**Chapter 1**

**Introduction**

* 1. **Introduction of the System**

It’s an online web application for 24x7 Bank Association which stores all the employee records of 24x7 Bank. Using this web application, Employee Association Management provides training materials for all employees to improve their knowledge, post latest news and announcements. Employees can discuss with other employees and the management in the discussion forum, and also the employees can compose or send mail and view the received mails.

**1.1.1 Project Title**

Employee Association System

**1.1.2 Project Category**

This project is a web based application comes under the **Relational Database Management System (RDBMS).** This application is developed with the help of PHP and MySQL Server.

**1.1.3 Overview**

In smaller businesses, if employees have any problems they can talk directly to their manager. However, in larger businesses that employs many people; it becomes extremely hard to do so. It is also hard for the management to make decisions when they have about 1000 employees. Employee Association System project saves the management’s lot of time because they do not have to meet individual employees to discuss problems.

This project is developed for 24x7 Bank and it provides several benefits for 24x7 Bank employees association. The benefits of this system are:

* Track employee records and experience details
* Association can provide training for employees improve the knowledge.
* Employee performance evaluations.
* Discussion forum –Discuss with employees and management
* Association can post latest news and announcements
* Employees can compose or send mail and they can also view the received mails.
  1. **Background**
     1. **Introduction of the Bank**

24x7 Bank Rajaji nagar Branch,

Padmamba Complex, Rajaji nagar,

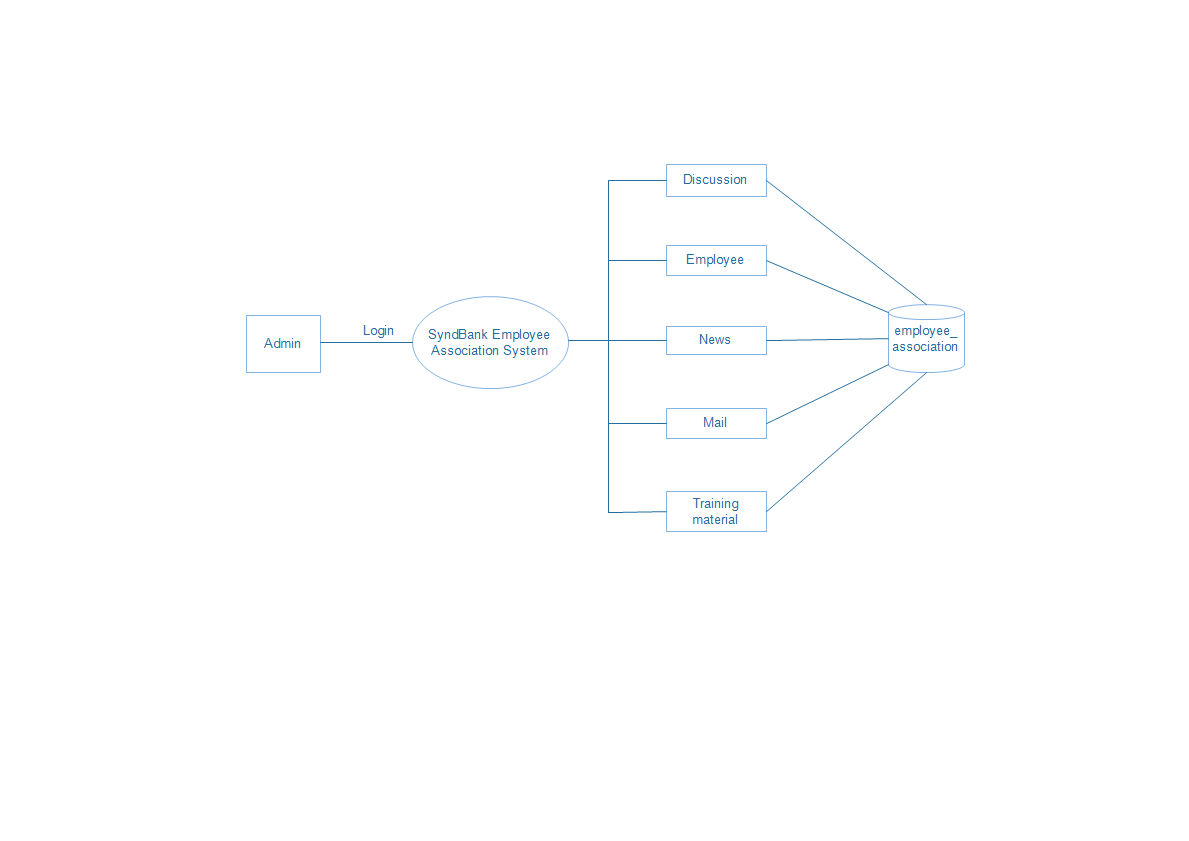
Bangalore D.K.

* + 1. **Brief note on Existing System**

The 24x7 Bank employees maintain their records in their system. For training purpose, the trainees have to go to Manipal to complete their training and then join the Bank. They don’t have the opportunity to view the latest news and events related to the Bank. If they want to discuss their bank related issues with other employees and the management, it’s hard for the management to take a decision to solve their problems.

Employee Association System project saves the management’s lot of time because they do not have to meet individual employees to discuss their issues; since a discussion forum is provided for them. The Association provides training materials to the new employees to improve their knowledge. Employees can view the latest news and announcements related to the Bank.

* 1. **Objectives of the System**
* The bank administrator can add branch details, designation details, discussion details and training details and employee records and also can view the same.
* The employees can view latest news and announcements related to the bank.
* The employees can compose or send mail and also view theirreceived mail.
* The Association provides training materials for the employees to improve their knowledge.
  1. **Scope of the System**
* SyndBank Employee Association System project saves the management’s lot of time because they do not have to see individual employees to discuss problems.
* This system will store all the employee records who work in the Bank.
* This system provides training materials for employees. Management can upload videos, documents, pdf files, etc.
* Employees can discuss with other employees in the discussion forum.
* Employees can compose or send mail and they can also view received mails.
* Employees can view latest news and events published by management.
  1. **Structure of the System**



**Fig 1.1 System Structure**

* 1. **System Architecture**

****

**Fig 1.2 System Architecture**

* 1. **End Users**

24x7 Bank

* 1. **Software/Hardware used for the development**
     1. **Hardware Requirements**

|  |  |
| --- | --- |
| **Processor** | Pentium Dual Xenon Processor |
| **RAM** | Minimum 512 MB |
| **Hard Disk** | Minimum 4. GB |
| **Operating System** | Windows XP/ Windows 7/ Windows 8 |

**Table 1.1: H/W Requirements**

* + 1. **Software Requirements**

|  |  |
| --- | --- |
| **Software** | XAMPP Software v 3.2.1 |
| **Server** | Apache Server 1.8.2 |
| **Database Server** | MySQL database Server 5.5 |
| **IDE** | Adobe Dreamweaver CS 6.0 |

**Table 1.2: S/W Requirements**

**1.9 Software/Hardware required for the implementation**

**1.9.1 Hardware Requirements**

|  |  |
| --- | --- |
| **Processor** | Pentium Dual Xenon Processor |
| **RAM** | Minimum 512 MB |
| **Hard Disk** | Minimum 4. GB |
| **Operating System** | Windows XP/ Windows 7/ Windows 8 |

**Table 1.3: H/W Requirements**

**1.9.2 Software Requirements**

|  |  |
| --- | --- |
| **Software** | XAMPP Software v 3.2.1 |
| **Server** | Apache Server 1.8.2 |
| **Database Server** | MySQL database Server 5.5 |
| **IDE** | Adobe Dreamweaver CS 6.0 |

**Table 1.4: S/W Requirements**

**Chapter 2**

**Software Requirement Specification**

**2.1 Introduction**

The SRS typically contains a brief description of the project. It describes what the proposed software should do without describing how the software will do it. A SRS provides a reference for the validation of the final product, i.e, the SRS helps the client to determine if the software meets the requirements. SRS is the medium through which the client and user needs are accurately specified to the developer.

The basic objective of SRS is to specify the important requirements of the proposed system that are gathered during the system analysis. To specify the basic goals, SRS should have some desirable characteristics: correctness, completeness, unambiguous, verifiable, modifiable and traceable which we have observed throughout during the process in entirely.

This document provides details about the entire Software Requirement Specification for Employee Association System.

**2.2 Overall Description**

This gives an overview of the functionality of the system. The clear understanding of the SyndBank Employee Association System and its functionality will allow for the correct software to be developed for the end user and will be used for the development of the future stages of the project. This system helps us to manage all the transactions to be done in an easily sorted manner. User friendliness is provided in the web application with various controls provided by Rich User Interface. This is a web application and it’s developing using PHP language. All the records store in MySQL Database.

**2.2.1 Product Perspective**

This project is an independent and is not the part of any other application. Our goal is to develop a replacement to the current Employee Association System used in 24x7 Bank; making it more userfriendly. Each user contains entries with unique login id and password provided by the Admin.

**2.2.2 Product Functions**

The major functions of the product are:

* **Employee module** – This module will store complete details of the employees. Here administrator can add employee records by entering their profile details.
* **Branch module** – This will store the details of the branch.
* **Designation module** - This will store the details of the designation.
* **Discussion module** –Here employees can discuss with other employees as well as management in the discussion forum.
* **News and Events module** – Administrator can post news and events in this module. Here the employees can view the latest news and events published by management.
* **Training material module** – This will provide training for employees to improve their knowledge. The management can upload training videos and documents in this module.
* **Subscription module** – Here the employees can pay their annual subscription charge by entering debit card and credit card details. After the subscription the employees can participate in discussion forum, and they can view training materials sent by management.
* **SMS and Email module**– The new employee will receive SMS and Email after the creating new employee account.

## User Characteristics

The new system shall provide a very intuitive and simple interface to the employees and the administrator so that the user can easily navigate through different pages. The SyndBank Employee Association can easily manage their employee records.

* **Administrator**: The administrator of the Bank is allowed to access all the services in the system. The username and password for the end user is given by the administrator. He has the full authority to edit web pages, delete and update them
* **Employee**: The employee is allowed to access the services given by the administrator like making discussions, viewing mails, latest news and events etc.
  + 1. **General Constraints**
* Designed to run on Windows XP and further versions.
* Latest browsers like Internet Explorer or Mozilla Firefox or Google Chrome are required
* Internet facility is essential to use this web application
  + 1. **Assumptions**
* The code should be free with compilation errors/syntax errors.
* The product must have an interface which is simple enough to understand.
* All necessary hardware and software are available for implementing and use of the tool.
* Although basic password authentication and role based security mechanisms will be used to protect from unauthorized access.
* Redundant Database is setup as the role of backup Database Server when primary database is failure

**2.3 Special Requirements**

**2.3.1** **Hardware Interface (Minimum)**

* **Operating System:** Windows XP/ Windows 7/ Windows 8
* **Hard Disk:** 4 GB
* **RAM:** 512 MB
* **Processor:** Pentium Dual Xenon Processor
  + 1. **Software Interface (Minimum)**
* **Design Interface:** HTML, CSS, AJAX
* **Front End:** PHP
* **Back End:** MySql Server 5.5
* **Server:** Apache Server 1.8.2
* **IDE:** Adobe Dreamweaver CS 6.0
  1. **Functional Requirements**
     1. **Module 1: Subscription**
* **Description and Priority**

This module is made for the 24x7 Bank Employees to add and view Subscription records. Every employee will have to pay a subscription charge. Employees can access website features after making the payment. This module will be a higher priority based module.

* **Stimulus/Response Sequences**

When the user clicks the link from the account page, the control will be redirected to this Subscription page. Here it accepts all the necessary field values from the user and when the user clicks the submit button, then it will validate the entered data and if it is correct then it stores the data.

* **Functional Requirements**

To Store the Information into the database:

Here this must add the Subscription details to the database and this must insert the fields like Employee Name, Subscription Charge, Start Date, End Date and Status of the Subscription .If all the entered fields are correct then it must be added to the database.

To Retrieve the Information from the database:

Here this has to display all the Subscribers and it must contain the details like Employee Name, Subscription Charge, Start Date, End Date and Status of the Subscription. This must be shown in a neat format so that the Individual records can be edited or deleted.

**2.4.2 Module 2: Account**

* **Description and Priority**

Employees must login to the website to access account module. This module is for the 24x7 Bank employees and the administrator where they can login to the website by entering login id and password. After login they can change their password and profile details. Each employee has a unique login id and password. Employees whose login id and password are registered are entitled for login. This will be a medium priority based module.

* **Stimulus/Response Sequences**

After selecting login option by the user the system prompts the user to enter the login id and passwords. If it is correct then it redirects to account page. Otherwise it shows an error message. Further it will also provide forgot password option if the user has forgotten the password. It can be recovered through Email id of the employee. And also the user can change their password if they want to change their current password.

* **Functional Requirements**

Logging of a user in the software:

This will check whether the entered Login ID and Password of a user will be listed in the database. If it is correct then it will allow the user to enter into the next page. Or else it won’t allow the user to access the next page. Once after logging only user can perform all Bank related operations.

## Module 3: Discussion

* **Description and Priority**

This module is made for the 24x7 Bank Employees to discuss with other employees in the discussion forum.This discussion forum could be an ideal place to establish communication and it keeps all the discussion records for future reference.This will be a higher priority based module.

* **Stimulus/Response Sequences**

When the user clicks the link from the account page, the control will be redirected to this Discussion page. Here it accepts all the necessary field value from the user and when the user clicks the submit button then it will validate the entered data and if it is correct then it stores the data.

* **Functional Requirements**

To Store the Information into the database:

Here this must post new Discussion forum details to the database and this must insert the fields like Employee Name, Category, Discussion Title, Description, Discussion Image, Publish Date, Uploads and Status of the Discussion .If all the entered fields are correct then it must be added to the database.

To Retrieve the Information from the database:

Here this has to display all the Discussion forum details and it must contain the details like Employee Name, Category, Discussion Title, Description, Discussion Image, Publish Date, Uploads and Status of the Discussion .This must be shown in a neat format so that the Individual records can be edited or deleted.

## Module 4: News and Announcement

* **Description and Priority**

This module is made for the 24x7 Bank Employees to view the latest news and announcements.In this module the management can post latest news and they can post events or meeting schedules .This will be a higher priority based module.

* **Stimulus/Response Sequences**

When the user clicks the link from the account page, the control will be redirected to this News page. Here it accepts all the necessary field value from the user and when the user clicks the submit button then it will validate the entered data and if it is correct then it stores the data.

* **Functional Requirements**

To Store the Information into the database:

Here this must publish news details to the database and this must insert the fields like Employee Name, News Type, News Category, Publish Date, News Title, Content, Image and Status of the News .If all the entered fields are correct then it must be added to the database.

To Retrieve the Information from the database:

Here this has to display all the latest news details and it must contain the details like Employee Name, News Type, News Category, Publish Date, News Title, Content, Imageand Status of the News .This must be shown in a neat format so that the Individual records can be edited or deleted.

## Module 5: Training Material

* **Description and Priority**

This module is to provide training for the 24x7 Bank Employees to improve their knowledge. In this module it contains training materials uploaded by management. This training material will help the new employees as well as the existing employees. This will be a higher priority based module.

* **Stimulus/Response Sequences**

When the user clicks the link from the account page, the control will be redirected to this Training material page. Here it accepts all the necessary field value from the user and when the user clicks the submit button then it will validate the entered data and if it is correct then it stores the data.

* **Functional Requirements**

To Store the Information into the database:

Here this must upload training material details to the database and this must insert the fields like Employee Name, Title, Training Category, Image Link, Video Link, Content and Status of the Training material .If all the entered fields are correct then it must be added to the database.

To Retrieve the Information from the database:

Here this has to display all the training material details and it must contain the details like Employee Name, Title, Training Category, Image Link, Video Link, Content and Status of the Training material. This must be shown in a neat format so that the Individual records can be edited or deleted.

## Module 6: Mail

* **Description and Priority**

This module is for the 24x7 Bank Employees to compose or send mail and they can also view the received mails. It has 3 sub modules, i.e; Inbox, Compose message, and sent messages. In this module employees can send mails to the management. This will be a medium priority based module.

* **Stimulus/Response Sequences**

When the user clicks the link from the account page, the control will be redirected to this mail page. Here it accepts all the necessary field value from the user and when the user clicks the submit button then it will validate the entered data and if it is correct then it stores the data.

* **Functional Requirements**

To Store the Information into the database:

Here this must add mail details to the database and this must insert the fields like Employee Name, Subject, Message and Attachment of the mail. If all the entered fields are correct then it must be added to the database.

To Retrieve the Information from the database:

Here this has to display all the mail details and it must contain the details like Employee Name, Subject, Message and Attachment of the mail. This must be shown in a neat format so that the Individual records can be edited or deleted.

## Module 7: Dashboard

* **Description and Priority**

This module is for the management where management can control all the activities of Syndbank Employee association system. This will be a medium priority based module.

* **Stimulus/Response Sequences**

When the user clicks the link from the account page, the control will be automatically redirected to this dashboard page. Only the authorized user can enter this page otherwise if the user enters a wrong password then it won’t allow the user to access dashboard page.

* 1. **Design Constraints**
* GUI is only in English.
* Login and password is used for the identification of users.
* Only employees and management will be authorized to use the services.
* Limited to HTTP/HTTPS.
* This system is working for single server.
  1. **System Attributes**
* **Correctness:** Error message will occur if the input is invalid. Each message should be validated, so it will give a proper message when error occurred.
* **Maintainability:** There will be no maintenance required for the software. The database is provided by the user and therefore is maintained by the user.
* **Reliability:** Reliability of the whole system will depend on the reliability of its parts.
* **Security:** Only authorized person can access the system.
* **Availability:** It is possible to update the software
* **Adaptability:** Thesoftware is adapted to any changes made by it.
* **Testability:** Different testing mechanisms can be performed.
  1. **Other Requirements**

**2.7.1 Safety Requirements**

The user authentication and authorization protocols have to be tight. In case of server default or crash due to virus of operating system failure, the system will do periodic backups through a live internet connection.

**2.7.2 Security Requirements**

The user authentication and authorization protocols have to be secure. The database contains encrypted login credentials. Administrator and the end user have only the right to open the software. They have their own user name and password.

**Chapter 3**

**System Design**

**3.1 Introduction**

The purpose of the Design Phase is to a solution of the problem specified in the requirements document. This is the first step in moving from problem domain to the solution domain. The design of a system is perhaps the most critical factor affecting the quality of the software. It has a major impact on the later phase - the Design Document. This Document is similar to the blueprint or a plan for the solution and is used later during implementation, testing and maintenance.

The Design activity is often divided into 2 separate phases – System Design and Detailed Design. System Design is sometimes also called Top-level Design. This system design aims to identify the modules that should be in the system, the specification of these modules, and how they interact with each other to produce the desired result. At the end of the System Design all the major data structures, file formats, and the major modules in the system and their specification are decided.Software design process through which requirements are translated into a representation of the software.

**3.2 Assumptions and Constraints**

**3.2.1 Assumptions**

* The system uploads all php files in the server.
* This web portal is user friendly and less cost because it works in apache server.
* This is aimed to provide satisfactory result to the user by providing the required information.
* Administrator is created in the system manually.
* All types of user need to enter their login id and password for authentication during login.
* The card payment is virtual payment system and it does not work like real payment system

**3.2.2 Constraints**

* More space is required to keep all the records.
* Database should not be overloaded.
* The tables of the database are designed as normalized table.
* Database information is not shared with clients.

**3.3 Functional decomposition**

Functional decomposition refers broadly to the process of resolving a functional relationship into its constituent parts in such a way that the original can be reconstructed from those parts by function components. In general this process of decomposition is undertaken either for the purpose of gaining insight into the identity of the constituent components (which may reflect individual physical process of interest) or for the purpose of obtaining a compressed representation of the global function, a task which is feasible only when the constituent processes possess a certain level of modularity (i.e. independence or non-interaction).

