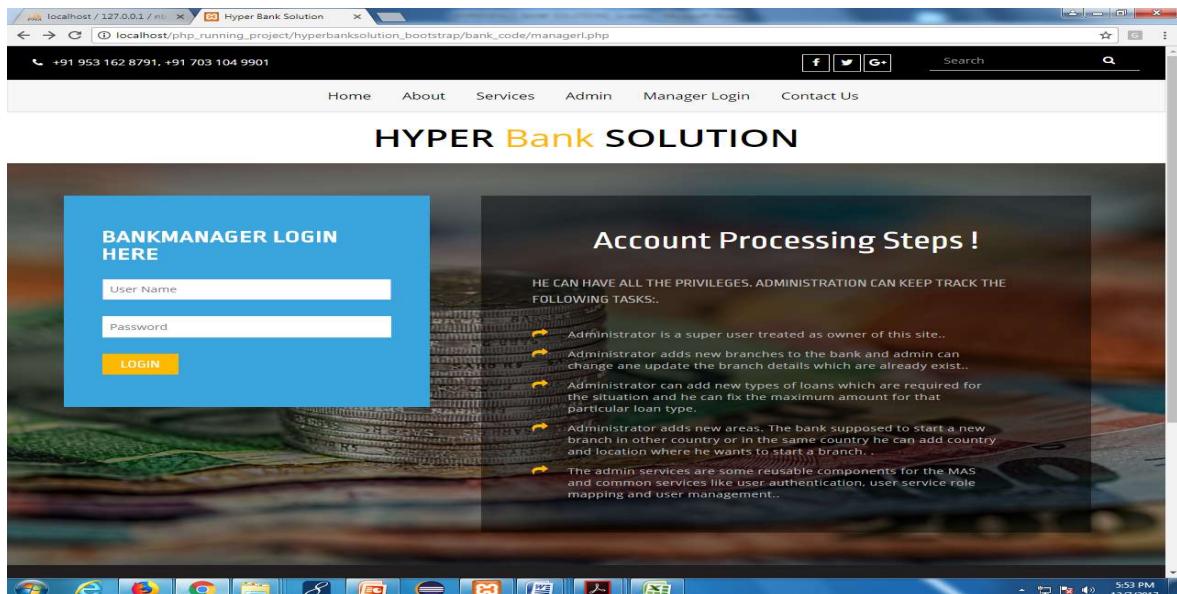
Add New Loans:



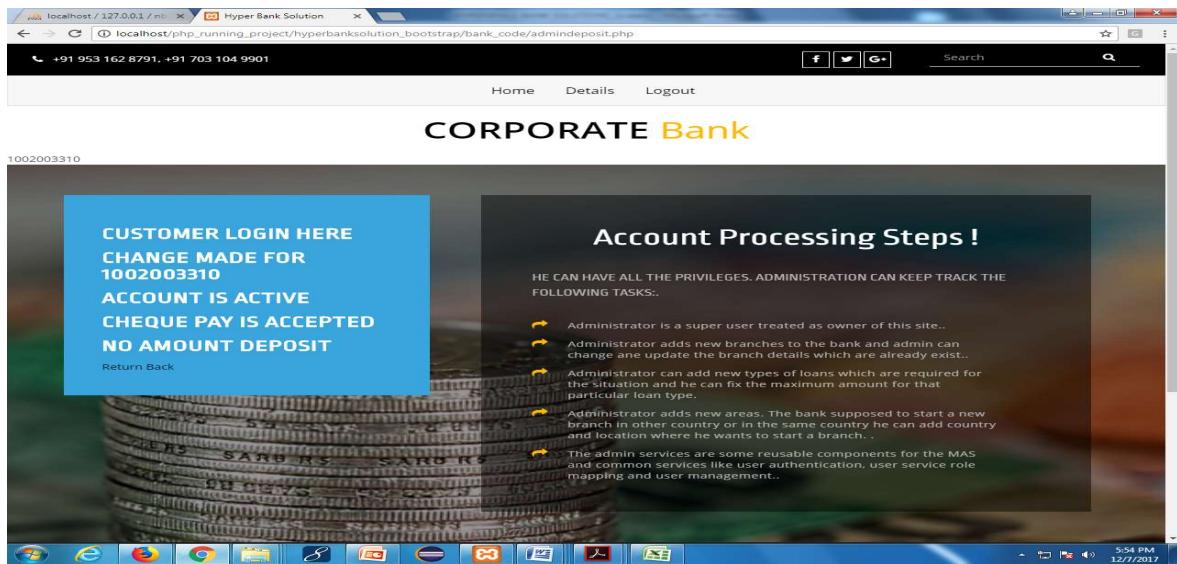
View Branch Details:



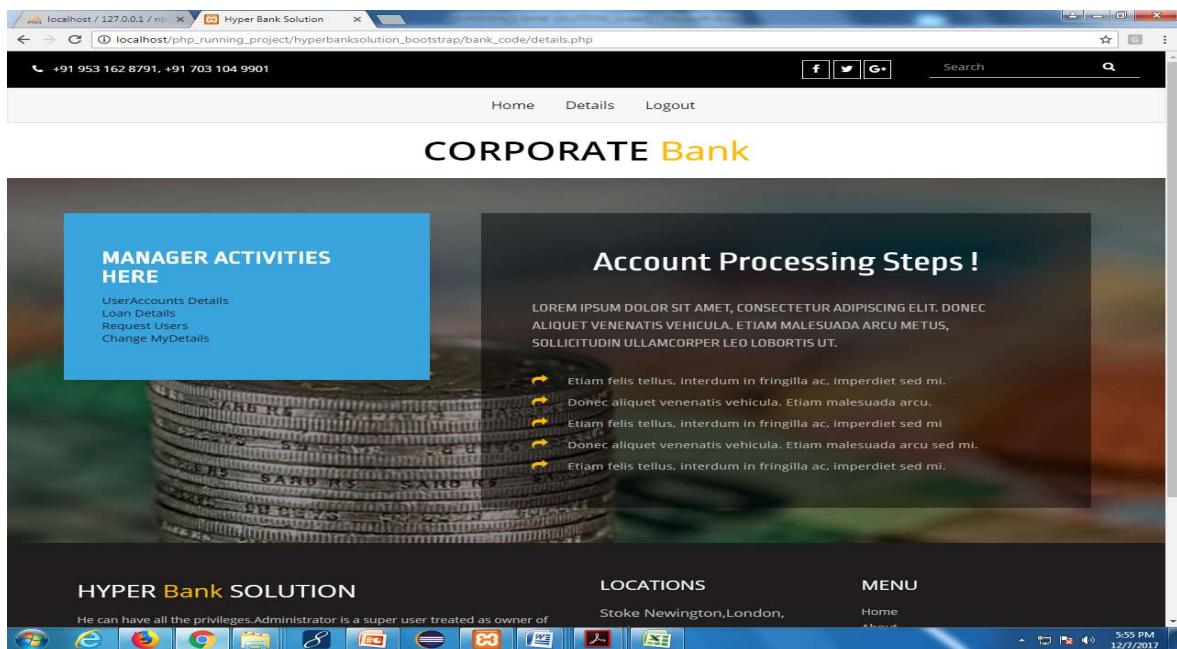
Manager Login:



Transfer Amount to Customer:



Details:



User Account Details:

The screenshot shows a web browser window titled "Hyper Bank Solution" with the URL "localhost:127.0.0.1/n... Hyper Bank Solution". The main content area is titled "CORPORATE Bank". On the left, there is a blue sidebar labeled "ACCOUNT DETAILS" containing the following fields:

Full Name	reoreo
Account Number	1002003301
Email	reo@gmail.com
Total Amount	8500
Join Date	2017-06-21 01:04:13
Cheque Pay	1
Disabled Accno	1
Address	r r r

The background of the page features a blurred image of coins and banknotes.

Loan Details:

The screenshot shows a web browser window titled "Hyper Bank Solution" with the URL "localhost:127.0.0.1/n... Hyper Bank Solution". The main content area is titled "CORPORATE Bank". On the left, there is a blue sidebar labeled "Loan DETAILS" containing the following fields:

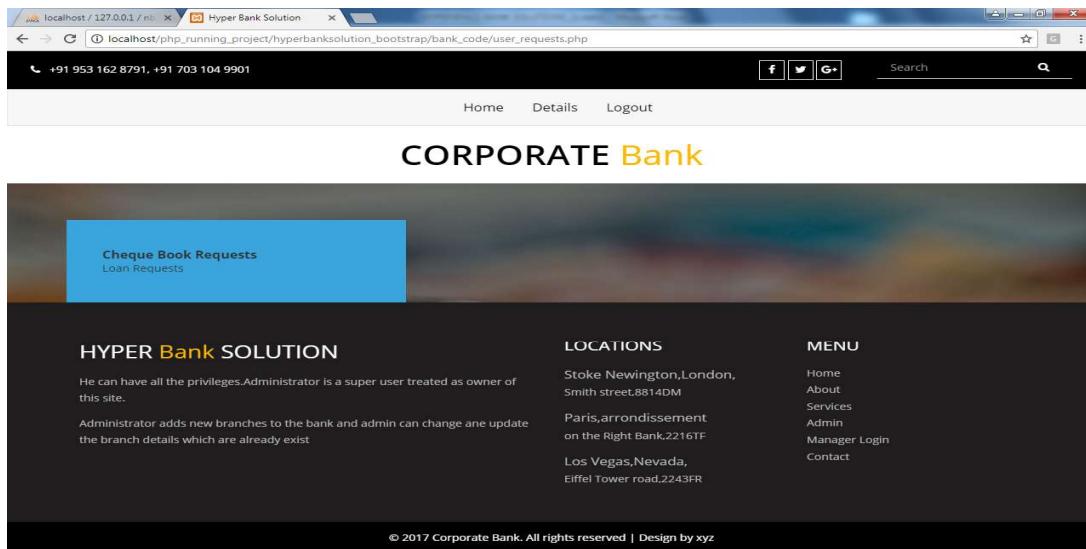
Loan Name	Education Loan
Notice Period	2 Years
EMI/per Lack	2000
Rate of Interest	5%
Maximum Amount	500000

The background of the page features a blurred image of coins and banknotes.

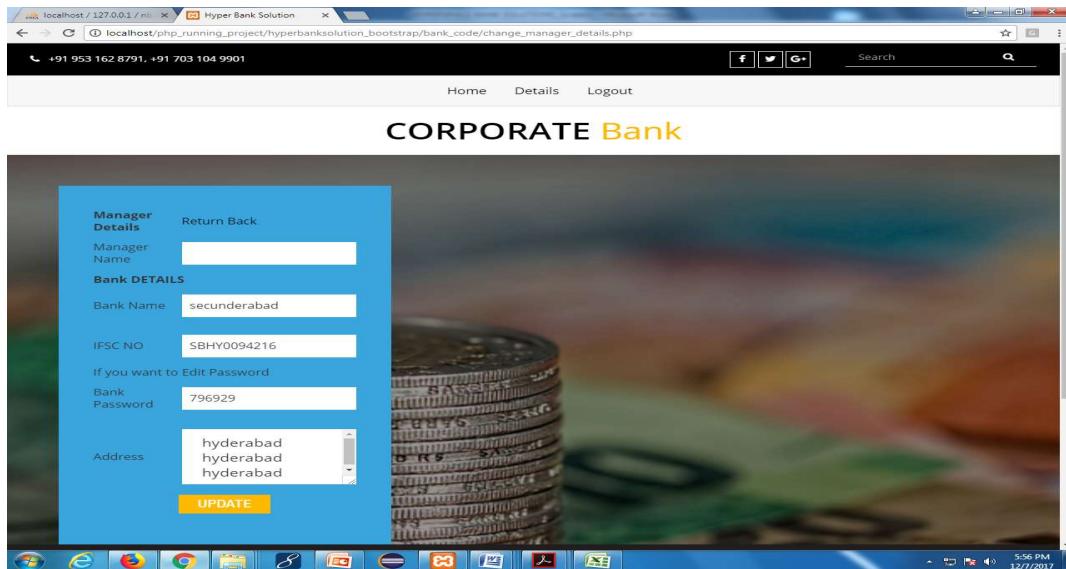
At the bottom of the page, there is a footer with the following information:

- HYPER Bank SOLUTION**
- A note: "He can have all the privileges. Administrator is a super user treated as owner of"
- LOCATIONS**: Stoke Newington, London, Smith Street 8P4ADM
- MENU**: Home, About

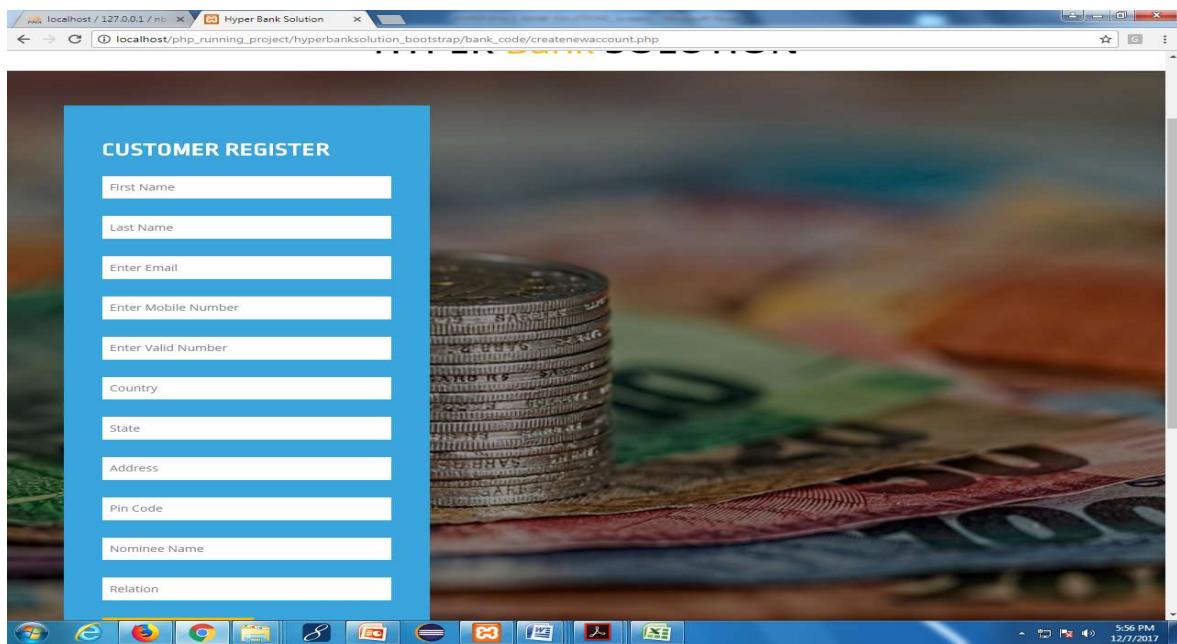
Request users:



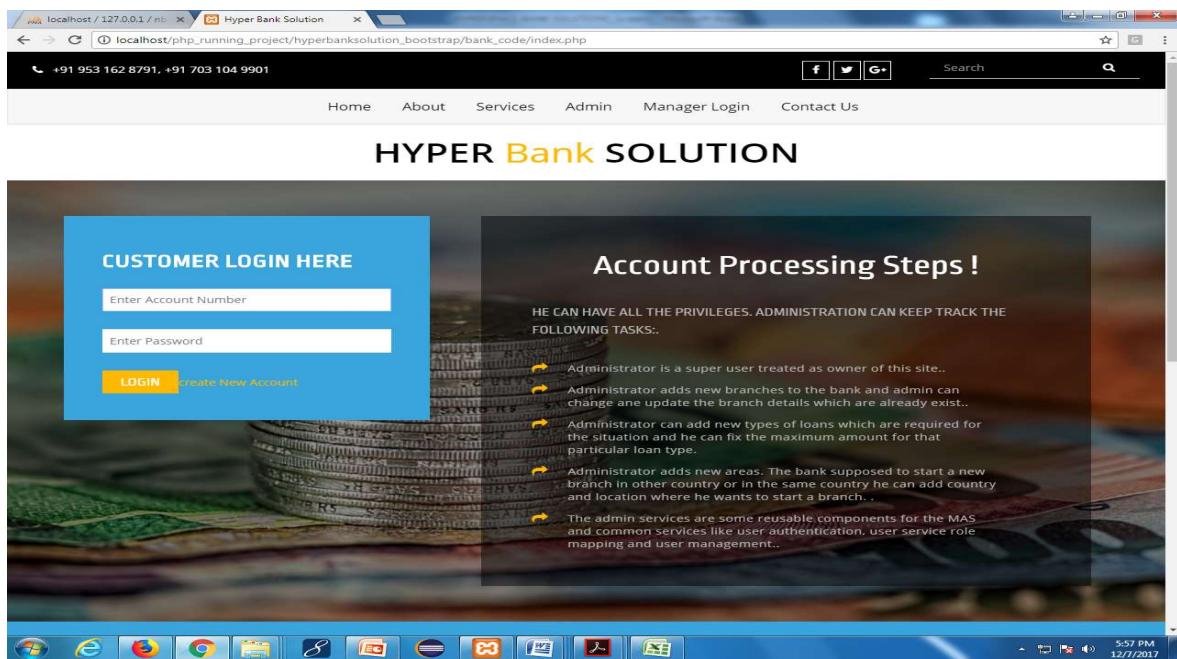
Change my details:



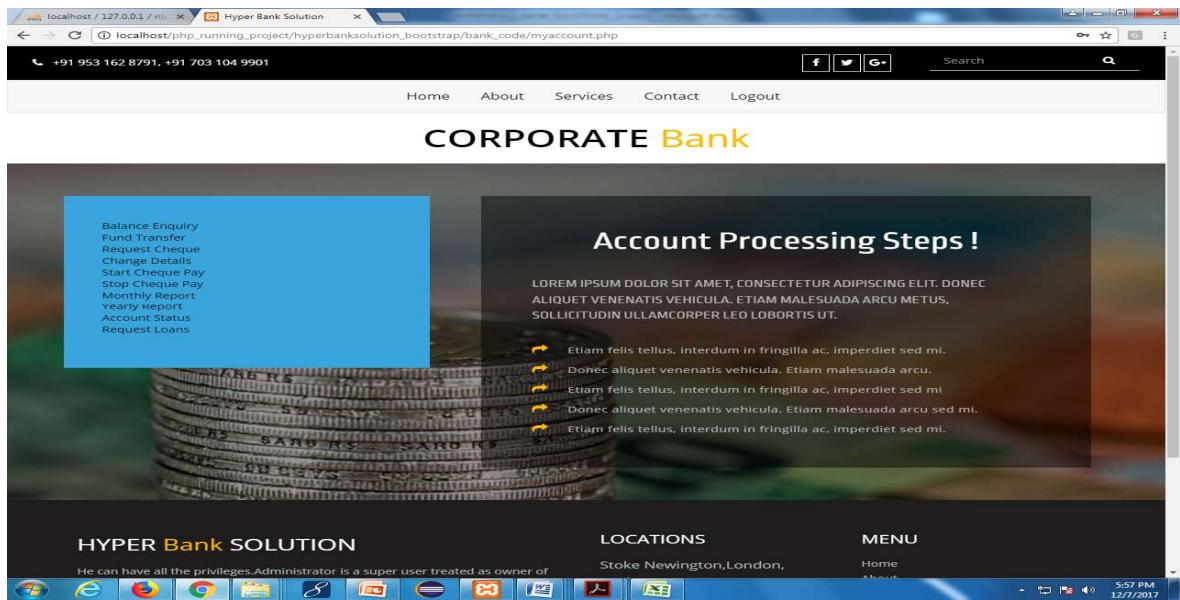
Customer Registration:



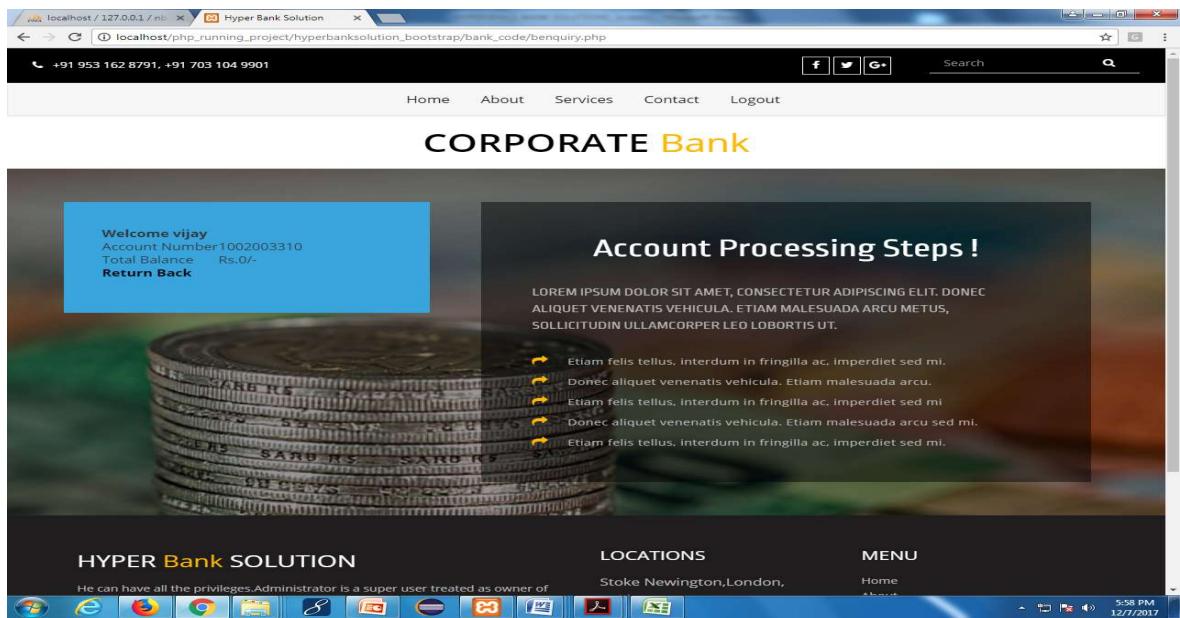
Customer Login:



Customer Home:



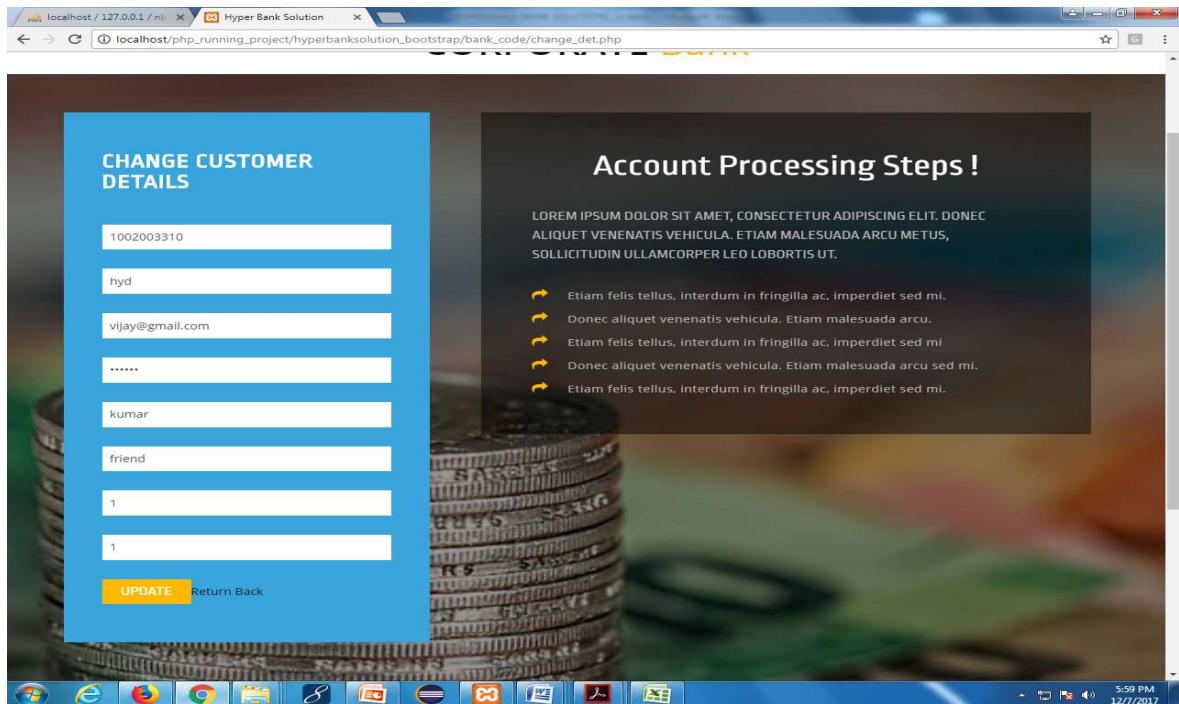
Balance enquiry:



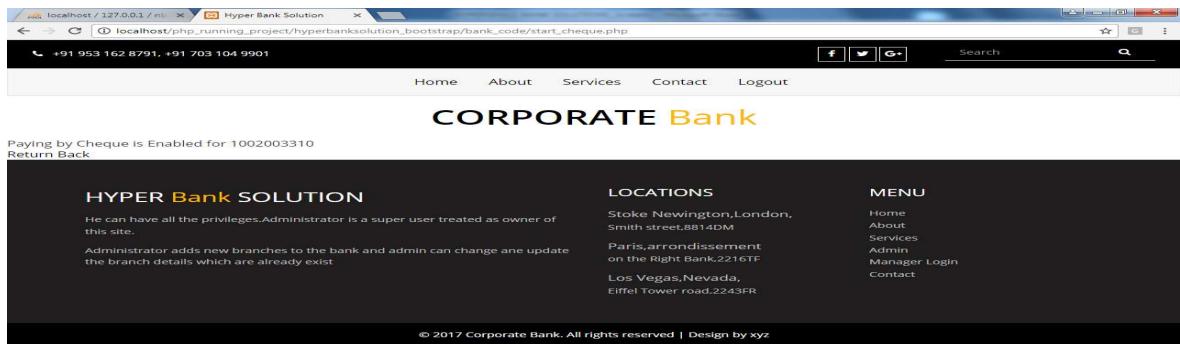
Transfer funds:



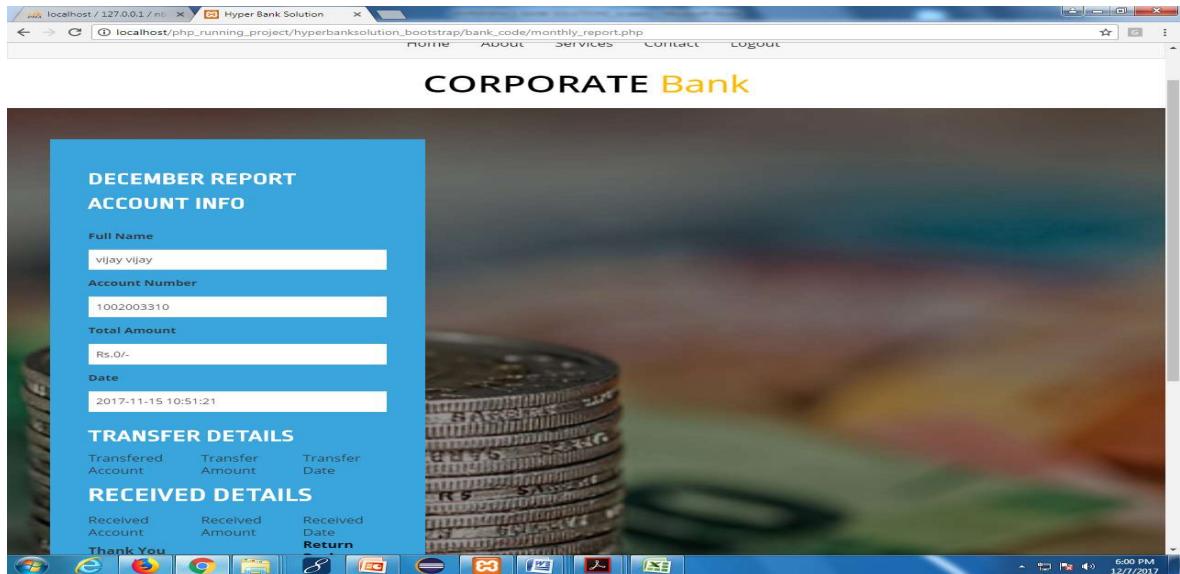
Updated profile:



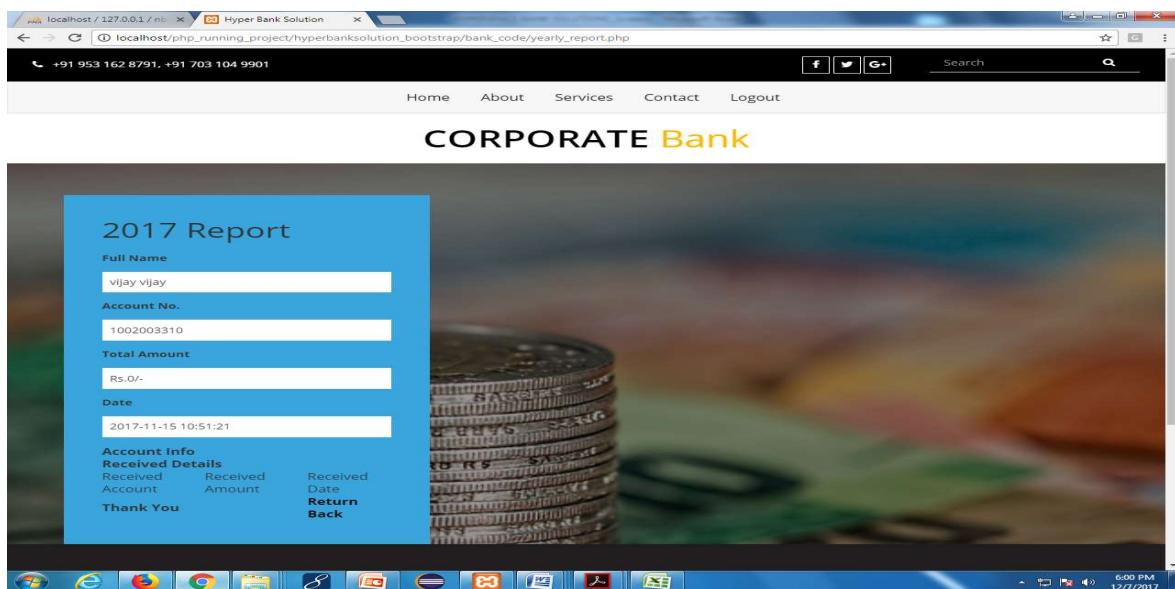
Start Cheque Pay:



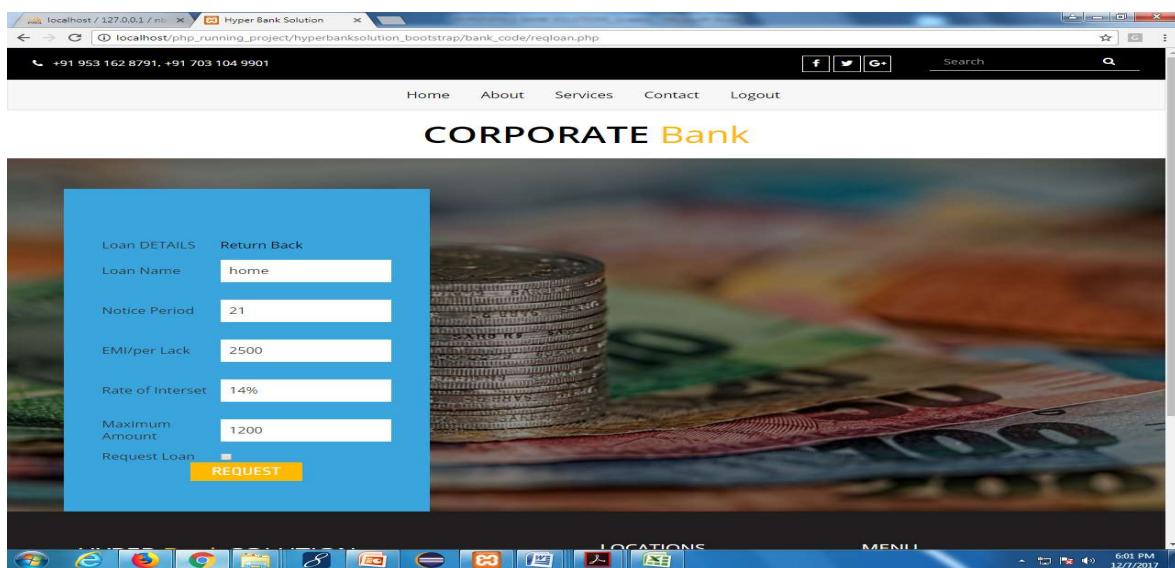
Monthly Report:



Yearly Report:

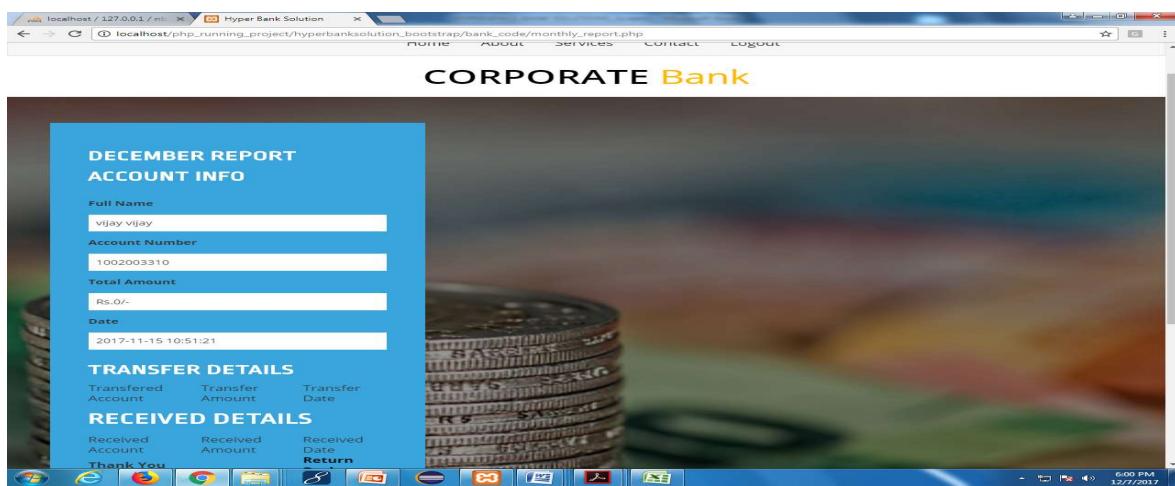


Request Loans:

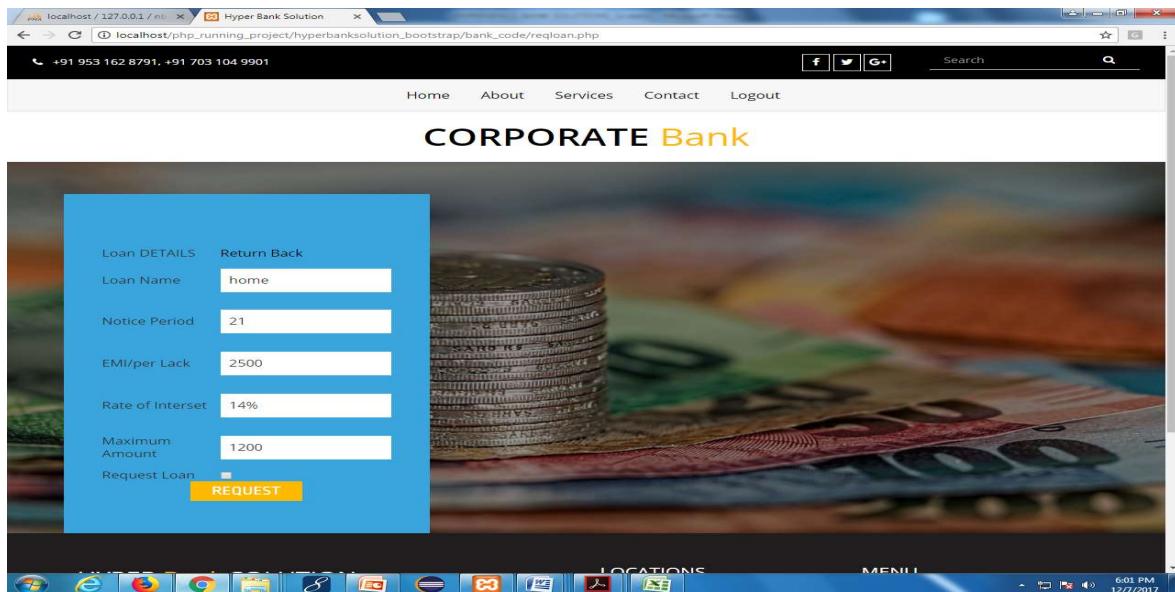


8. Reports

Customer Details Report:



Loan Request Reports:



9. Source Codes

Introduction

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on Implementation, designing of methods to achieve changeover and evaluation of changeover methods

Sample code

Add Loan code

```
<?php  
$t=date('h:i:s');  
$time = date('Y-m-d',time());  
$time=$time." ".$t;  
//echo "$time";  
// $time = date('Y-m-d',time());  
$check_pay = 1;  
$disable_account = 1;  
if(isset($_POST['btnSub']))  
{  
  
extract ($_REQUEST);  
$con=mysqli_connect("localhost","root","","nb5");  
$sql = "SELECT * FROM customer WHERE email='".$email"';  
$data = mysqli_query($con,$sql);
```

```

if(mysqli_num_rows($data) == 1)

{
    echo "<script>alert('User with this email id existed')</script>";
    //echo " User with this email id existed ";

    echo "<script>location='createnewaccount.php';</script>";

}

else

{

    $sql2 = "SELECT * FROM customer order by sno asc ";

    $result = mysqli_query($con,$sql2);

    $account_number = 1002003300;

    $unique_id = 4488;

    while($rec = mysqli_fetch_row($result))

    {

        $account_number = $rec[3];

        $unique_id = $rec[16];

    }

    $account_number = $account_number + 1;

    $unique_id = $unique_id + 1;

    $password = rand(111111,999999);
}

```

```

        $total_amount = '0';

        $con=mysqli_connect("localhost","root","","","nb5");

        $sql=mysqli_query($con,"inserinto
customer(first_name,last_name,account_number,password,email,mobile,adharno,country,stat
e,address,pincode,total_amount,join_date,cheque_pay,disable_account,unique_id,namineena
me,relation)

values('$fname','$lname','$account_number','$password','$email','$mobile','$adharno','
$country','$state','$address','$pincode','$total_amount','$time','$check_pay','$disable_account',
'$unique_id','$nm','$nr')");

        if($sql)

        {

            echo "<script>alert('succesful inserted')</script>";

        }

        else

        {

            echo"<script>alert('data not intersted')</script>";

        }

        mysqli_close($con);

    }

?>

```

```

<html>
<head>
<?php include('header.php');?>

```

```

</head>

<!-- //Head -->

<!-- Body -->

<body>

    <?php include('topmenu.php');?>

    <!-- Top-Bar -->

    <?php include('topbar.php');?>

    <!-- //Top-Bar -->

    <div class="banner-main jarallax">

        <div class="container">

            <div class="banner-inner">

                <div class="col-md-5 banner-left">

                    <form method="post">

                        <h3>Your Account is Created</h3>

                        <label>Your Account Number is</label>

                        <input type="text" value="<?php echo $account_number ; ?>"/>

                        <label>Your Account Password is</label>

                        <input type="text" value="<?php echo $password ; ?>"/>

                        <a href="index.php">Return Home</a>

```

10. Testing

10.1 Unit Testing

Introduction

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of software as a system element and attendant costs associated with a software failure are motivating factors for we planned, through testing. Testing is the process of executing a program with the intent of finding an error. The design of tests for software and other engineered products can be as challenging as the initial design of the product itself.

There of basically **two types of testing** approaches.

One is **Black-Box testing** – the specified function that a product has been designed to perform, tests can be conducted that demonstrate each function is fully operatedThe other is **White-Box testing** – knowing the internal workings of the product ,tests can be conducted to ensure that the internal operation of the product performs according to specifications and all internal components have been adequately exercisedWhite box and Black box testing methods have been used to test this package. The entire loop constructshave been tested for their boundary and intermediate conditions. The test data was designed with a view to check for all the conditions and logical decisions. Error handling has been taken care of by the use of exception handlers.

Unit Testing:

This testing method considers a module as single unit and checks the unit at interfaces and communicates with other modules rather than getting into details at statement level. Here the module will be treated as a black box, which will take some input and generate output. Outputs for a given set of input combination are pre-calculated and are generated by the module.