**3.3.1 System Software Architecture**

The term Software Architecture intuitively denotes the high level structure of a software system. It can be defined as the set of structures needed to understand the software system, defined the software elements, the relations between them and the properties of both elements and relations.

Software Architecture also denotes the set of practices used to select, define or design the software architecture.



**Fig 3.1 System Software Architecture**

## 3.3.2 System Technical Architecture

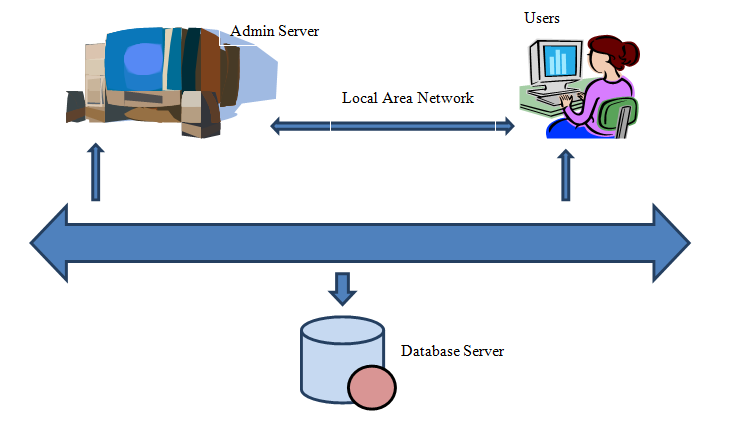
A Technical Architecture is the conceptual model that defines the structure, behavior and more views of a system.

An architectural description is a formal description and representation of a system, organized in a way that supports reasoning about the structure of the system which comprises system components, the externally visible properties of those components, the relationships between them and provides a plan from which products can be procured and systems developed, that will work together to implement the overall system.



**Fig 3.2 System Technical Architecture**

## 3.3.3 System Hardware Architecture

 **Fig 3.3 System Hardware Architecture**

**3.4 Description of Programs**

**3.4.1 Context Flow Diagram (CFD)**

Context flow diagram is a top level(also known as level 0) data flow diagram. It contains only one process node that generalizes the functions of the entire system in relationship to external entities. In context diagram the entire system is treated as single process and all its inputs, outputs, sinks and sources are identified and shown below:

****

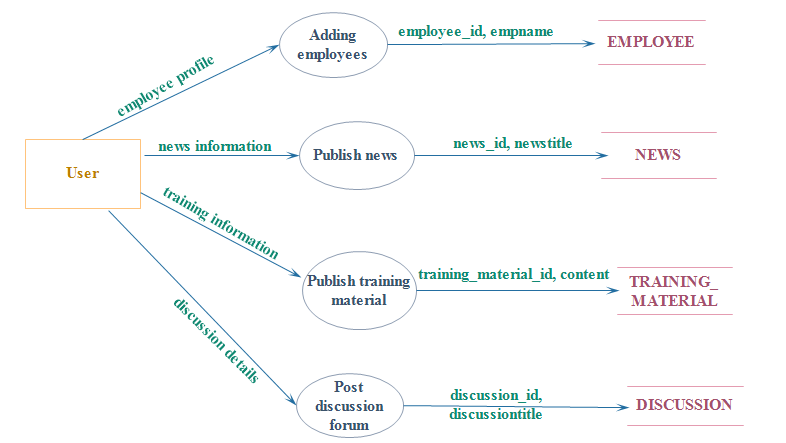
**Fig 3.4 Context Flow Diagram**

It only contains one process node that generalizes the function of the entire system in relationship to external entities. In context diagram the entire system is treated as a single process and all its inputs, outputs, sinks and sources are identified and shown.

**3.4.2 Data Flow Diagrams (DFDs)**

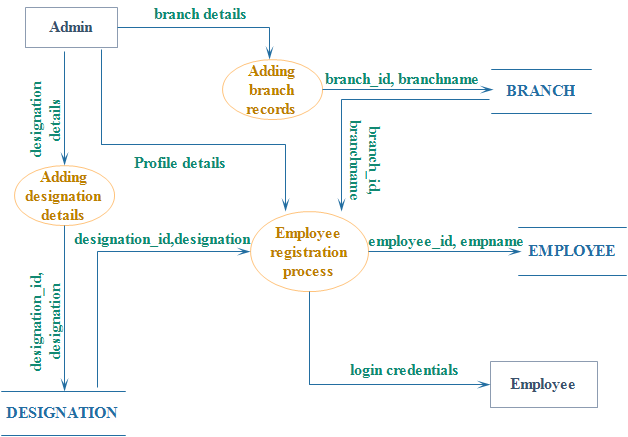
Data Flow Diagram is a graphical representation of a system or a portion of the system. It consists of data flow, process, sources and sink and stores all the description through the use of easily understandable symbols. DFD is one of the most important modelling tools. It is used to model the system, components that interact with the system, uses the data and information flows in the system. DFD shows the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information moves from input or output. DFD is also known as bubble chart or Data Flows Graphs. DFD may be used to represent the system at any level of abstraction.

**Top Level DFD (Level 0 DFD):**

****

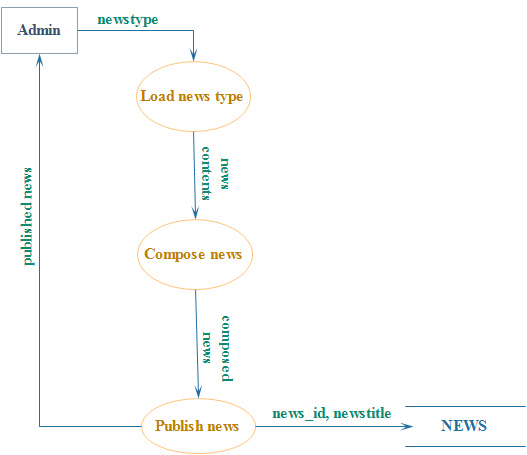
**Fig 3.5 Top Level DFD**

**DFD Level 1: Employee registration process**

****

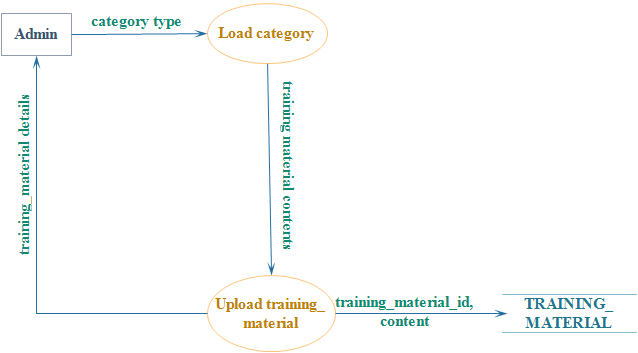
**Fig 3.6 Level 1 DFD**

**DFD Level 2: Publish news**



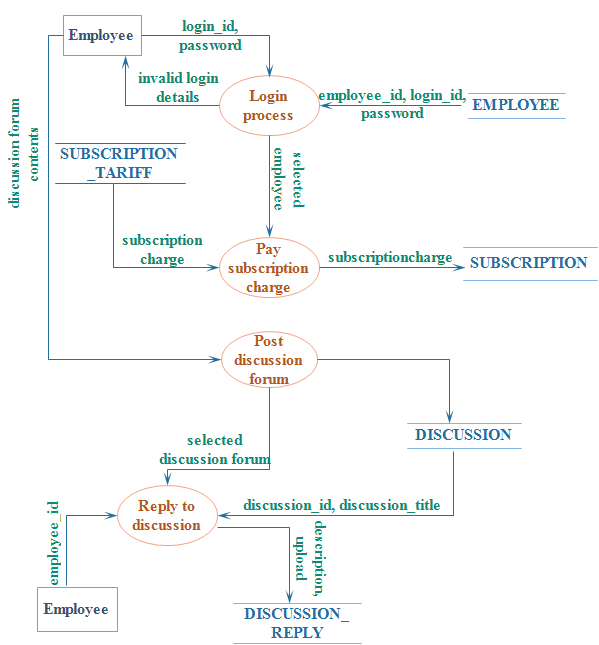
**Fig 3.7 Level 2 DFD**

**DFD Level 3: Upload training material**

****

**Fig 3.8 Level 3 DFD**

**DFD Level 4: Post discussion forum**

****

**Fig 3.9 Level 4 DFD**

* 1. **Description of Components**
     1. **Functional Component 1: Employee registration process**
* **Input:** Adding branch details like branch name, notes and status of the employee, adding designation details like designation, description and status of the employee and adding employee details like employee name, branch, designation, login name, password, confirm new password, gender, date of birth, address, country, city, state, pin code, date of joining, date of retirement, note, image and status of the employee
* **Process:** Here the system checks the entered branch details like branch name, notes and status of the employee, entered designation details like designation, description and status of the employee and entered employee details like employee name, branch, designation, login name, password, confirm new password, gender, date of birth, address, country, city, state, pin code, date of joining, date of retirement, note, image and status of the employee. Hence the system checks whether the entered information of the form is valid or not.
* **Output:** The system inserts employee record, designation record and branch records in the database. If the entered information is valid then the user gets a message stating that “Record inserted successfully..” Hence the user gets the acknowledgement that whether the transaction is successful or not.
  + 1. **Functional Component 2: Publish news**
* **Input:** Entering the news details like news content, news type, news title, news publish date
* **Process:** Here the system checks the entered news content, news type, news title, news publish date. It also checks whether the validation of the entered information is correct or not.
* **Output: :** If entered data valid then the system inserts the training materials to the database The system publish news on published date and the user get to know about the latest news published on a particular published date. The employee can view the latest news and events published

**3.5.3 Functional Component 3: Upload training material**

* **Input:** Select category type and upload training materials details like Employee Name, Title, Training Category, Image Link, Video Link, Content and Status of the training
* **Process:** The system checks the entered training material details like Employee Name, Title, Training Category, Image Link, Video Link, Content and Status. Hence it checks that the entered information of the form is uploaded or not.
* **Output:** If the entered data is valid then the system inserts the training materials to the database. Hence the user gets the acknowledgment stating “Record uploaded successfully..” and the employee can view the training details in the database

**3.5.4 Functional Component 4: Post discussion forum**

* **Input:** Employee login to the system through login id & password and post new discussion forum details like Employee Name, Category, Discussion Title, Description, Discussion Image, Publish Date, Uploads and Status of the Discussion
* **Process:** The system checks the entered post new discussion forum details like Employee Name, Category, Discussion Title, Description, Discussion Image, Publish Date, Uploads and Status. Hence the system checks whether the entered information valid or not.
* **Output:** If the entered discussion form is valid then the system inserts the new discussion records to the database and the user gets an acknowledgement stating the “Record inserted successfully…”. Hence the employees can view the discussion forum.

**Chapter 4**

**Database Design**

**4.1 Introduction**

Database Design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters. Needed to generate a design in a data definition language, which can be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of the logical design of the relation to the base data structure used to store the data. In the relational model these are the classes and the named relationships.However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the [database management system](http://en.wikipedia.org/wiki/Database_management_system) (DBMS).

**4.2 Purpose and Scope**

**4.2.1 Purpose**

Database description describes the entire database used in the software to store all the records. The database in turn is further described in detail giving all the fields used with their data type, constraints include primary key, foreign key, etc., which allow the entities to be uniquely identified.

This database requirement specification describes the function and performance requirements of the employee\_association database. This database stores records such as employee details, branch details, discussion details, designation details, subscription details, news details, training material details and mail details.

The purpose of this database specification (DS) for the Coconut SyndBank Employee Association System.is to provide

* A description of the Employee Association database, including database identification, interfaces, organization, mapping of database files, storage allocation, and physical allocation;
* Database administration information, including database management system (DBMS) configuration, hardware configuration, database software utilities, database physical structure and sizing, security; and
* A description of applications software requirements, including information needed in the development, maintenance, and enhancement of the software.

**4.2.2 Scope**

A good database is one that is simple to understand and well planned. The database doesn’t have redundant tables. One can use ER Diagram (Entity Relationship Diagrams) in order to make a good database. This database design is used to understand the software Coconut Processing Inventory System.

* Organize the system into modules.
* Organize sub-modules for each module.
* Allocate tasks to processors.
* Choose an approach to manage data store.
* Handle access to global resources.
* Choose implementation logic.

**4.3 Database Identification**

Database table name and column names defined without leaving space. Lower case used to create database tables and columns. Primary key and foreign key defined with same name. And the table used in the database are as follows:

* employee
* subscription
* branch
* designation
* discussion
* mail
* news
* subscription\_tariff
* discussion\_reply
* training\_material

**4.4 Schema Information**

## Table :branch

|  |  |  |  |
| --- | --- | --- | --- |
| **branch\_id** | **Branchname** | **description** | **status** |

## Table :designation

|  |  |  |  |
| --- | --- | --- | --- |
| **designation\_id** | **designation** | **description** | **status** |

## Table : subscription

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **subscription\_id** | **employee\_id** | **subscriptioncharge** | **startdate** | **enddate** | **status** |

## Table : employee

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **emp\_id** | **branch\_id** | **designation\_id** | **empname** | | **dob** | **gender** | **login\_id** |
| **password** | empimage | **address** | | **doj** | **dor** | **notes** | **status** |

## Table :subscription\_tariff

|  |  |  |
| --- | --- | --- |
| **subscription\_tariff\_id** | **designation\_id** | **subscriptioncharge** |

## Table :discussion

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **discussion\_id** | **employee\_id** | **Discussiontitle** | **category** | **discussionimage** | **uploads** |
| **description** | **Publishdate** | **Status** |

## Table :discussion\_reply

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **reply\_id** | **discussion\_id** | **employee\_id** | **description** | **uploads** | **publishdate** | **status** |

## Table : mail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **mail\_id** | **employee\_id1** | **employee\_id2** | **messagetitle** | messagecontent | attachments |
| senddate | reply\_id | status1 | status2 |

## Table :news

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **news\_id** | **employee\_id** | **newstype** | **newscontent** | **newstitle** | **newsdate** | **Newscategory** |
| **newsimage** | **Status** |

**Table :training\_material**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| training\_material\_id | | employee\_id | title | content | training\_category | image\_link |
| video\_link | Status |

**4.5 Table Definition**

## Table Name: branch

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **branch\_id** | int(10) | Primary Key | Branch unique ID |
| branchname | varchar(25) | Not Null | Branch name |
| description | Text | Null | Branch detail |
| Status | varchar(10) | Not Null | Branch status |

## Table 4.1 branch

## Table Name: designation

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **designation\_id** | int(10) | Primary Key | Designation unique ID |
| designation | varchar(25) | Not Null | Designation |
| description | Text | Null | Designation detail |
| Status | varchar(10) | Not Null | Designation status |

## Table 4.2 designation

## Table Name: discussion

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **discussion\_id** | int(10) | Primary Key | Discussion unique ID |
| employee\_id | int(10) | Foreign Key | Employee unique ID |
| discussiontitle | varchar(250) | Not Null | Discussion title |
| Category | varchar(25) | Not Null | Discussion category |
| discussionimage | varchar(100) | Not Null | Discussion image |
| Uploads | varchar(100) | Not Null | Discussion uploads |
| description | Text | Null | Discussion detail |
| publishdate | Datetime | Not Null | Discussion publish date |
| Status | varchar(10) | Not Null | Discussion status |

**Table 4.3 discussion**

**Table Name: discussion\_reply**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **reply\_id** | int(10) | Primary Key | Reply unique ID |
| discussion\_id | int(10) | Foreign Key | Discussion unique ID |
| employee\_id | int(10) | Foreign Key | Employee unique ID |
| Description | Text | Null | Discussion reply detail |
| Uploads | varchar(100) | Not Null | Discussion reply uploads |
| publishdate | Datetime | Not Null | Discussion reply publish date |
| Status | varchar(10) | Not Null | Discussion reply status |

**Table 4.4 discussion reply**

## Table Name: employee

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **employee\_id** | int(10) | Primary Key | Employee unique ID |
| branch\_id | int(10) | Foreign Key | Branch unique ID |
| designation\_id | int(10) | Foreign Key | Designation unique ID |
| empname | varchar(25) | Not Null | Employee Name |
| Dob | Date | Not Null | Employee date of birth |
| Gender | varchar(10) | Not Null | Employee gender |
| Address | varchar(250) | Not Null | Employee address |
| Country | varchar(25) | Not Null | Employee country |
| State | varchar(25) | Not Null | Employee state |
| City | varchar(25) | Not Null | Employee city |
| Pincode | varchar(10) | Not Null | Employee pincode |
| Doj | Date | Not Null | Employee date of joining |
| Dor | Date | Not Null | Employee date of retirement |
| login\_id | varchar(25) | Not Null | Employee login ID |
| password | varchar(250) | Not Null | Employee password |
| empimage | varchar(100) | Not Null | Employee image |
| Note | Text | Null | Employee note |
| Status | varchar(10) | Not Null | Employee status |

## 

## Table 4.5 employee

## Table Name: mail

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **mail\_id** | int(10) | Primary Key | Mail unique ID |
| employee\_id1 | int(10) | Foreign Key | Employee1 unique ID |
| employee\_id2 | int(10) | Foreign Key | Employee2 unique ID |
| messagetitle | varchar(150) | Not Null | Mail title |
| messagecontent | Text | Not Null | Mail content |
| attachments | varchar(100) | Not Null | Mail attachments |
| senddate | Datetime | Not Null | Mail senddate |
| Status | varchar(10) | Not Null | Mail status |

## Table 4.6 mail

## Table Name: news

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **news\_id** | int(10) | Primary Key | News unique ID |
| employee\_id | int(10) | Foreign Key | Employee unique ID |
| newstype | varchar(25) | Null | News type |
| newstitle | varchar(150) | Null | News title |
| newsdate | Datetime | Null | News date |
| newsimage | varchar(100) | Null | News image |
| newscontent | Text | Null | News content |
| newscategory | varchar(25) | Null | News category |
| Status | varchar(10) | Null | News status |

## Table 4.7news

## Table Name: subscription

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **subscription\_id** | int(10) | Primary Key | **Subscription unique ID** |
| employee\_id | int(10) | Foreign Key | Employee **unique ID** |
| Startdate | Date | Not Null | **Subscription senddate** |
| Enddate | Date | Not Null | **Subscription enddate** |
| Subscriptioncharge | float(10,2) | Not Null | **Subscription charge** |
| Status | varchar(10) | Not Null | **Subscription status** |

## Table 4.8 subscription

## Table Name: subscription\_tariff

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **subscription\_tariff\_id** | int(10) | Primary Key | **Subscription tariff unique ID** |
| designation\_id | int(10) | Foreign Key | Designation unique ID |
| Subscriptioncharge | float(10,2) | Not Null | Subscription charge |

## Table 4.9subscription tariff

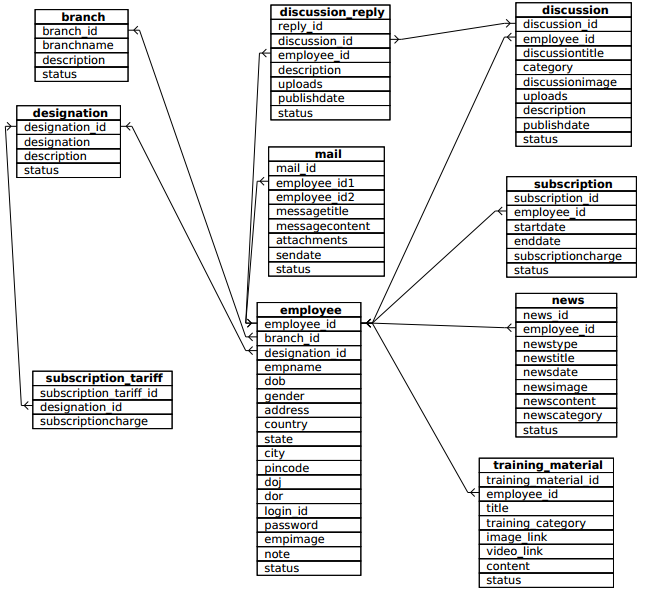
## Table Name: training\_material

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Index** | **Description** |
| **training\_material\_id** | int(10) | Primary Key | **Training material unique ID** |
| employee\_id | int(10) | Foreign Key | **Employee unique ID** |
| Title | varchar(150) | Not Null | **Training material title** |
| training\_category | varchar(25) | Not Null | **Training material category** |
| image\_link | varchar(100) | Not Null | **Training material image** |
| video\_link | varchar(100) | Not Null | **Training material video** |
| Content | Text | Not Null | **Training material content** |
| Status | varchar(10) | Not Null | **Training material status** |

**Table 4.10 training material**

**4.6 Physical Design**

The primary goal of physical database design is data processing efficiency. We will concentrate on choices often available to optimize performance of database services. Physical Database design requires information gathered during earlier stages of the design process.



**Fig 4.1 Physical Design**

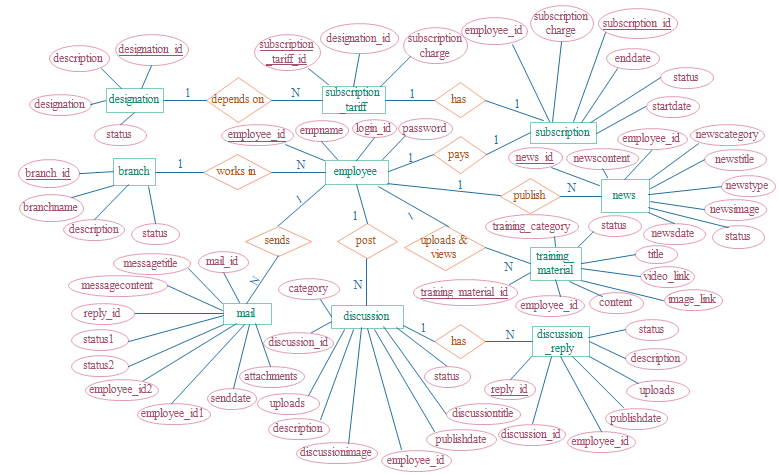
**4.7 Data Dictionary**

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Primary key** | **Foreign key** |
| Employee | employee\_id | branch\_id, designation\_id |
| Branch | branch\_id | - |
| Designation | designation\_id | - |
| Discussion | discussion\_id | employee\_id |
| Subscription | subscription\_id | employee\_id |
| subscription\_Tariff | subscription\_tariff\_id | designation\_id |
| discussion\_reply | reply\_id | employee\_id,discussion\_id |
| Mail | mail\_id | employee\_id |
| training\_material | training\_material\_id | employee\_id |
| News | news\_id | employee\_id |

**Table 4.12 Data Dictionary**

**4.8 ER Diagram**

An Entity Relationship(ER) Diagram is a specialized graphics that illustrates the interrelationship between entities in a database. ER diagram often use symbols to represent three types of information. Boxes are commonly used to represent relationships and ovals are used to represent attributes

****

**Figure 4.2 Entity Relationship Diagram**

**4.9 Database Administration**

**4.8.1 System Information**

The name of the database is “employee\_association”. The MySQL Database Engine is the core service for storing, processing, and securing data. The Database Engine provides controlled access and rapid transaction processing to meet the requirements of the most demanding data consuming applications within your enterprise.

**4.8.2 DBMS Configuration**

The main employee\_association database will be managed by a Database Administrator (DBA). The responsibilities of the DBA include installation of DBMS software components, overall control of the database structure and related utilities, data administration and setting up and monitoring database.

Here we now developed using local host System

* Version: MySQL 5.5.16
* Protocol version:10
* Supported Operating System : Windows 2000, XP, [XP Professional x64](http://en.wikipedia.org/wiki/Windows_XP_Professional_x64_Edition), Linux
* MySQL client version :mysqlnd 5.0.8-dev-20102224-$
* Revision:310735$

**4.8.3 Support software required**

This is a web based application no special software’s are needed in client side only updated browser enough. Here the system installs Mysql server while installing Xampp software. All the backup content stores in mysql data folder while this is loaded in localhost.

**4.8.4 Storage requirements**

If the system crashes, all the information stored in the computer will get destroyed, so we should make some storage requirements by keeping the information in server or storing it in any external storage devices.

## The storage engine represents the heart of a MySQL Server.

* The storage engine has a number of duties including:
* Recovering the database from system failure
* Management of files and database pages used to store data
* Manage data buffers and system IO to the physical data pages
* Manage locking and concurrency issues

**4.8.5 Backup and recovery**

There is no any other special software to recover the lost database, so it is important to keep a backup of database information in external hard disk or local server.

**Chapter 5**

**Detailed Design**

**5.1 Introduction**

The detailed design starts after the system design phase is complete. The main goal of the detailed design is to specify the logic for the different modules that have been specified during the system design or coding for the module can be developed such that specification of the module may be given precisely. Once the module is precisely specified the internal logic for the module that will implement the given specification as decided.

This specification should be such that they are complete unambiguous and precise and they are not suggested any particular implementation. A well defined design language like PDL (Process Design Language) is used for detailed design.

In this project the front end consists of PHP forms. The form design is done using Dreamweaver. We are also using MySQL as backend tool for computerisation of Employee Association System.

**5.2 Applicable Documents:**

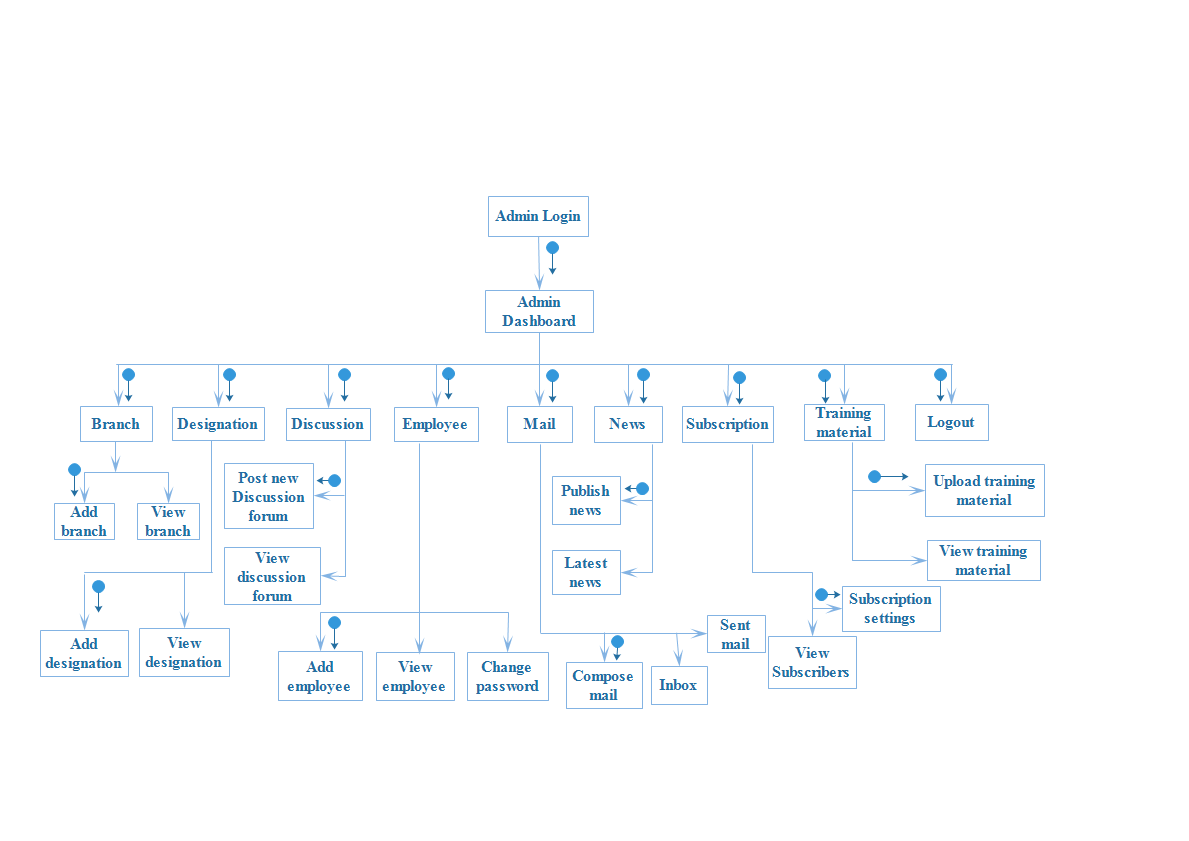
The documents used during detailed design are

* System Requirements Document
* System Design
* Database Design

The detailed design defines the system design document. Hence the first applicable document here is system design. Also we are referring the data structures. Hence the second applicable document is database design. Since this project is user friendly, all users can operate it efficiently. The project helps the user to know the details of the employees working in the bank; the association can post latest news and events and also provide training to improve employee’s knowledge. Also the employee can discuss with other employees and the management in the discussion forum. The admin can have the up to date information about the various transactions made by the account holder. All work that was done by writing in papers can now be stored in database through the computer.

**5.3 Structure of software package:**

**Structure Chart for Admin :**



**Fig 5.1 Structure Chart for Admin**

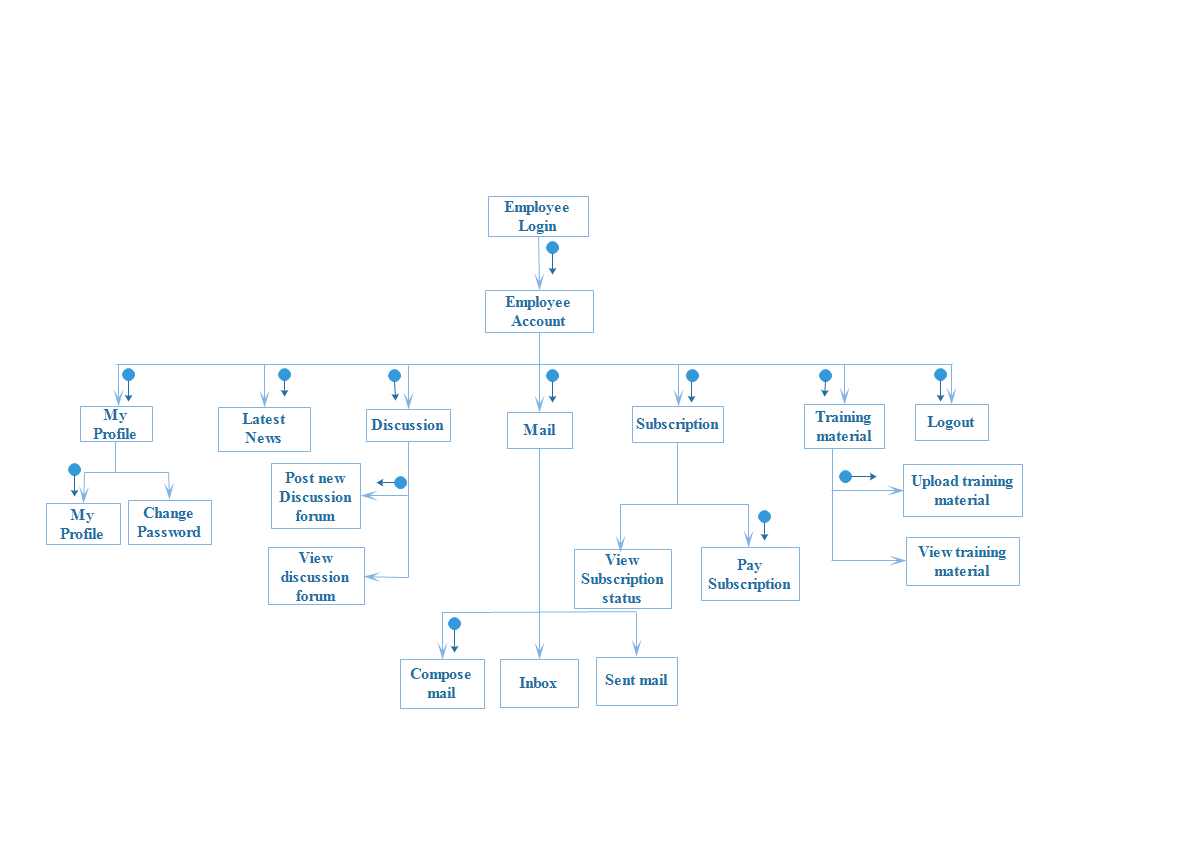
**Module Design**

**Input :** Login ID, Password

**Table :** employee

**Functions :** Used to authenticate users

**Structure Chart for Employee :**



**Fig 5.2 Structure Chart for Employee**

**Module Design**

**Input :** Login ID, Password

**Table :** employee

**Functions :** Used to authenticate users

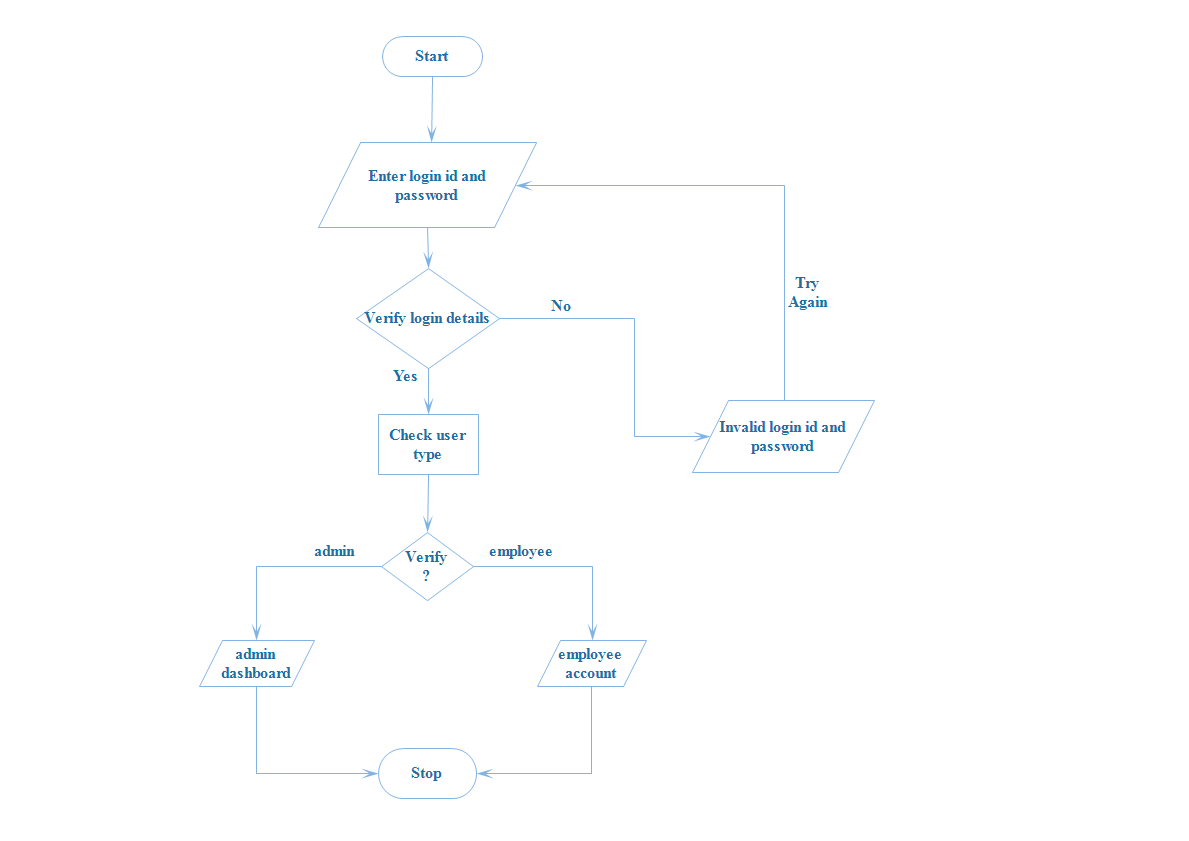
**The various functional components used are listed below :**

* Functional Component 1 : Login page
* Functional Component 2 : Branch details
* Functional Component 3 : Designation details
* Functional Component 4 : Employee details
* Functional Component 5 : Discussion details
* Functional Component 6 : News details
* Functional Component 7 : Subscription details
* Functional Component 8 : Training material details

**5.4 Modular decomposition of the System :**

**5.4.1 Login page :**

* **Input :** Login ID, Password
* **Procedural Details (Flowchart) :**

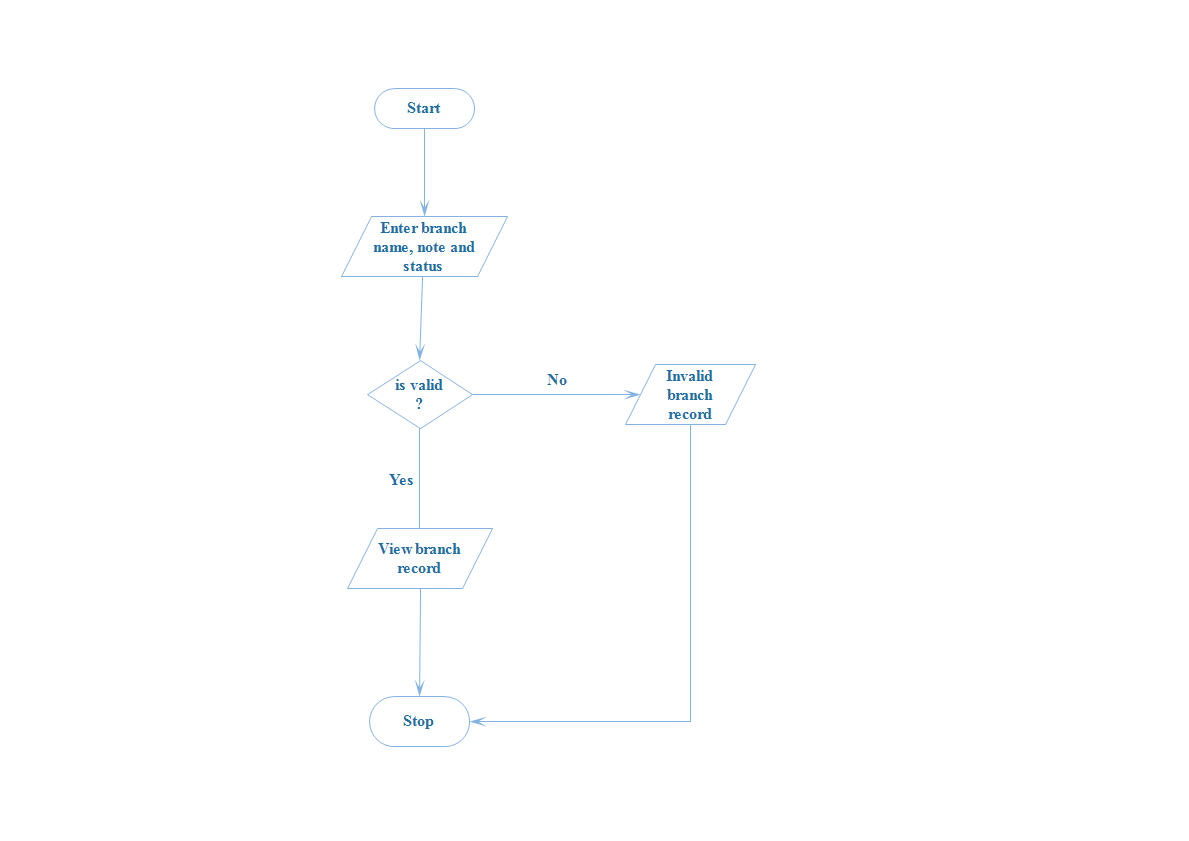


**Fig 5.3 Flowchart for Login Page**

* **File I/O Interface :** Graphical User Interface
* **Output :** If the entered Login ID and Password are valid then it will check whether it is of Admin type or of Employee type.Else if the login details are invalid, it will ask the user to enter the Login ID and Password again. Furtherif the Login ID and Password of Admin are valid, it will move toAdmin Dashboard or else if the entered Login ID and Password isofEmployee, it will move to Employee Account