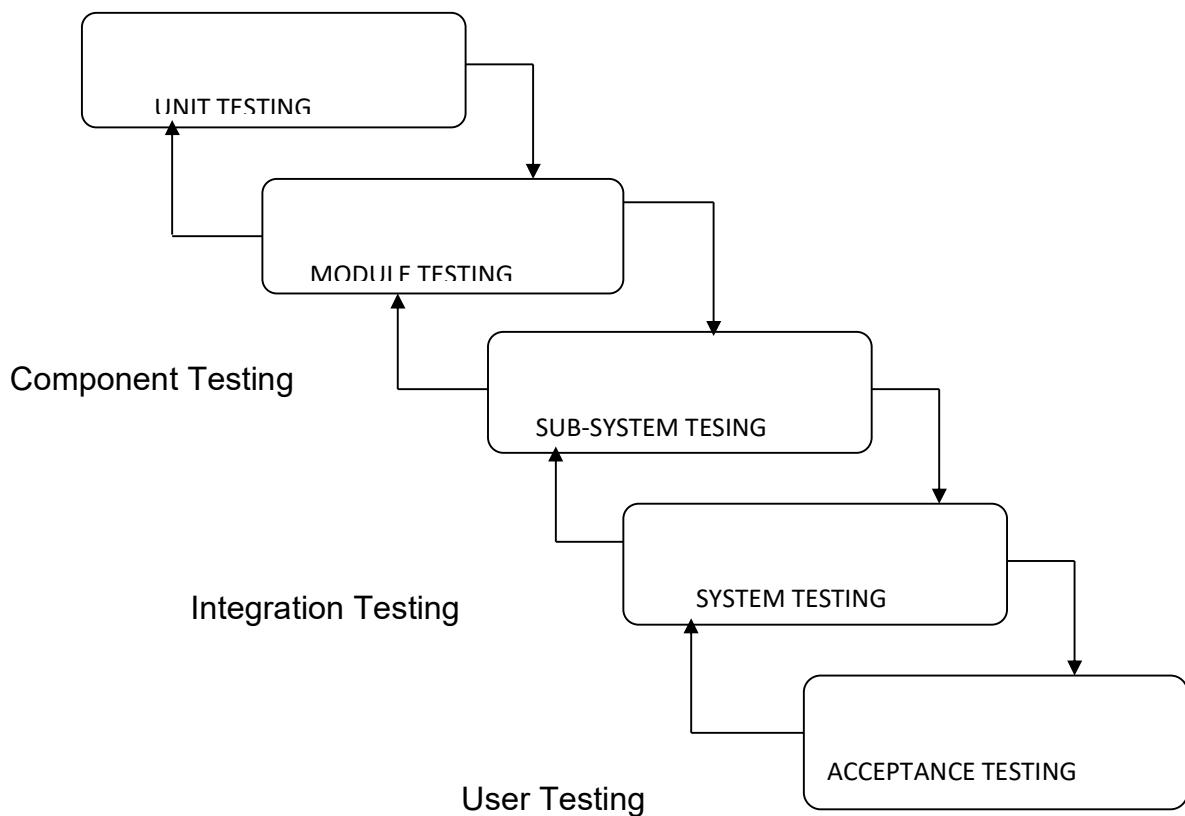
System Testing:

Here all the pre tested individual modules will be assembled to create the larger system and tests are carried out at system level to make sure that all modules are working in synchronous with each other. This testing methodology helps in making sure that all modules which are running perfectly when checked individually are also running in cohesion

with other modules. For this testing we create test cases to check all modules once and then generated test combinations of test paths through out the system to make sure that no path is making its way into chaos.

Integrated Testing

Testing is a major quality control measure employed during software development. Its basic function is to detect errors. Sub functions when combined may not produce than it is desired. Global data structures can represent the problems. Integrated testing is a systematic technique for constructing the program structure while conducting the tests. In a non - incremental integration all the modules are combined in advance and the program is tested as a whole. In incremental testing the program is constructed and tested in small segments where the errors are isolated and corrected. Different incremental integration strategies are top – down integration, bottom – up integration, regression testing.



Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

1. White Box Testing

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

2.Basic Path Testing

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

$$V(G) = E - N + 2 \text{ or}$$

$$V(G) = P + 1 \text{ or}$$

$$V(G) = \text{Number Of Regions}$$

Where $V(G)$ is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

3. Conditional Testing

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

4. Data Flow Testing

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The *definition-use chain* method was used in this type of testing. These were particularly useful in nested statement

5. Loop Testing

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

- All the loops were tested at their limits, just above them and just below them.
- All the loops were skipped at least once.
- For nested loops test the inner most loop first and then work outwards.
- For concatenated loops the values of dependent loops were set with the help of connected loop.
- Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated.

10.2 Test Report

Test Case for Login Page

Test Case ID /Name	Test Case Description	Expected Result	Actual Result	Test Case Results
1	Login as admin /Customer with incorrect login details	Error message will be occurred	Error message will be occurred	Pass
2	Login as admin with valid user id and password	Depend on given user id and password the admin account is open	Administrator logged into the application	Pass
3	Login as Customer with valid user id and password	Depending on given user id and password Customer account will be open	Customer logged into his/her account	Pass
4	Check Whether the user interface are professionally designed or not	User interface showed be designed professionally	User interface are designed professionally	Pass

Test Case for Reports :

Test Case ID /Name	Test Case Description	Expected Result	Actual Result	Test Case Results
1	Check whether the report come to Administrator or not when units status is in progress	Report should not display to the administrator	The report displays to the Administrator	Fail
2	Check whether the report come to administrator or not when status is updated	Depend on the customer name report should be generated and show to the admin	Report is displayed to administrator	Pass

11. Conclusion

The “**HYPERSPACE BANK SOLUTIONS**” was successfully designed and is tested for accuracy and quality. During this project we have accomplished all the objectives and this project meets the needs of the organization. The developed will be used in searching, retrieving and generating information for the concerned requests.

Benefits

The project is identified by the merits of the system offered to the user. The merits of this project are as follows: -

- ✓ Reduced entry work.
- ✓ Easy retrieval of information
- ✓ Reduced errors due to human intervention
- ✓ User friendly screens to enter the data
- ✓ Portable and flexible for further enhancement
- ✓ Web enabled.
- ✓ Fast finding of information requested

Limitations

- The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.
- Training for simple computer operations is necessary for the users working on the system.

12. Bibliography

References for the Project Development Were Taken From the following Books and Web Sites.

FOR PHP:

- PHP5 and MySQL by W Jason Gilmore (Apress)
- PHP Web Development by Allan Kent and David Powers (Apress)
- Professional PHP4 by Christopher Scollo
- PHP 5 Unleashed by John M. Coggeshall
- Advanced PHP Programming by George Schlossnagle
- Programmings PHP by Kevin Tatroe
- Zend PHP Certification Study Guide by Zend Technologies

FOR MySQL:

- MySQL (2nd Edition) (Developer's Library) by Paul Dubois
- Php and MySql---Larry Ullman

FOR JavaScript:

<http://www.w3schools.com/>

<http://www zend com/php/beginners/>

<http://www.devshed.com/c/b/PHP/>

<http://www.phpfreaks.com/>