**5.4.2 Branch Details :**

* **Input :** Branch Name, Notes and Status
* **Procedural Details (Flowchart) :**

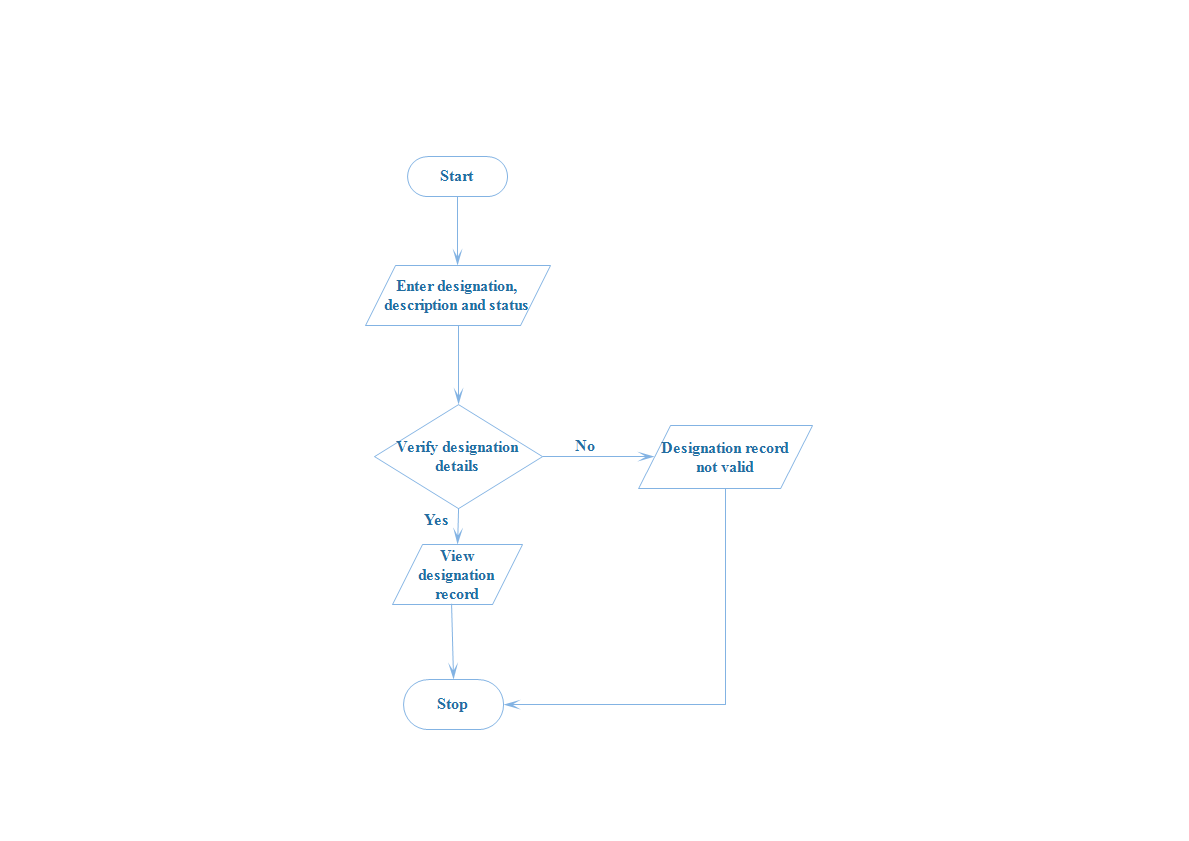


**Fig 5.4 Flowchart for Branch**

* **File I/O Interface :** Graphical User Interface
* **Output :** If the entered Branch details are valid then the data will be stored in the respective database and will be displayed in the Dashboard/Account. Else if the enterd data is not valid, it will display invalid barnch record.

**5.4.3 Designation details :**

* **Input :** Designation, Description and Status
* **Procedural Details (Flowchart) :**

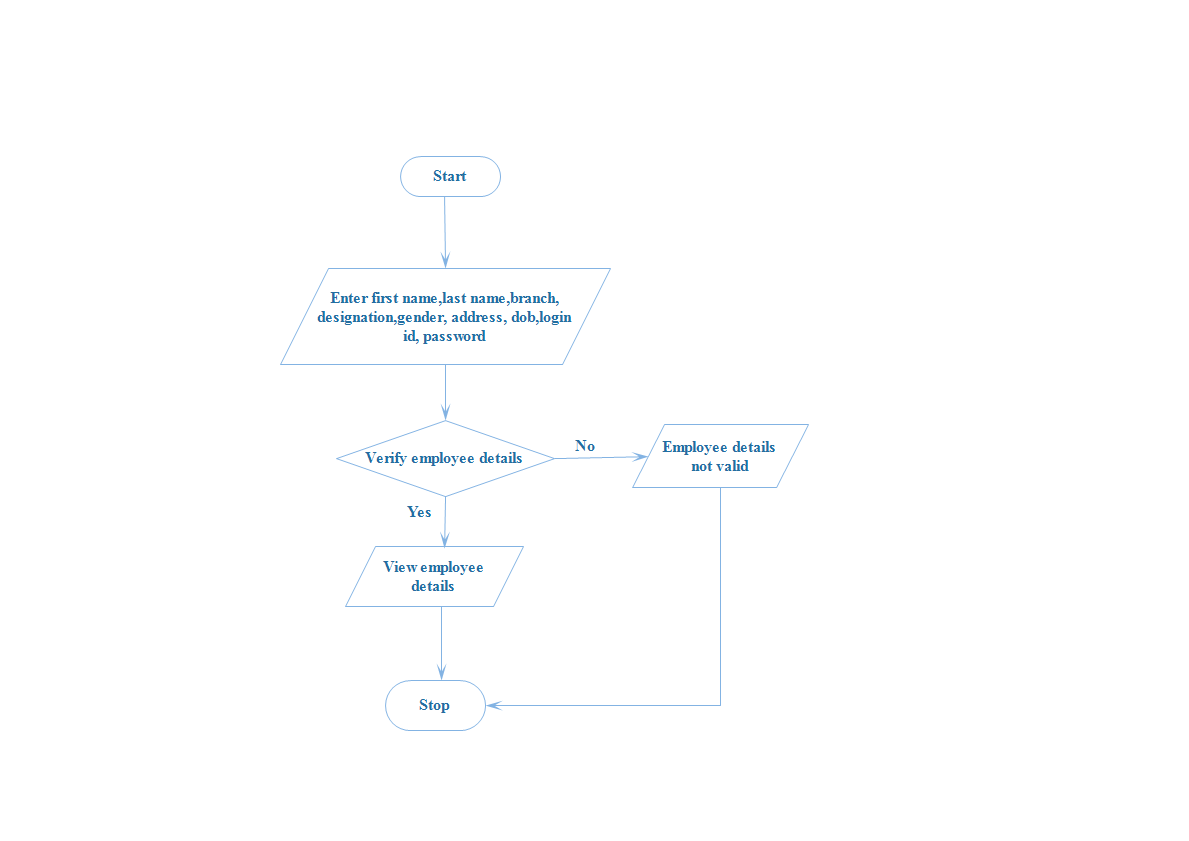


**Fig 5.5 Flowchart for Designation**

* **File I/O Interface :** Graphical User Interface
* **Output :** If the entered Designation details are valid then the data will be stored in the respective database and will be displayed in the Dashboard/Account. Else if the enterd data is not valid, it will display Designation record not valid.

**5.4.4 Employee Details :**

* **Input :** Employee name, branch, designation, login name, password, confirm new password, gender, date of birth, address, country, city, state, pin code, date of joining, date of retirement, note, image and status
* **Procedural Details (Flowchart) :**

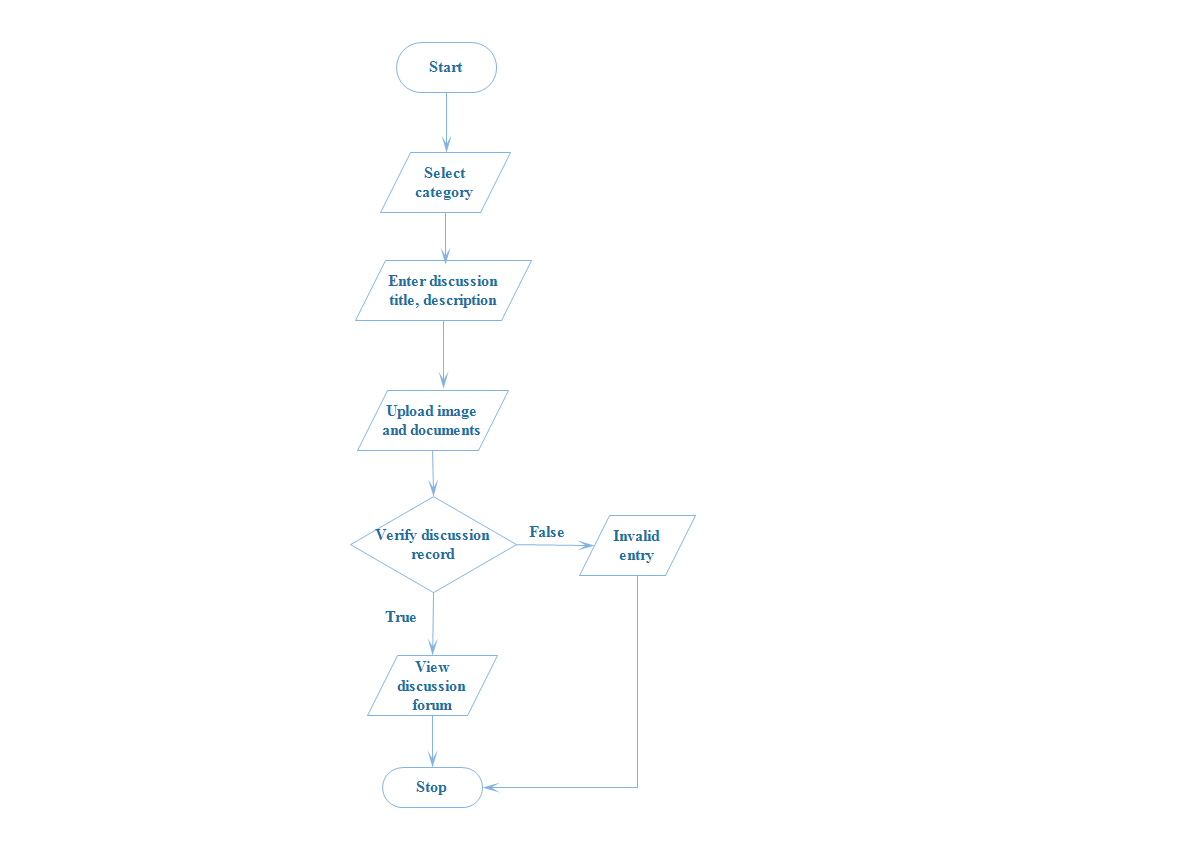


**Fig 5.6 Flowchart for Employee**

* **File I/O Interface :** Graphical User Interface
* **Output :** If the entered Employee details are valid then the data will be stored in the respective database and will be displayed in the Dashboard/Account. Else if the enterd data is not valid, it will display Employee details not valid.

**5.4.5 Discussion Details :**

* **Input :** Category, Title, Description, Discussion Image, Uploads and Status.
* **Procedural Details (Flowchart) :**

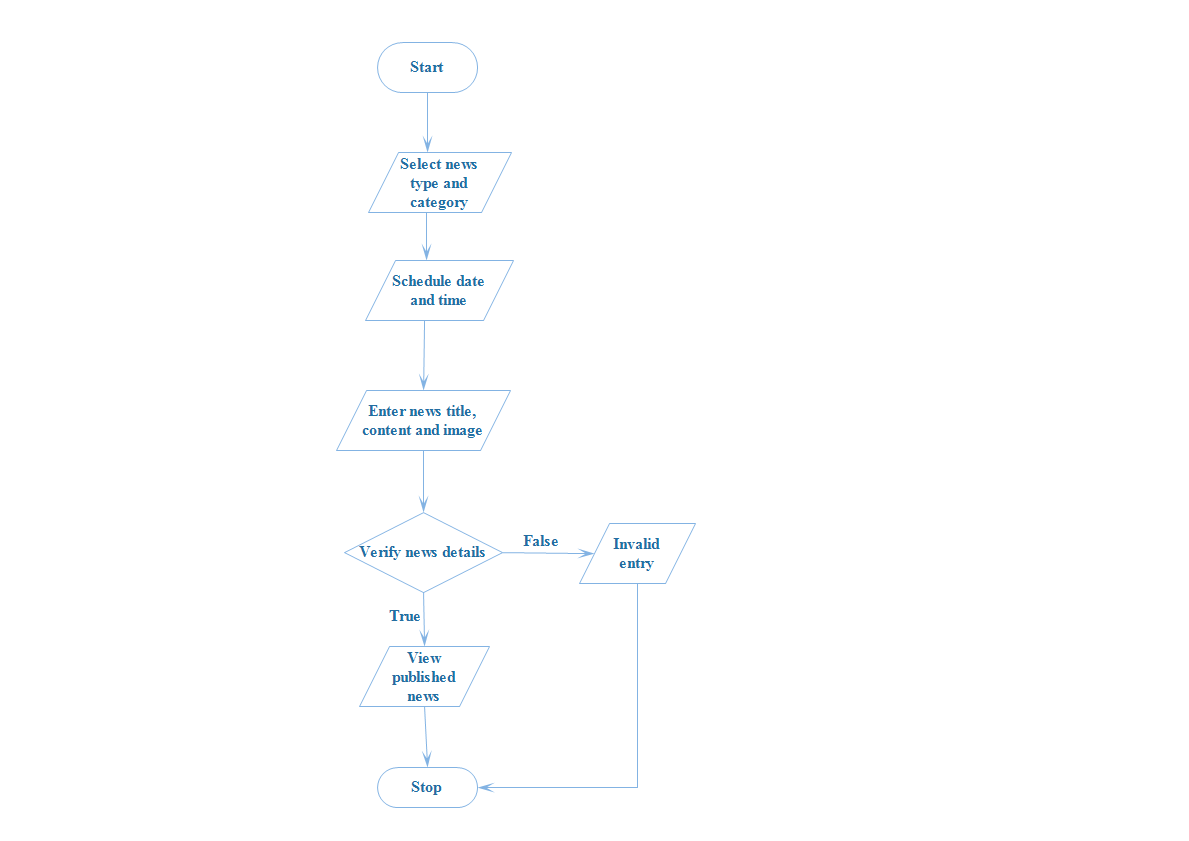


**Fig 5.7 Flowchart for Discussion**

* **File I/O Interface :** Graphical User Interface
* **Output :** If the entered Discussion details are valid then the data will be stored in the respective database and will be displayed in the Dashboard/Account. Else if the enterd data is not valid, it will display Invalid entry.

**5.4.6 News Details :**

* **Input :** News Type, News Title, News Category, Publish Date, Publish Time, Content, Image and Status.
* **Procedural Details (Flowchart) :**

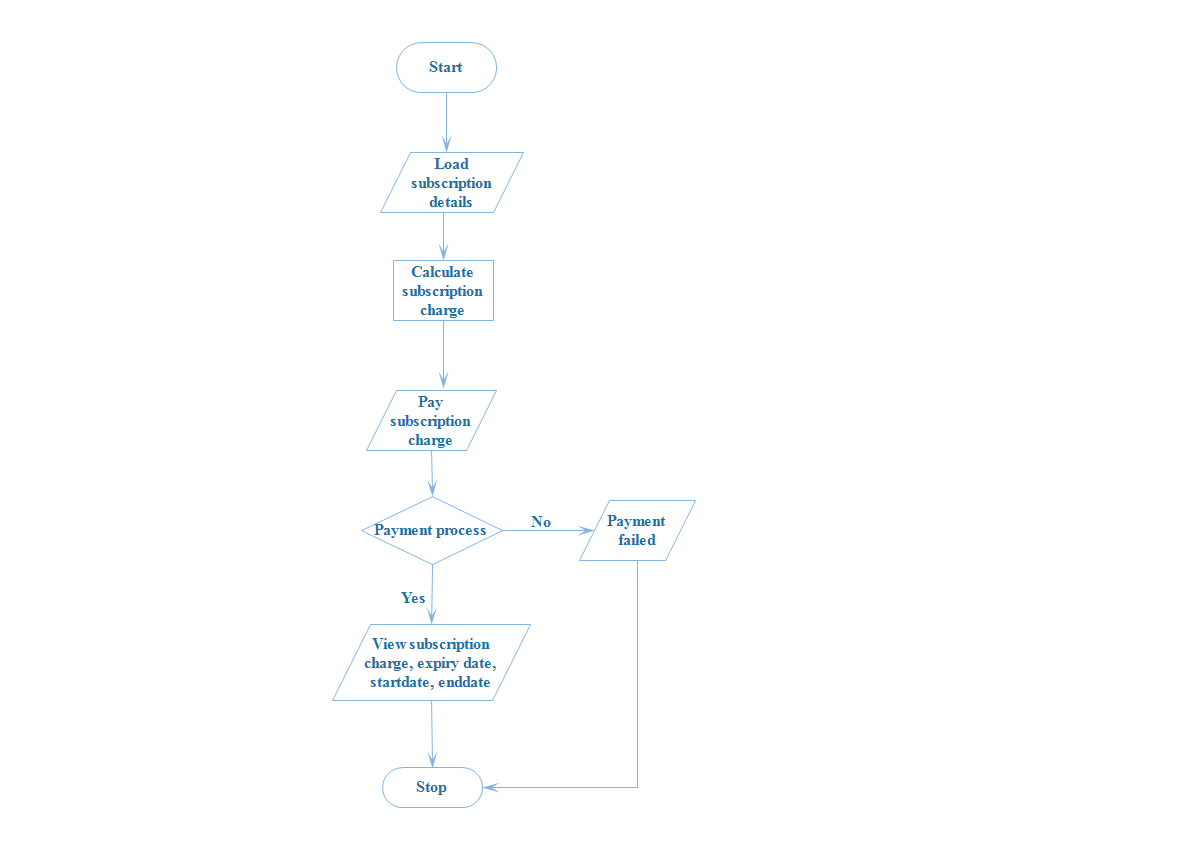


**Fig 5.8 Flowchart for News**

* **File I/O Interface :** Graphical User Interface
* **Output :** If the entered News details are valid then the data will be stored in the respective database and will be displayed in the Dashboard/Account. Else if the enterd data is not valid, it will display Invalid entry.

**5.4.7 SubscriptionDetails :**

* **Input :**Card Number, CVV Number and Expiry Date.
* **Procedural Details (Flowchart) :**

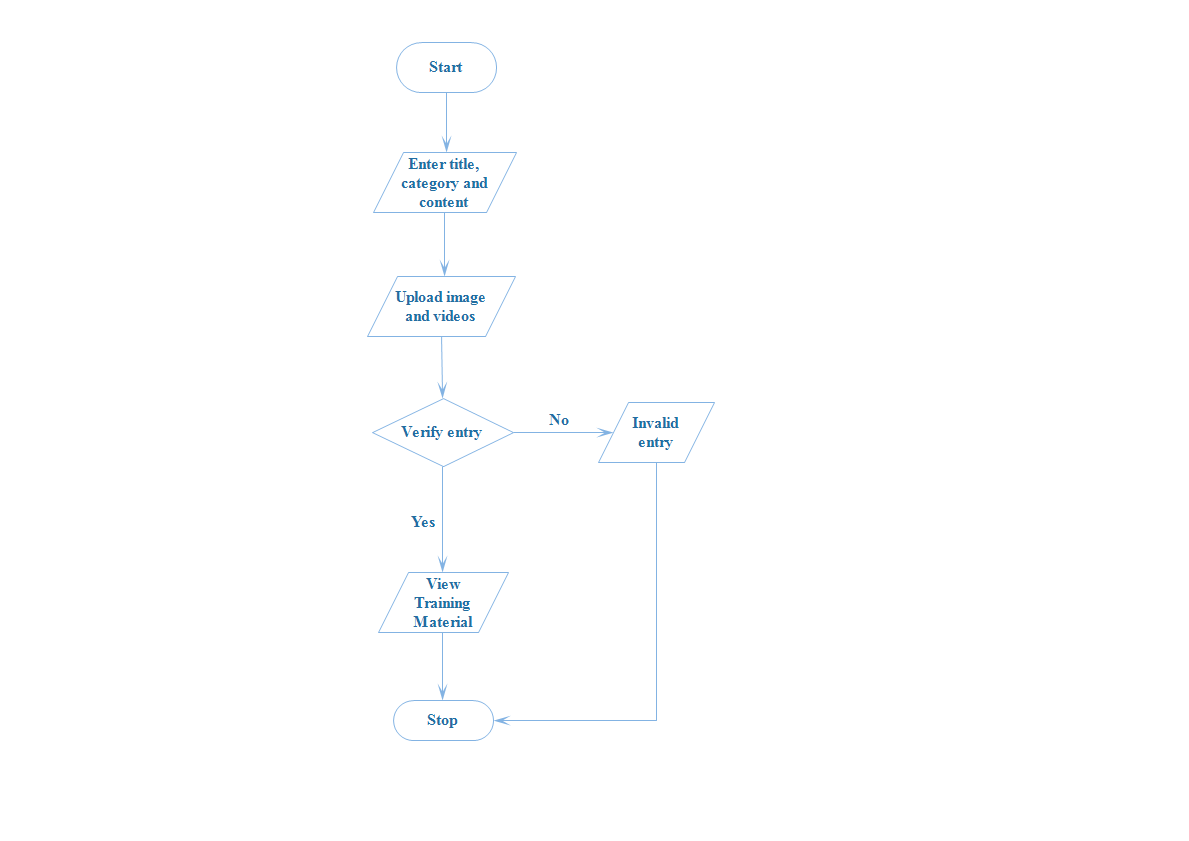


**Fig 5.9 Flowchart for Subscripton**

* **File I/O Interface :** Graphical User Interface
* **Output :** If the entered Subscription details are valid then the data will be stored in the respective database and will be displayed in the Dashboard/Account. Else if the enterd data is not valid, it will display Payment failed.

**5.4.8 Training material flowchart :**

* **Input :** Title, Training Category, Content, Image Link, Video Link and Status.
* **Procedural Details (Flowchart) :**



**Fig 5.10 Flowchart for Training Material**

* **File I/O Interface :** Graphical User Interface
* **Output :** If the entered Training material details are valid then the data will be stored in the respective database and will be displayed in the Dashboard/Account. Else if the enterd data is not valid, it will display Invalid

**Chapter 6**

**Coding**

**6.1 Database Connection (dbconnection.php) :**

<?php

$con = mysqli\_connect("localhost","root","gube","employee\_association");

if(mysqli\_connect\_error())

{

echo "Failed to connect to MySQL:".mysqli\_connect\_error();

}

?>

**6.2 Authorization/ Authentication (index.php) :**

<?php

session\_start();

include("dbconnection.php");

if(isset($\_SESSION[empid]))

{

**//Redirect to dashboard page if the employee has already logged in**

header("Location: dashboard.php");

}

if(isset($\_POST[btnsignin]))

{

$sql= "SELECT \* FROM employee WHERE login\_id='$\_POST[login\_id]' AND password='$\_POST[password]' AND status= 'Active' " ;

if(!$qrs=mysqli\_query($con,$sql))

{

echomysqli\_error($con);

}

if(mysqli\_num\_rows($qrs)==1)

{

$rs=mysqli\_fetch\_array($qrs);

$\_SESSION[empid]=$rs[employee\_id];

$\_SESSION[designation\_id] = $rs[designation\_id];

header("Location: dashboard.php");

}

else

{

**//Alert messegefor Invalid Data Entry**

echo "<script>alert(Invalid Login..};</script>";

}

}

include("header1.php");

include("slider.php");

?>

<div class="template-page-wrapper">

<form class="form-horizontal templatemo-signin-form" role="form" action="index.php" method="post" name="frmindex" onsubmit="return validateform()">

<div class="form-group">

<div class="col-md-12">

<label for="username" class="col-sm-2 control-label">Username<font color="red">\*</font></label>

<div class="col-sm-10">

<input type="text" class="form-control" id="username" placeholder="Username" name="login\_id">

</div>

</div>

</div>

<div class="form-group">

<div class="col-md-12">

<label for="password" class="col-sm-2 control-label">Password<font color="red">\*</font></label>

<div class="col-sm-10">

<input type="password" class="form-control" id="password" placeholder="Password" name="password">

</div>

</div>

</div>

<div class="form-group">

<div class="col-md-12">

<div class="col-sm-offset-2 col-sm-10">

<div class="checkbox">

<label>

<input type="checkbox"> Remember me

</label>

</div>

</div>

</div>

</div>

<div class="form-group">

<div class="col-md-12">

<div class="col-sm-offset-2 col-sm-10">

<input type="submit" value="Sign in" class="btnbtn-default" name="btnsignin">

</div>

</div>

</div>

</form>

<center><a href="forgot\_password.php"><strong> Forgot Password>></strong></a></center>

</div>

</div>

</body>

</html>

<script type="application/javascript">

functionvalidateform()

{

if(document.frmindex.username.value==""&&document.frmindex.password.value= ="")

{

alert("Failed Sign in..");

document.frmindex.username.focus();

return false;

}

if(document.frmindex.username.value=="")

{

alert("User name should not be empty..");

document.frmindex.username.focus();

return false;

}

else if(document.frmindex.password.value=="")

{

alert("Kindly enter the password..");

document.frmindex.password.focus();

return false;

}

else

{

return true;

}

}

</script>

**6.3branch.php :**

<?php

session\_start();

include("header.php");

include("leftsidebar.php");

include("dbconnection.php");

if(isset($\_POST[btnsubmit]))

{

**//Code for updating the existing employee record**

if(isset($\_GET[editid]))

{

$sql="UPDATE branch SET branchname=’$\_POST[branchname]’,

description='$\_POST[notes]',status='$\_POST[status]' WHERE

branch\_id='$\_GET[editid]'";

mysqli\_query($con,$sql);

**//After Successful update, page redirects to view employee Page**

echo "<script>alert('Branch record updated successfully...');</script>"; }

else

**//Code for Inserting New employee Record**

{

$sql = "INSERT INTO branch(branchname,description,status)

values('$\_POST[branchname]','$\_POST[notes]','$\_POST[status]')";

mysqli\_query($con,$sql);

echo "<script>alert('Branch record inserted successfully...');</script>";

}

}

**//Code for retrievingemployee Record**

if(isset($\_GET[editid]))

{

$sql="SELECT \* FROM branch where branch\_id='$\_GET[editid]'";

$qsql=mysqli\_query($con,$sql);

$rsrec = mysqli\_fetch\_array($qsql);

}

?>

<div class="templatemo-content-wrapper">

<div class="templatemo-content">

<h1><b>Add Branch</b></h1>

<p class="margin-bottom-15">Kindly enter the following details to add branch</p>

<div class="row">

<div class="col-md-12">

<form role="form" name="frmbranch" method="post" onsubmit="return

validateform()">

<div class="row">

<div class="col-md-6 margin-bottom-15">

<label for="branchname" class="control-label">Branch Name <font

color="red">\*</font></label>

<input type="text" class="form-control" name="branchname" value="<?php echo

$rsrec[branchname]; ?>" >

</div>

</div>

<div class="row"></div>

<div class="row"></div>

<div class="row"></div>

<div class="has-success has-feedback">

<div class="row"></div>

</div>

<div class="row">

<div class="col-md-12 margin-bottom-15">

<label for="notes">Notes</label>

<textarea class="form-control" rows="3" name="notes"><?php echo $rsrec[description];

?></textarea>

</div>

</div>

<div class="row">

<div class="col-md-6 margin-bottom-15">

<label for="status">Status <font color="red">\*</font></label>

<select class="form-control margin-bottom-15" name="status">

<?php

$arr = array("Select","Active","Inactive");

foreach($arr as $val)

{

if($val == $rsrec["status"])

{

echo "<option value='$val' selected>$val</option>";

}

else

{

echo "<option value='$val'>$val</option>";

}

}

?>

</select>

</div>

</div>

<div class="row"></div>

<div class="row"></div>

<div class="row"></div>

<div class="row templatemo-form-buttons">

<div class="col-md-12">

<input type="submit" name="btnsubmit" class="btnbtn-primary">

<button type="reset" class="btnbtn-default">Reset</button>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

<?php

include("footer.php");

?>

<script type="application/javascript">

**//Validation for Required fields**

functionvalidateform()

{

if(document.frmbranch.branchname.value=="" &&document.frmbranch.status.value

=="Select")

{

alert("Enter the branch details..");

document.frmbranch.branchname.focus();

return false;

}

if(document.frmbranch.branchname.value=="")

{

alert("Branch name should not be empty..");

document.frmbranch.branchname.focus();

return false;

}

else if(document.frmbranch.status.value=="Select")

{

alert("Kindly select the status..");

document.frmbranch.status.focus();

return false;

}

else

{

return true;

}

}

</script>

**6.4 view\_branch.php :**

<?php

include("header.php");

include("leftsidebar.php");

include("dbconnection.php");

**//Code for deletingemployee Record**

if(isset($\_GET[delid]))

{

$sql ="DELETE FROM branch where branch\_id='$\_GET[delid]'";

$qsql = mysqli\_query($con,$sql);

if(mysqli\_affected\_rows($con) ==1 )

{

echo "<script>alert('Branch record deleted successfully...');</script>";

}

}

?>

<div class="templatemo-content-wrapper">

<div class="templatemo-content">

<h1><b>View Branch</b></h1>

<div class="row">

<div class="col-md-12">

<div class="table-responsive">

<h4 class="margin-bottom-15">View branch details</h4>

<div class="table-responsive">

<style type="text/css">

/\*code for auto number \*/

table{counter-reset:section;}

.count:before

{

counter-increment:section;

content:counter(section);

}

</style>

<table width="735" class="table table-striped table-hover table-bordered">

<thead>

<tr>

<th width="24">#</th>

<th width="195"><center>Branch Name</center></th>

<th width="265"><center>Description</center></th>

<th width="65"><center>Edit</center></th>

<th width="80"><center>Status</center></th>

<th width="78"><center>Delete</center></th>

</tr>

</thead>

<tbody>

<?php

$sql = "SELECT \* FROM branch WHERE status!='Deleted'";

$rsquery = mysqli\_query($con,$sql);

while($rs = mysqli\_fetch\_array($rsquery))

{

?>

<tr>

<td class= "count"></td>

<td><?php echo $rs[branchname]; ?></td>

<td><?php echo $rs[description]; ?></td>

<td><a href="branch.php?editid=<?php echo $rs[branch\_id]; ?>" class="btnbtn-

default">Edit</a></td>

<td>

<?php

echo $rs[status];

?>

</td>

<td><a href="view\_branch.php?delid=<?php echo $rs[branch\_id]; ?> " class="btnbtn-

default">Delete</a></td>

</tr>

<?php

}

?>

</tbody>

</table>

</div>

</div>

<ul class="pagination pull-right">

<li class="disabled"><a href="#">&laquo;</a></li>

<li class="active"><a href="#">1 <span class="sr-only">(current)</span></a></li>

<li><a href="#">2 <span class="sr-only">(current)</span></a></li>

<li><a href="#">3 <span class="sr-only">(current)</span></a></li>

<li><a href="#">4 <span class="sr-only">(current)</span></a></li>

<li><a href="#">5 <span class="sr-only">(current)</span></a></li>

<li><a href="#">&raquo;</a></li>

</ul>

</div>

</div>

</div>

</div>

<?php

include("footer.php");

?>

**Chapter 7**

**User Interface**

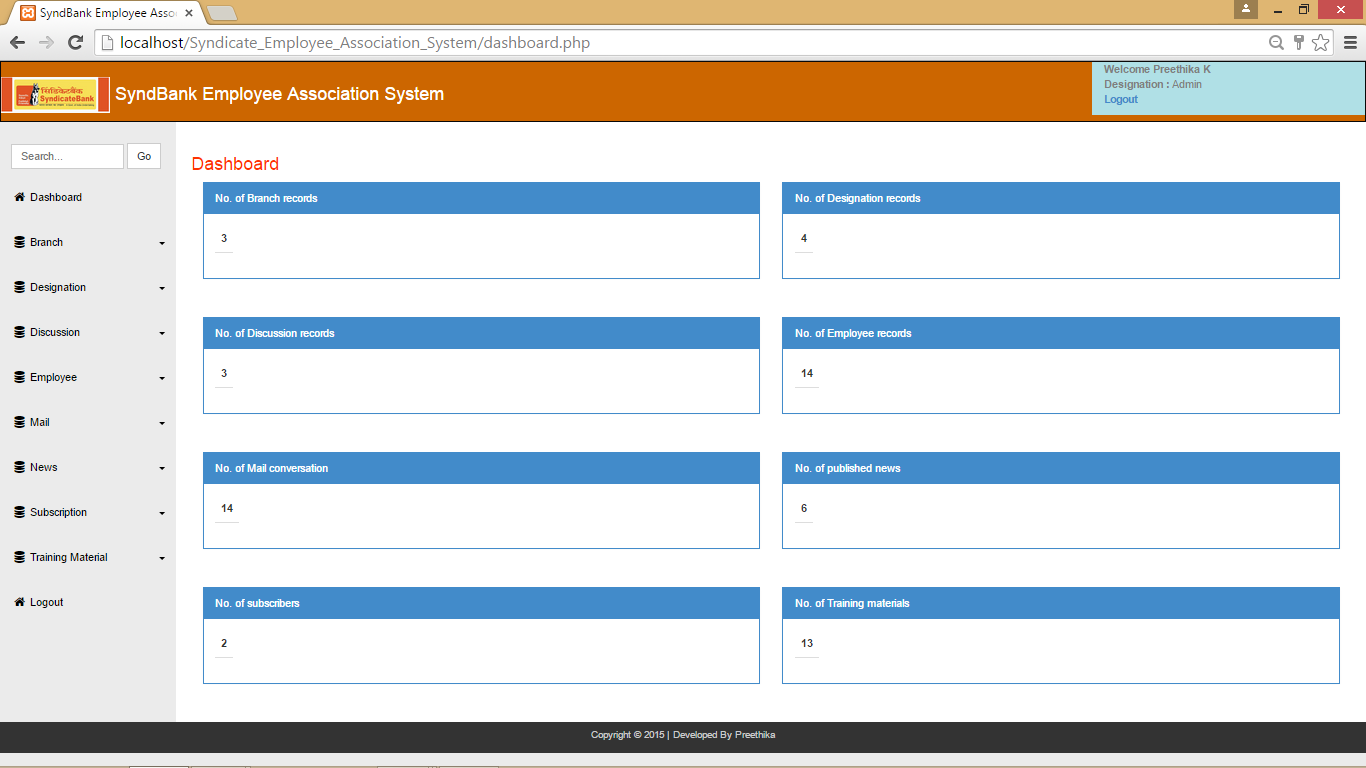
* **Login Page:**



**Fig 7.1 Login**

This is the start up screen that will appear every time when admin and employee log on the system. The user need to enter the valid Username and Password, if both are valid then the login is success and the user is allowed to enter into the SyndBank Employee Association

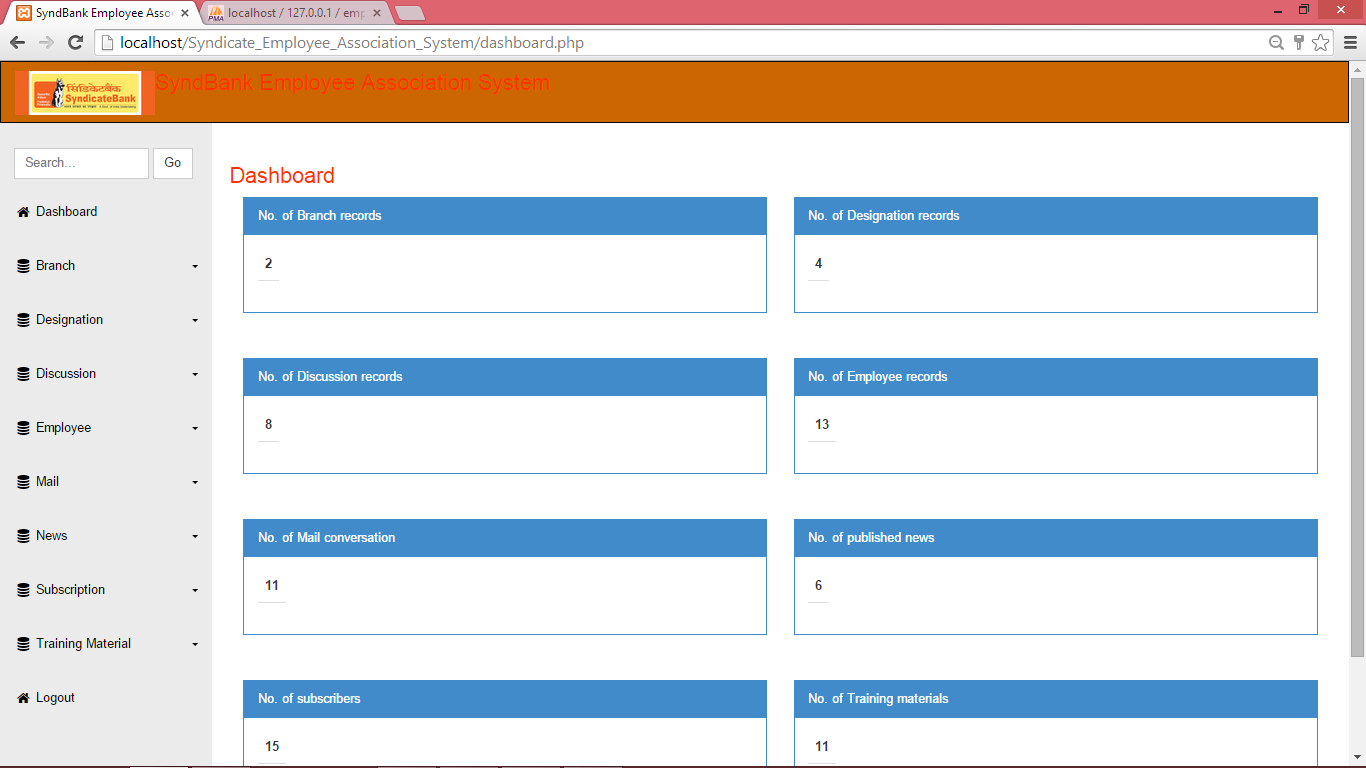
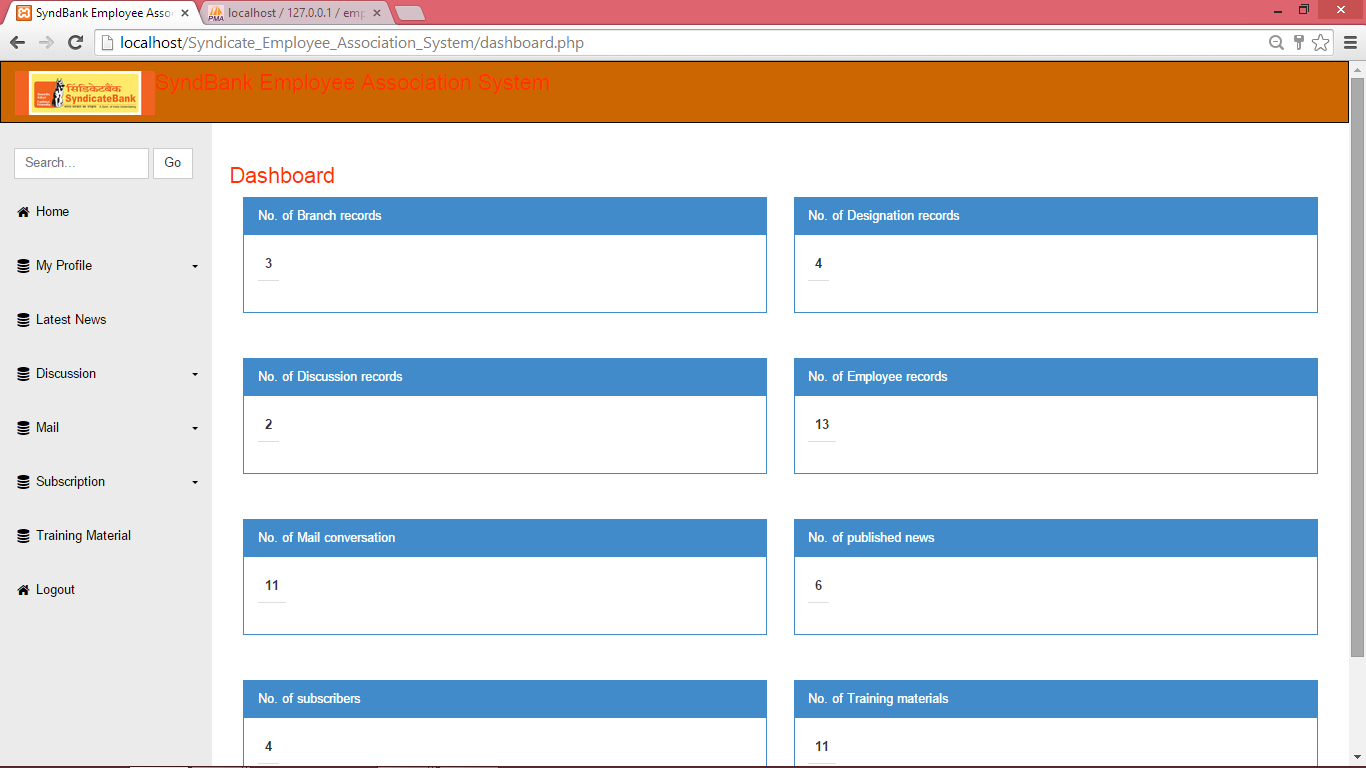
* **Home Screen:**



**Fig 7.2 Home Screen**

Home Screen is named as Admin Dashboard when the Admin logs on the system otherwise the Home Screen is named as Employee account when the Employee logs on the system. Home Screen displays the quick information like No. of Branches, No. of Designation, No. of Discussions, No. of Employees etc.

* **Menus for Admin and Employee:**

**Fig 7.3 Menu for Admin Fig 7.4 Menu for Employee**

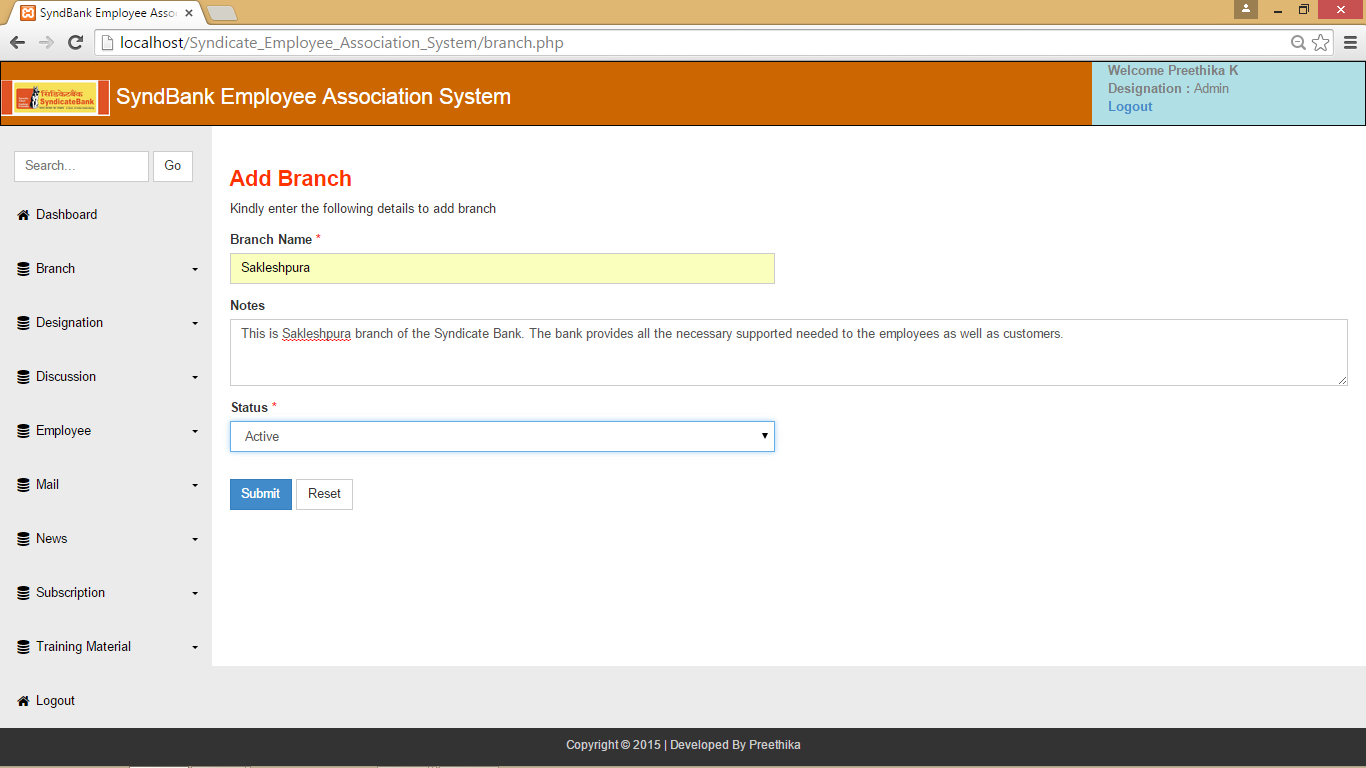
There are the following menus displayed in the left side of each page in the Project:

**(For Admin):**

* Dashboard
* Branch
* Designation
* Discussion
* Employee
* Mail
* News
* Subscription
* Training Material
* Logout

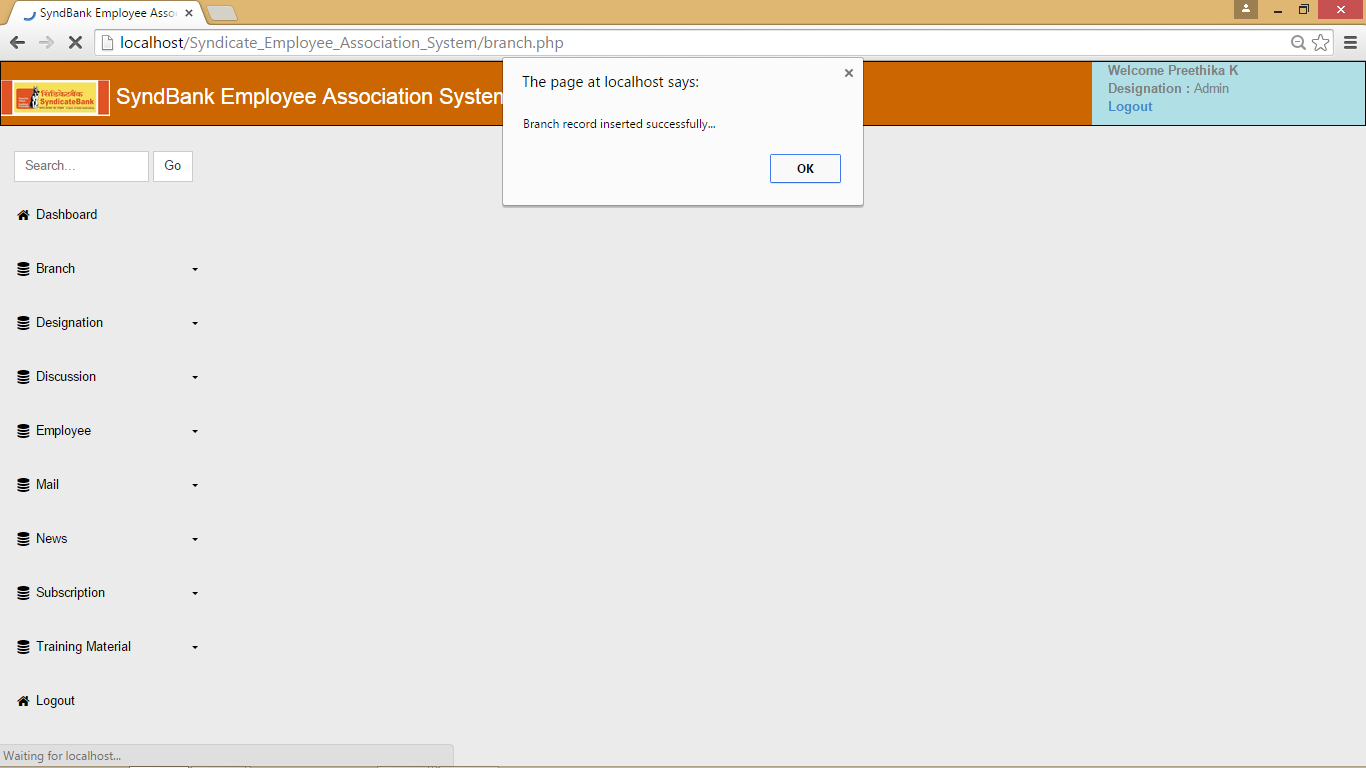
**(For Employee):**

* Home
* My Profile
* Latest News
* Discussion
* Mail
* Subscription
* Training Material
* Logout
* **Branch Record:**



**Fig 7.5 Add Branch**

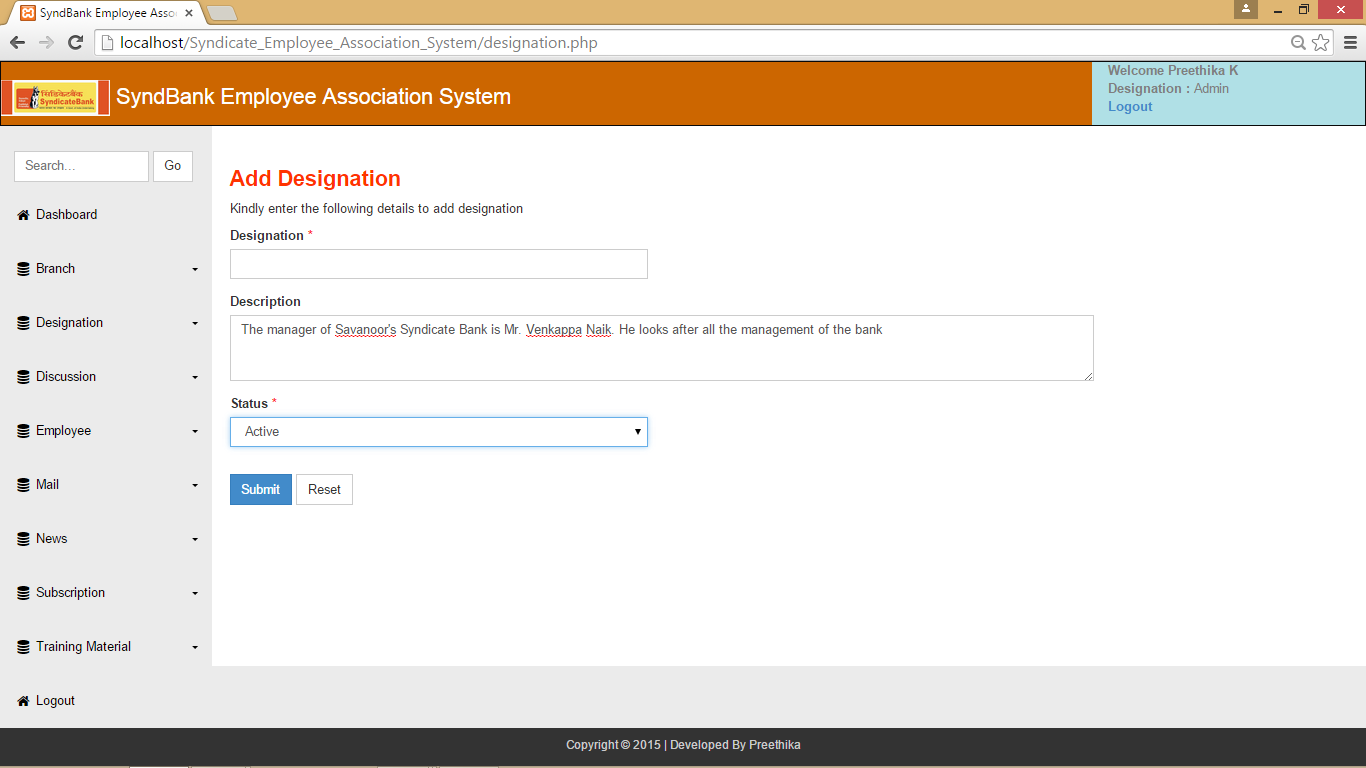
Here the Admin can add branch details like Branch Name, Notes and Status. Similarly he have the power to add designation details, discussion details, training details, news details, subscription details, employee details and mail details.



**Fig 7.6 Branch record inserted**

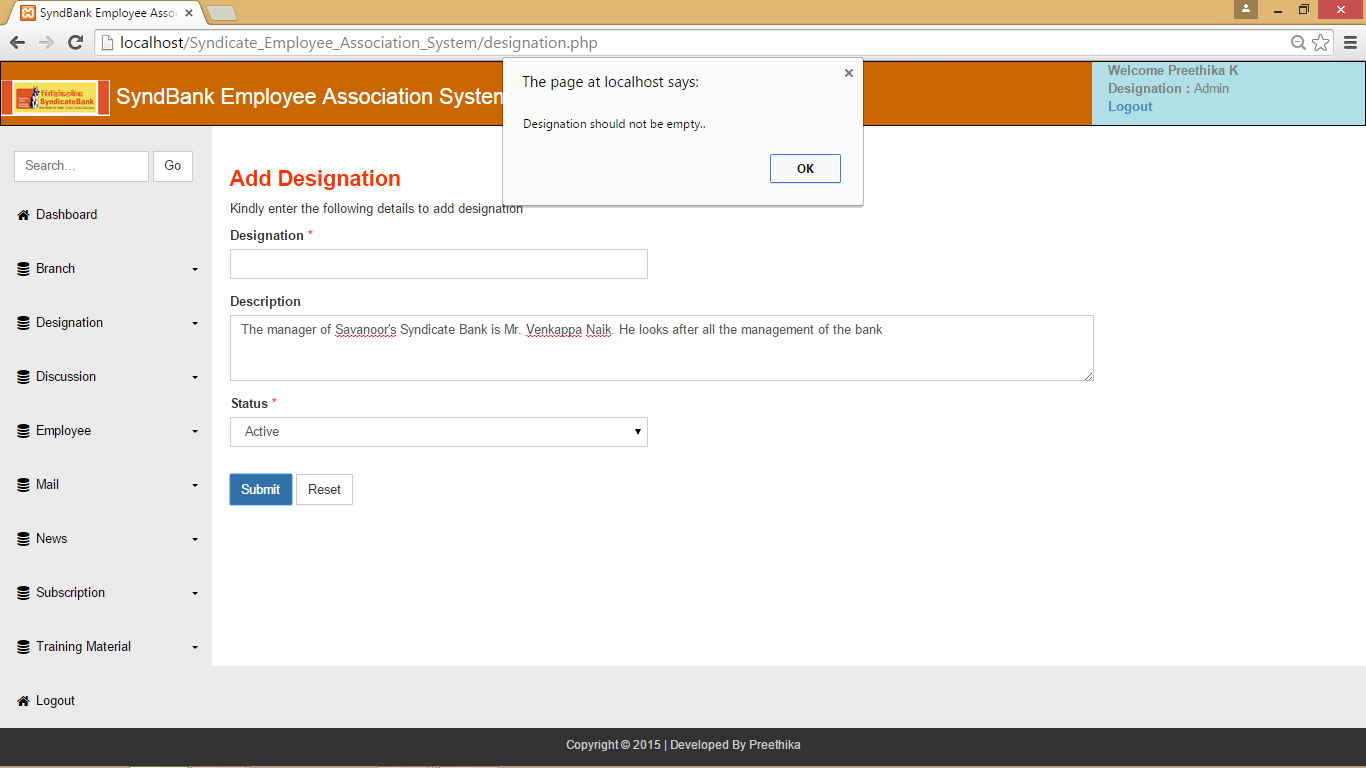
When the Admin clicks on Submit Button after adding branch record, the record is inserted successfully in the database and a message is displayed on the screen. Similarly the Admin can delete and update the record.

* **Validation (Designation) :**



**Fig 7.7 Add Designation**

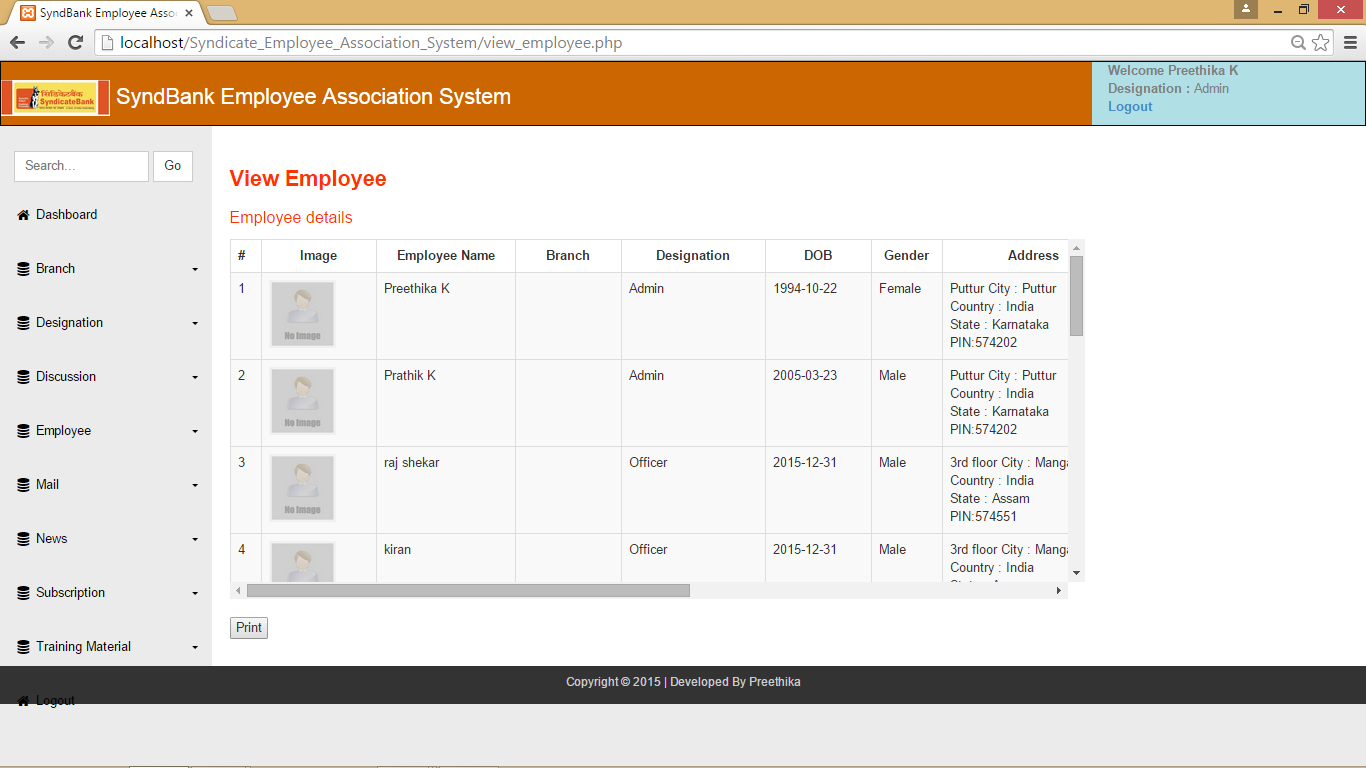
Here the Admin inserts a record with Designation, Description and Status without entering the Designation field.



**Fig 7.8 Validation for Designation field**

Here the user gets the validation for not entering the mandatory field “Designation” with a message displaying Designation should not be empty. Similarly all the other mandatory fields are validated.

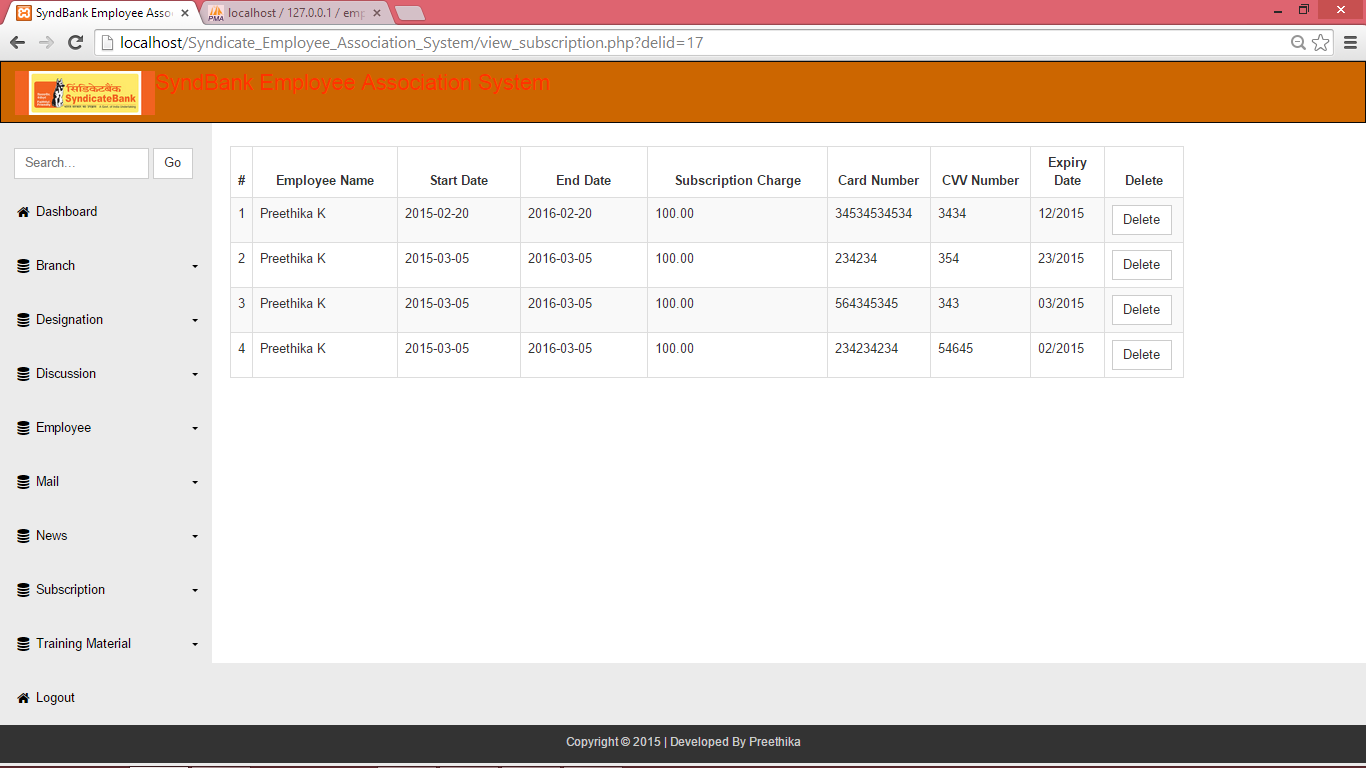
* **View Employee:**



**Fig 7.9 View Employee details**

Here the admin can view all the employee details like Employee Name, Branch , Designation, Login ID, Password, Image, Date of Birth, Date of Join, Date of Retirement, Email ID, Contact No., Address, Gender, Country, City, State, Pin Code etc.

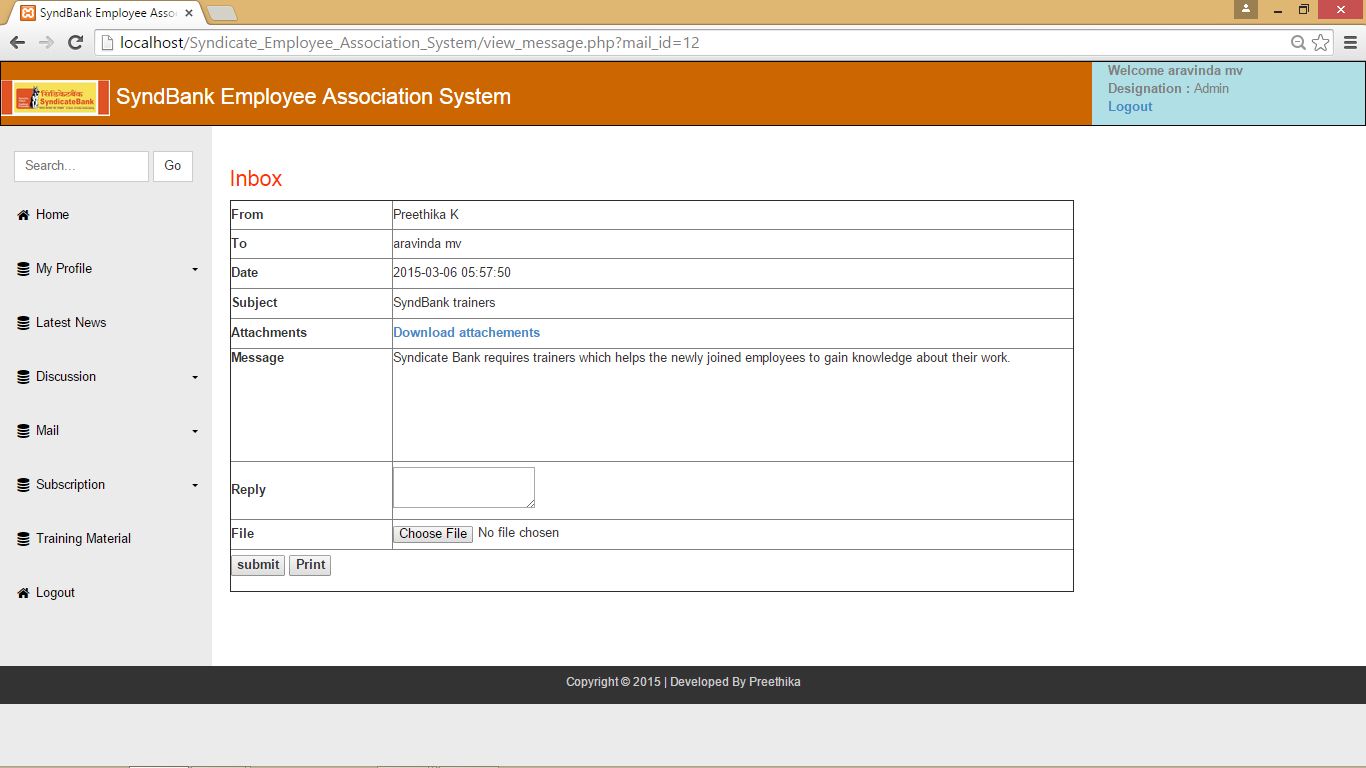
* **On Screen Reports (Subscription) :**



**Fig 7.10 On Screen Report**

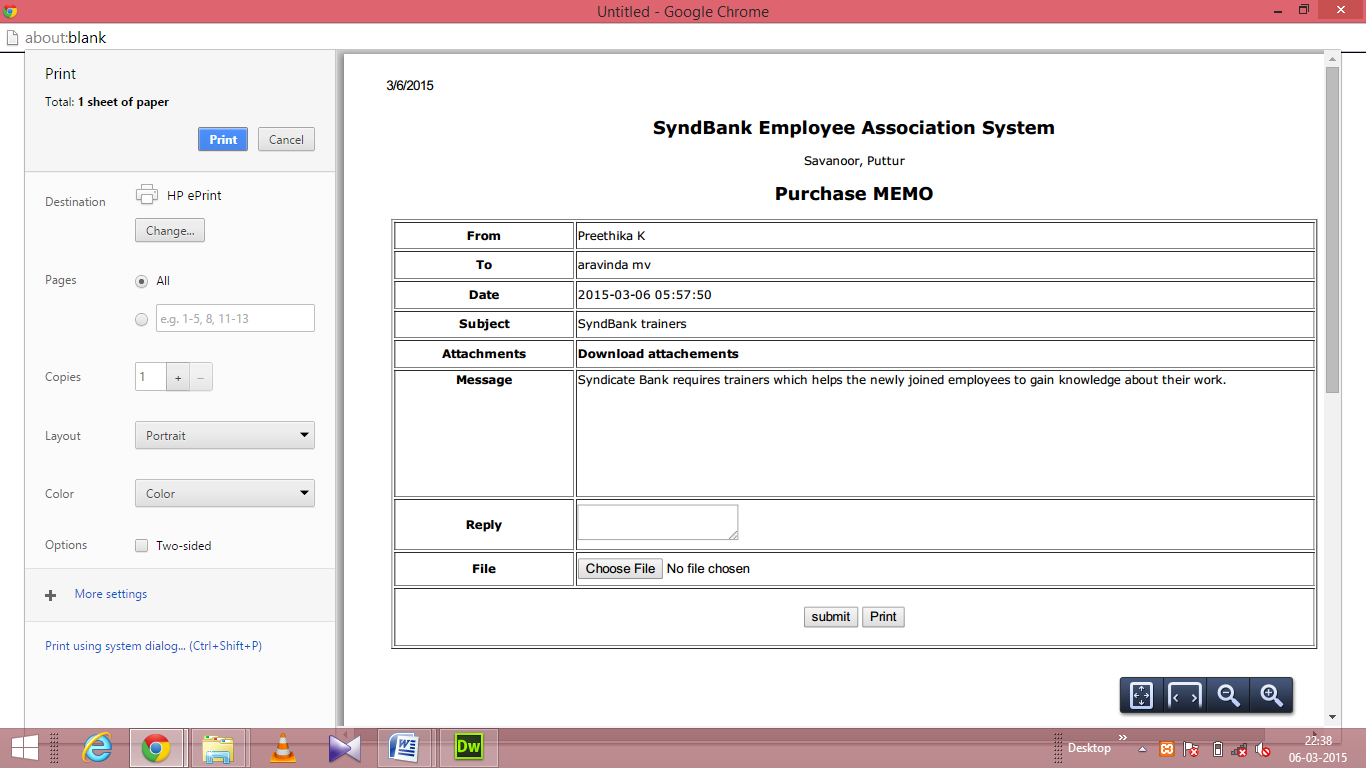
Here the user can view the Subscription Report which displays Employee Name, Start Date, End Date, Subscription Charge, Card Number, CVV Number and Expiry Date

* **Data Reports (mail) :**



**Fig 7.11 Inbox mail**

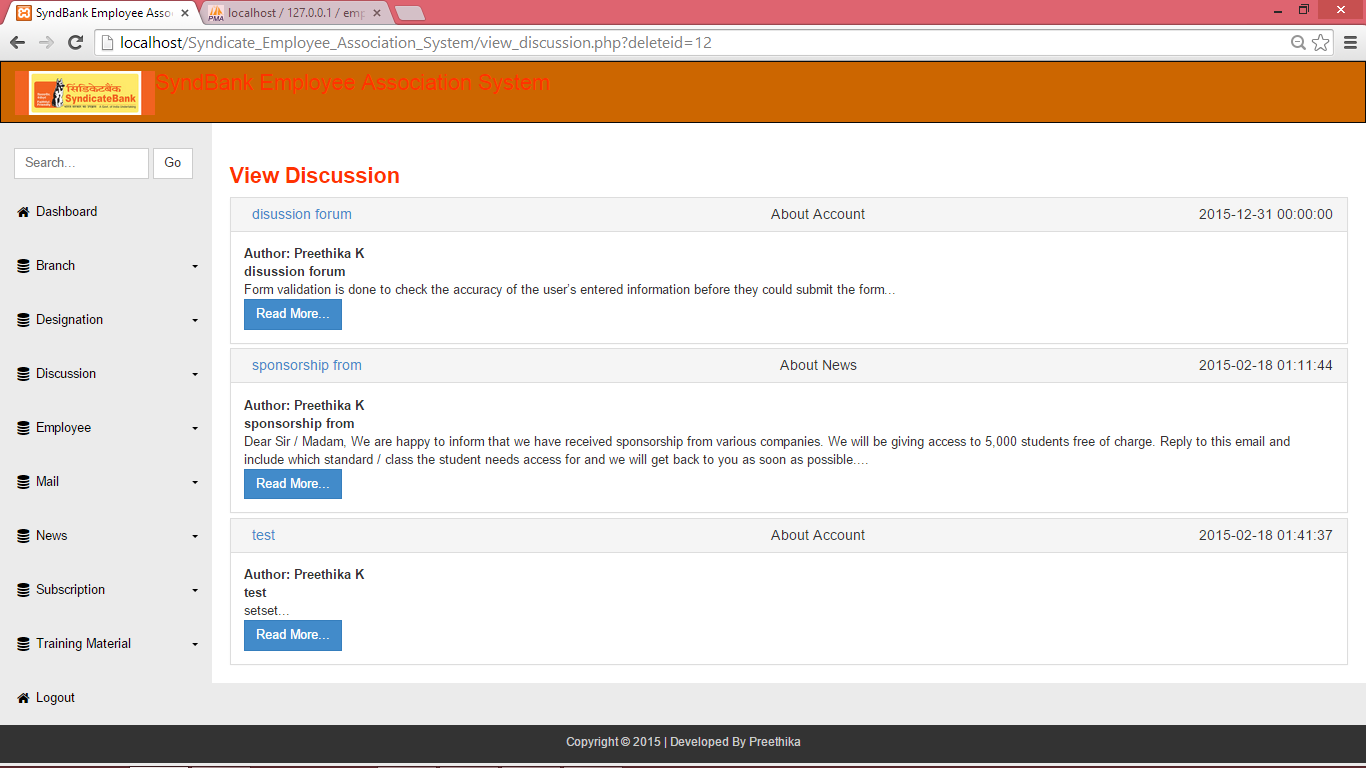
This is the screen view of mail; where we can have the detailed view of the inbox sent by other employee. If we want a printout of the mail, we can do so by just clicking on the Print button on the screen.



**Fig 7.12 Data Report**

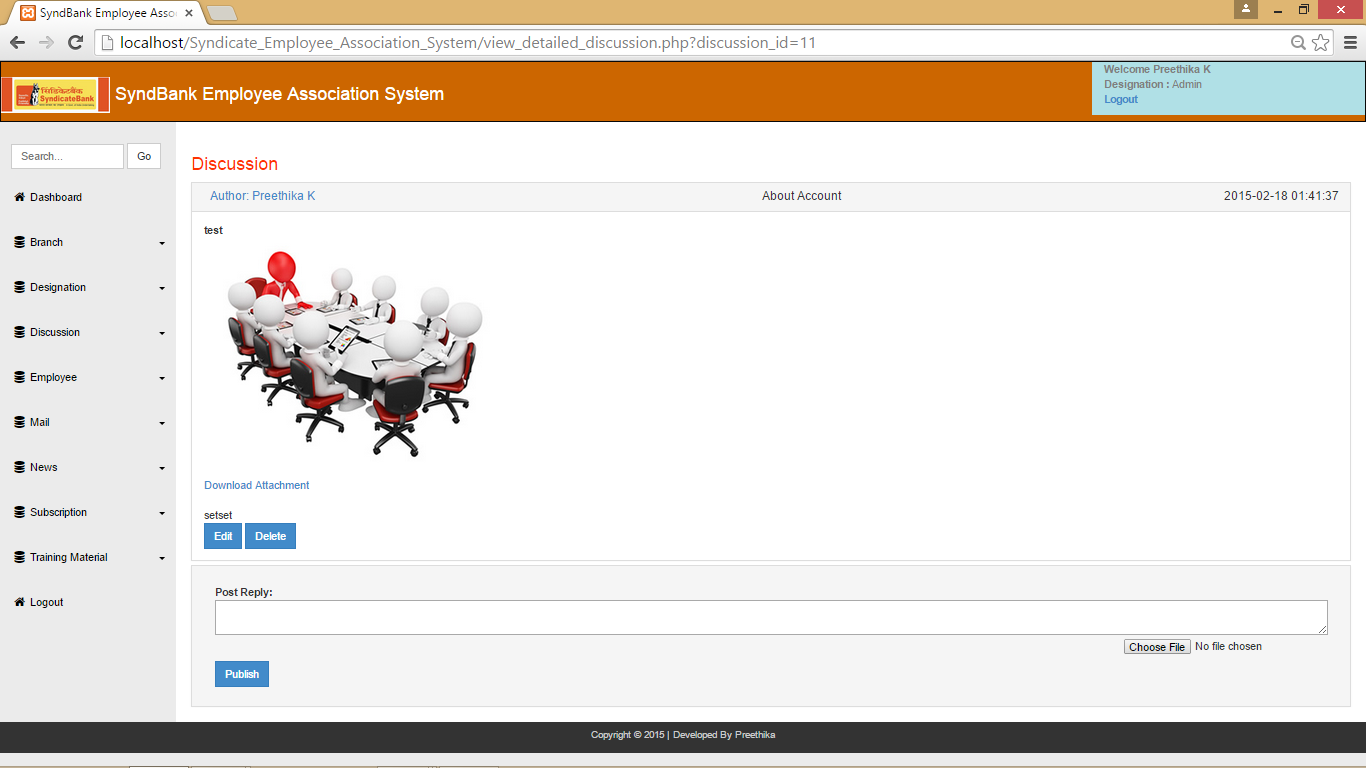
After clicking on Print button, a new page displaying the page with print options and features appears on the screen. Through the help of this we can take the printout of the mail.

* **Alerts ( View Discussion):**



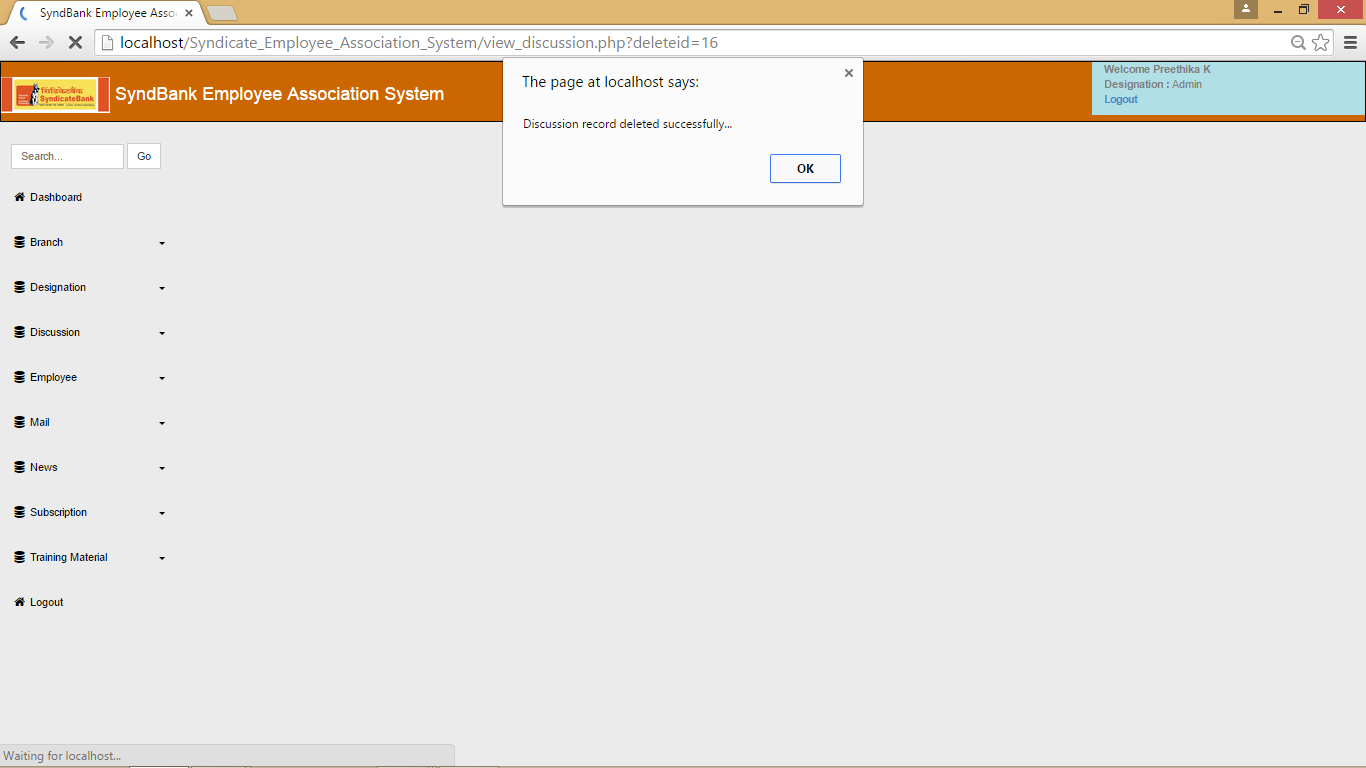
**Fig 7.13 View Discussion forum**

This is the screenview of discussion. Here the admin can view the discussion forum details which consist of several discussions. Each discussion contains category, Discussion title, published date and time, author name and description about the particular discussion. If the user wants to read the discussion in detail, then Read more option is being clicked.



**Fig 7.14 View Discussion in detail**

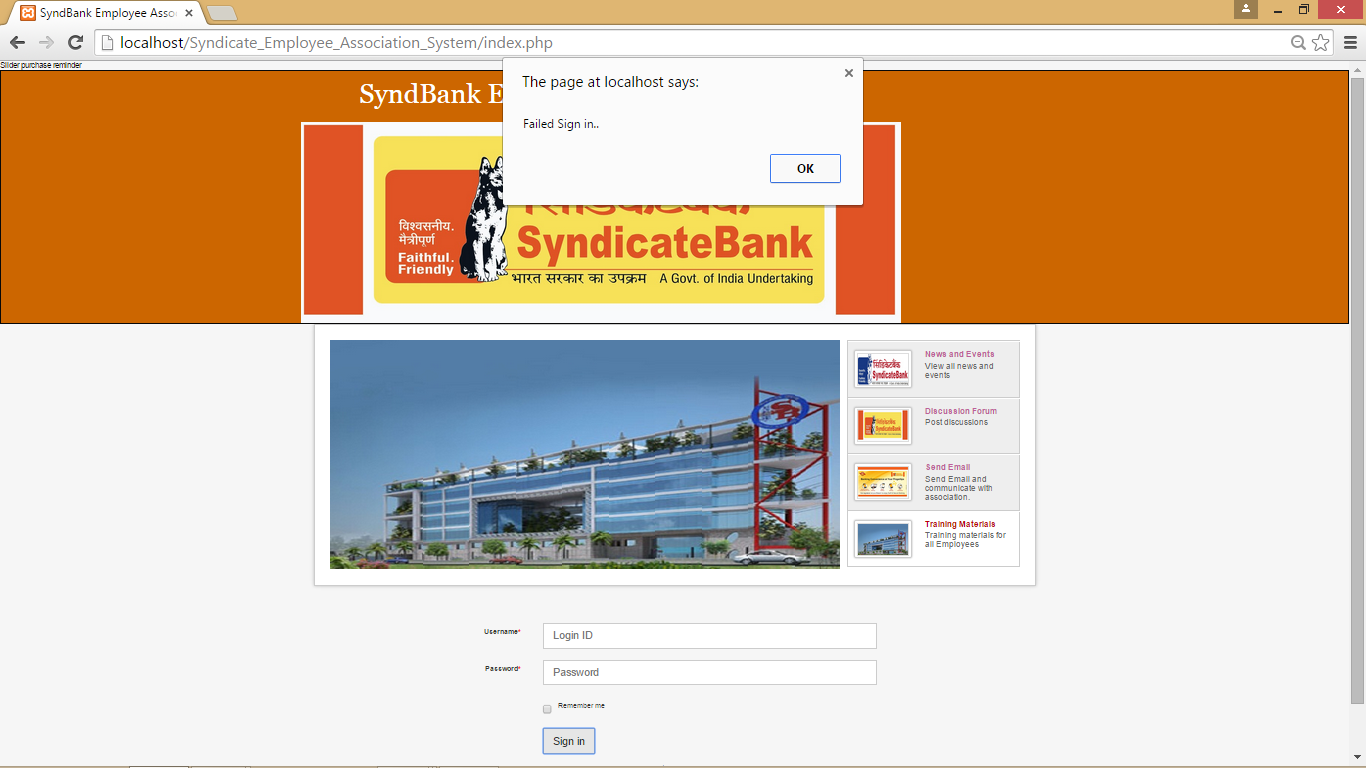
Here we can view a particular discussion with the category, Discussion title, published date and time, author name and full description about the discussion with an image. Even we can also download an attachment if provided.



**Fig 7.15 Alerts for deleting the Designation record**

While viewing the discussion forum, if the admin wants to delete any particular discussion then he can delete it by pressing Delete button the screen and an alert message will be displayed on the screen stating “Discussion record deleted successfully..”

* **Error message:**



**Fig 7.16 Error Message for Login page**

Some errors occur in case of errors in communication. When the user clicks on Sign in without entering the Login details like User Name and Password, he gets an error message displaying Failed Sign in..

**Chapter 8**

**Testing**

* 1. **Introduction :**

Testing is the process of running a system with the intention of finding errors. Testing enhance the integrity of the system by detecting the deviations in designs and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in the system. Testing also adds values to the product by confirming to the user requirement.

The main purpose is to detect errors and error-prone areas in the system. Testing must be done in well planned manner. A partially tested system is as bad as an untested system. And also the price of an untested and under-tested system is high.

The implementation is the final and important phase. It involves user training, system testing in order to ensure successful running of the proposed system. The user tests the system and changes are made according to their needs. The testing involves the testing of the developed system using various kinds of data. While testing, errors are noted and correctness is made.

* 1. **Objectives of Testing :**

The objectives of the testing are:

* Testing in a process of executing a program with the intent of finding errors.
* A successful test case is one that uncovers an as yet undiscovered error.

System testing is a stage of implementation which is aimed at ensuring that the system works correctly and efficiently as per the user need, before the live operation commences. As stated before, testing is vital to the success of a system. System testing makes logical assumption that if all the system is correct, the goal will be successfully achieved. A series of tests are performed before the system is ready of user acceptance test.

* 1. **Test Reports :**
* **Unit Testing:**

In this testing each sub module is tested individually with test data and then integrated to the overall system. Unit testing focuses verification effort on the smallest unit of the software design in the module or component. In fund flow analysis all modules of the system are tested separately. The testing was carried out during the coding stage itself. Each module was tested and it was being developed. All the modules were correct for error and finally worked satisfactorily.

**Unit Testing Tables :**

* **Sign in :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Test Condition** | **Expected Result** | **Result** |
| 1 | Click on Sign in button without entering Username and Password | Failed Sign in.. | Successful |
| 2 | Click on Sign in button without entering Username | Username should not be empty.. | Successful |
| 3 | Click on Sign in button without entering Password | Kindly enter the password.. | Successful |

**Table 8.1 login testing**

* **Forgot password :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Test Condition** | **Expected Result** | **Result** |
| 1 | Click on Recover password button without entering Login ID and Email ID | Enter Login ID and Email ID.. | Successful |
| 2 | Click on Recover password button without entering Login ID | Login ID should not be empty.. | Successful |
| 3 | Click on Recover password button without entering Email ID | Kindly enter the Email ID.. | Successful |

**Table 8.2 Forget Password testing**

* **Branch :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Test Condition** | **Expected Result** | **Result** |
| 1 | Click on Submit button without entering Branch Name, Notes and Status | Enter the branch details.. | Successful |
| 2 | Click on Submit button without entering Branch Name | Branch Name should not be empty.. | Successful |
| 3 | Click on Submit button without selecting Status | Kindly select the Status.. | Successful |

**Table 8.3 Branch Testing**

* **Designation :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Test Condition** | **Expected Result** | **Result** |
| 1 | Click on Submit button without entering Designation, Description and Status | Enter the designation details.. | Successful |
| 2 | Click on Submit button without entering Designation | Designation should not be empty.. | Successful |
| 3 | Click on Submit button without selectingStatus | Kindly select the Status.. | Successful |

**Table 8.4 Designation Testing**

* **Discussion :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Test Condition** | **Expected Result** | **Result** |
| 1 | Click on Submit button without entering Category, Discussion Title, Description, Discussion Image, Uploads and Status | Enter the discussion details.. | Successful |
| 2 | Click on Submit button without entering Category | Kindly select the Category.. | Successful |
| 3 | Click on Submit button without entering Discussion Title | Discussion Title should not be empty.. | Successful |
| 4 | Click on Submit without addingDiscussion Image | Kindly add an image.. | Successful |
| 5 | Click on Submit button without adding Uploads | Uploads should not be empty.. | Successful |
| 6 | Click on Submit button without selecting Status | Kindly select the Status.. | Successful |

**Table 8.5 Discussion Testing**

* **Employee :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Test Condition** | **Expected Result** | **Result** |
| 1 | Click on Submit button without entering First Name, Last name, Branch, Designation, Login Name, Password, Confirm New Password, Date of Birth, Gender, Address, Email Id, Contact No., Country, State, City, Pin Code, Date of Join, Date of Retirement, Image and Status | Enter the employee details.. | Successful |
| 2 | Click on Submit button without entering First Name | Kindly enter the First Name.. | Successful |
| 3 | Click on Submit button without entering Last Name | Kindly enter the Last Name.. | Successful |
| 4 | Click on Submit without selecting Branch | Kindly select the Branch.. | Successful |
| 5 | Click on Submit button without selecting Designation | Kindly select the Designation.. | Successful |
| 6 | Click on Submit button without entering Login Name | Login Name should not be empty.. | Successful |
| 7 | Click on Submit button without entering Password | Kindly enter a password for the Login name.. | Successful |
| 8 | Click on Submit button without entering Confirm New Password | Kindly confirm the entered password.. | Successful |
| 9 | Click on Submit button without entering Date of Birth | Date of Birth should not be empty.. | Successful |
| 10 | Click on Submit button without entering Gender | Kindly choose the Gender.. | Successful |
| 11 | Click on Submit button without entering Contact Number | Contact Number should not be empty.. | Successful |
| 12 | Click on Submit button without entering Email ID | Email ID should not be empty.. | Successful |
| 13 | Click on Submit button without entering Address | Address should not be empty.. | Successful |
| 14 | Click on Submit button without entering Country | Kindly select the country.. | Successful |
| 15 | Click on Submit button without entering State | Kindly select the State.. | Successful |
| 16 | Click on Submit button without entering City | City should not be empty.. | Successful |
| 17 | Click on Submit button without entering Pin Code | Pin Code should not be empty.. | Successful |
| 18 | Click on Submit button without entering Date of Join | Date of Join should not be empty.. | Successful |
| 19 | Click on Submit button without entering Date of Retirement | Date of Retirement should not be empty.. | Successful |
| 20 | Click on Submit button without entering Image | Kindly add an Image.. | Successful |
| 21 | Click on Submit button without selecting Status | Kindly select the Status.. | Successful |

**Table 8.6 Employee testing**

* **Mail :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Test Condition** | **Expected Result** | **Result** |
| 1 | Click on Submit button without entering To, Subject and  Message | Enter the mail details.. | Successful |
| 2 | Click on Submit button without selecting To | Kindly select the senderid.. | Successful |
| 3 | Click on Submit button without entering Subject | Subject should not be empty.. | Successful |
| 4 | Click on Submit button without entering Email ID | Kindly enter the message.. | Successful |

**Table 8.7 Mail Testing**

* **News :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Test Condition** | **Expected Result** | **Result** |
| 1 | Click on Submit button without entering News Type, News Category, Publish Date, Publish Time, News Title, Content, Image and Status | Enter the news details.. | Successful |
| 2 | Click on Submit button without selecting News Type | Kindly select the News Type.. | Successful |
| 3 | Click on Submit button without selecting News Category | Kindly select the News Category.. | Successful |
| 4 | Click on Submit button without entering Publish Date | Publish Date should not be empty.. | Successful |
| 5 | Click on Submit button without entering Publish Time | Publish Time should not be empty.. | Successful |
| 6 | Click on Submit button without entering News Title | News Title should not be empty.. | Successful |
| 7 | Click on Submit button without entering Content | Content should not be empty.. | Successful |
| 8 | Click on Submit button without adding an Image | Kindly upload an Image.. | Successful |
| 9 | Click on Submit button without selecting Status | Kindly select the Status.. | Successful |

**Table 8.8 News Testing**

* **Subscription :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Test Condition** | **Expected Result** | **Result** |
| 1 | Click on Submit button without entering Card Number, CVV Number and Expiry Date | Enter the subscription details.. | Successful |
| 2 | Click on Submit button without entering Card Number | Kindly enter the Card Number.. | Successful |
| 3 | Click on Submit button without entering CVV Number | Kindly enter the CVV Number.. | Successful |
| 4 | Click on Submit button without entering Expiry Date | Expiry month and expiry year should not be empty.. | Successful |

**Table 8.9 Subscription Testing**

* **Training Material :**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Test Condition** | **Expected Result** | **Result** |
| 1 | Click on Submit button without entering Title, Training Category, Content, Image Link, Video Link and Status | Enter the training details.. | Successful |
| 2 | Click on Submit button without entering Title | Title should not be empty.. | Successful |
| 3 | Click on Submit button without selecting Training Category | Kindly select the Training Category.. | Successful |
| 4 | Click on Submit button without entering Content | Content should not be empty.. | Successful |
| 5 | Click on Submit button without uploading Image Link | Kindly upload an Image.. | Successful |
| 6 | Click on Submit button without uploading Video Link | Kindly upload a Video.. | Successful |
| 7 | Click on Submit button without selecting Status | Kindly select the Status.. | Successful |

**Table 8.10 Training Material Testing**

* **Integrated Testing:**

This is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors with interfacing. Data can be lost across an interface; one module can have effect on other module; sub functionality may not be achieved. The testing was done with simple data. The developed data was run successfully with the test data. The need to integrate testing is to find overall system performance. All modules are combined and tested as a whole. Here errors uncovered are corrected for next testing steps.

**Integrated Testing Tables:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Test Case Objective** | **Test Case Description** | **Expected Result** |
| 1 | Check the interface link between Login and Dashboard module | Enter the login credentials and click on Sign in Button | To be directed to the dashboard |
| 2 | Check the interface link between View Branch and Delete Branch record | From View Branch, select the record and click on delete button | Selected record deleted from the database |
| 3 | Check the interface link between View Designation and Delete Designation record | From View Designation, select the record and click on delete button | Selected record deleted from the database |
| 4 | Check the interface link between View Discussion and Delete Discussion record | From View Discussion, select the record and click on delete button | Selected record deleted from the database |
| 5 | Check the interface link between View Employee and Delete Employee record | From View Employee, select the record and click on delete button | Selected record deleted from the database |
| 6 | Check the interface link between Inbox and Delete mail record | From Inbox, select the record and click on delete button | Selected record deleted from the database |
| 7 | Check the interface link between Sent mail and Delete mail record | From Sent mail, select the record and click on delete button | Selected record deleted from the database |
| 8 | Check the interface link between View Detailed News and Delete News record | From View Detailed News, select the record and click on delete button | Selected record deleted from the database |
| 9 | Check the interface link between View Subscribers and Delete Subscribers record | From View Subscribers, select the record and click on delete button | Selected record deleted from the database |
| 10 | Check the interface link between View Detailed Training material and Delete Training material record | From View Detailed Training material, select the record and click on delete button | Selected record deleted from the database |

**Table 8.11 Integration Testing**

* **System Testing:**

System testing is a stage of implementation. This is the first time end to end testing of application on the complete & fully integrated software product before it is launch to the market. This is to check whether the system works accurately and efficiently before the live operation commences. Testing is vital to the success of the system. The candidate system is subject to a variety of tests: on line response, volume, stress, recovery, security and usability test series of tests are performed for the proposed system is ready for user acceptance testing

**System Testing Tables:**

|  |  |  |
| --- | --- | --- |
| **SI.No** | **Test Condition** | **Test Results** |
| 1 | System Loading | Successful |
| 2 | System Run Procedure | Successful |
| 3 | File I/O Operation | Successful |
| 4 | Database communication | Successful |
| 5 | Server/Client interaction | Successful |
| 6 | Memory Usage | Normal |
| 7 | System Processor usage | Normal |
| 8 | Authentication/Authorization | Successful |

**Table 8.12 System Testing**

**Conclusion**

The project titled “SyndBank Employee Association System” has been designed and developed in PHP. Many user friendly controls are added to make it very interactive application. The project is tested successfully and report is generated. The project has satisfied all the requirements. The software has been developed in an attractive user interactive fashion.

The software has fulfilled all the objectives identified and provides an attractive user friendly interface. The reports requested by the user have been generated and all documentation required for operation and maintenance of the module has been provided.

**Limitations**

* It cannot be used as offline since it is online program.
* The Subscription amount module has virtual payment gateway and its not working like real-time payment gateway.
* Since it is made for only employee of 24x7 Bank, it cannot be used by the employees of other banks
* Program won’t work when server is down.

**Scope for Future Enhancement**

* In future we can add Web conference system feature.
* In future a mobile application will be made to provide user friendly environment for employees
* We can use cloud hosting for up gradation

**Bibliography**

**Text Books Referred**

1] An integrated approach to Software Engineering, PankajJalote, 2ndedition, Narosa Publishing House 2004

2] Software Testing Tools, Dr.K.V.K.K Prasad, Dreamtech press

3] Sams Teach Yourself PHP, MySQL and Apache All in One (5th Edition), Julie Meloni

4]Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites, Robin Nixon

**Web References**

PHP Tutorial -

http://www.w3schools.com/php/default.asp

http://www.tutorialspoint.com/php/

MySQL Tutorial -

http://www.w3schools.com/php/php\_mysql\_intro.asp

http://www.tutorialspoint.com/mysql/index.htm

JavaScript -

http://www.w3schools.com/js/default.asp

http://www.tutorialspoint.com/javascript/index.htm

CSS -

http://www.w3schools.com/css/default.asp

http://www.tutorialspoint.com/css/index.htm

HTML -

http://www.w3schools.com/html/default.asp

http://www.tutorialspoint.com/html/index.htm

HTML 5 - http://www.tutorialspoint.com/html5/index.htm

AJAX -

http://www.w3schools.com/ajax/default.asp

http://www.tutorialspoint.com/ajax/index.htm

Question and answer site:

www.stackoverflow.com