Contents

[Introduction/Objectives 2](#_Toc346468497)

[Requirement and Analysis 4](#_Toc346468498)

[Problem definition 4](#_Toc346468499)

[Requirement Specification 4](#_Toc346468500)

[Functional Requirements 4](#_Toc346468501)

[Gantt chart 11](#_Toc346468502)

[Tracking Gantt 11](#_Toc346468503)

[Pert chart 12](#_Toc346468504)

[Context Diagram 15](#_Toc346468505)

[DFD 15](#_Toc346468506)

[Level 0 15](#_Toc346468507)

[Level 1 16](#_Toc346468508)

[Level 2 17](#_Toc346468509)

[ER Diagram 18](#_Toc346468510)

[Class Diagram 19](#_Toc346468511)

[Module Description 23](#_Toc346468512)

[Estimation 26](#_Toc346468513)

[Data Structure 27](#_Toc346468514)

# Introduction/Objectives

Expense Manager Software will let users track expenses and earnings. This software is basically developed for individual users so that they can track their daily expense, income and keep track of their money but this application will be useful to a company as well. Users can add daily earnings & expenses. It is to know how and where we are spending our money. It is the perfect system for individuals, families, and small business to manage their income and expense tracking. The friendly user interface makes it quick and easy to capture your income and expense transactions for each day. Expenses management software updates all our accounts daily so we'll always have access to the most current and accurate information about our finances and can clearly see how much money we have and owe, and where is your hard-earn money going.

At any point of time he can review his total expenses and generate report. There will be three different User Interfaces to allow the users use the software anywhere all the time. The User interfaces are Desktop Interface, Mobile Interface & Web Interface. The data from all the interfaces can be synced and merged to generate a final report.

The Desktop Interface is the main & fully featured version of the software. Users can add new data, browse old expenses and sync expenses from Mobile & Web Interfaces. The Mobile Interface will allow users to add data using their mobile and use customized feature set. While shopping people carry their mobile along with them and then they can add expenses and earnings instantly to avoid forgetting about certain expenses. The Web Interfaces provide the ultimate flexibility of login to the user accounts in the web and add/browse expenditures.



Fig 1: Different interfaces of Expense Manager Software

The main features of this software are listed below:

1. Calendar view to select any date and add/view the expenses & earnings.
2. Add tag with expenses so that user can remember the reason for spending
3. View Available balance
4. View Total expenses
5. Create a contact book for add contact of the person.
6. Generate weekly, monthly and yearly Expenses report Income report Profit and Loss report.
7. Online sync
8. Reminder facility for future expenses.
9. Searching the expenses.
10. On the go expense tracking using Mobile & Web Interfaces
11. Syncing data from all the interfaces such as: Desktop, Mobile & Web.

 System Analysis

# Requirement and Analysis

## Problem definition

Now people are spending everywhere with multiple payment options. It is becoming very difficult to track expenses and earning. Almost every person is having a mobile device always with them. So this software will allow them to enter the data about earning & expenses in the mobile device. And they can sync the data with desktop and calculate the final details.

Incase user does not have any mobile device for data entry, he can also use web interface for entering the data. The data from web will be synced in desktop.

Desktop will allow additional features for managing expenses and earnings. EMS will manage expenses and earnings in an efficient manner.

EMS will allow you to set expense limit and it will notify while expensing.

## Requirement Specification

### Functional Requirements

#### Enter new Expenses and Earning

**Introduction**

Entered new expenses and new earning store into the account.

**Input**

Earning and expenses data with purpose of earning apply for which class.

**Processing**

EMS saves the expense and earning details in database.

**Output**

EMS generates expense id and earning id for future reference.

#### View Report for the Income, expense and period transaction

**Introduction**

User can view the report for particular or total income, expenses of the weekly, monthly, yearly, or period transaction.

**Input**

Select weekly, monthly, yearly, or period transaction.

**Processing**

EMS queries the expense and earning details from database and prepares the report.

**Output**

User can see the report.

#### Graphical representation for the Income, expense of weekly, monthly, yearly and PERIOD TRANSACTION

**Introduction**

User can view the graphically for particular income, expenses of the weekly, monthly, yearly, or period transaction using timing and total graphically representation of the weekly, monthly, yearly, or period transaction.

**Input**

Select weekly, monthly, yearly, or period transaction.

**Processing**

EMS queries the expense and earning details from database and prepares the data to be displayed.

**Output**

User can see the graphically representation.

#### Search transaction

**Introduction:**

Search transaction for income and expense.

**Input:**

Select timing, date, name of expenses or earning.

**Processing:**

The **EMS** will search for the requirement.

**Output:**

The **EMS** will display the search result.

#### Sync web & mobile data in desktop

**Introduction:**

Sync web & mobile data from desktop application

**Input:**

Select web account or mobile device

**Processing:**

The **EMS** will sync with web account or mobile device and save the data in database

**Output:**

The **EMS** will generate a sync id and display confirmation message.

#### Changing Password and Username

**Introduction**

Change existing username and password

**Input**

New username and password

**Processing**

Old username and password will be replaced by user provided new username and password after authenticating.

**Output**

Password and Username can be changed according to the Employee requirement whenever they want to change for better security of the System.

#### Mobile data entry & query

**Introduction:**

**EMS** data can be entered and queried using a mobile device.

**Input:**

User will enter the expenses and earning in the mobile device.

**Processing:**

The device will stored the entered data and sync with Server while manual sync operation. While querying device will search its internal storage for the query and display the result.

**Output:**

The mobile device will display the search result.

#### Web data entry & query

**Introduction:**

**EMS** data can be entered and queried using a web interface.

**Input:**

Admin will new user details as well as search query.

**Processing:**

Web interface will store new entry in the Google doc storage and while searching it will search its internal storage. Web interface will sync with main server while manual sync.

**Output:**

Website will show all the related information.

 Identification of Need

Expense Management provides the most complete and configurable full-service cloud solution for expense management and automation. Expense Management achieves greater control over costs, improved compliance and increased visibility into operations through data capture and reporting. It will help our self and company is improved control and management of overall expenditures.

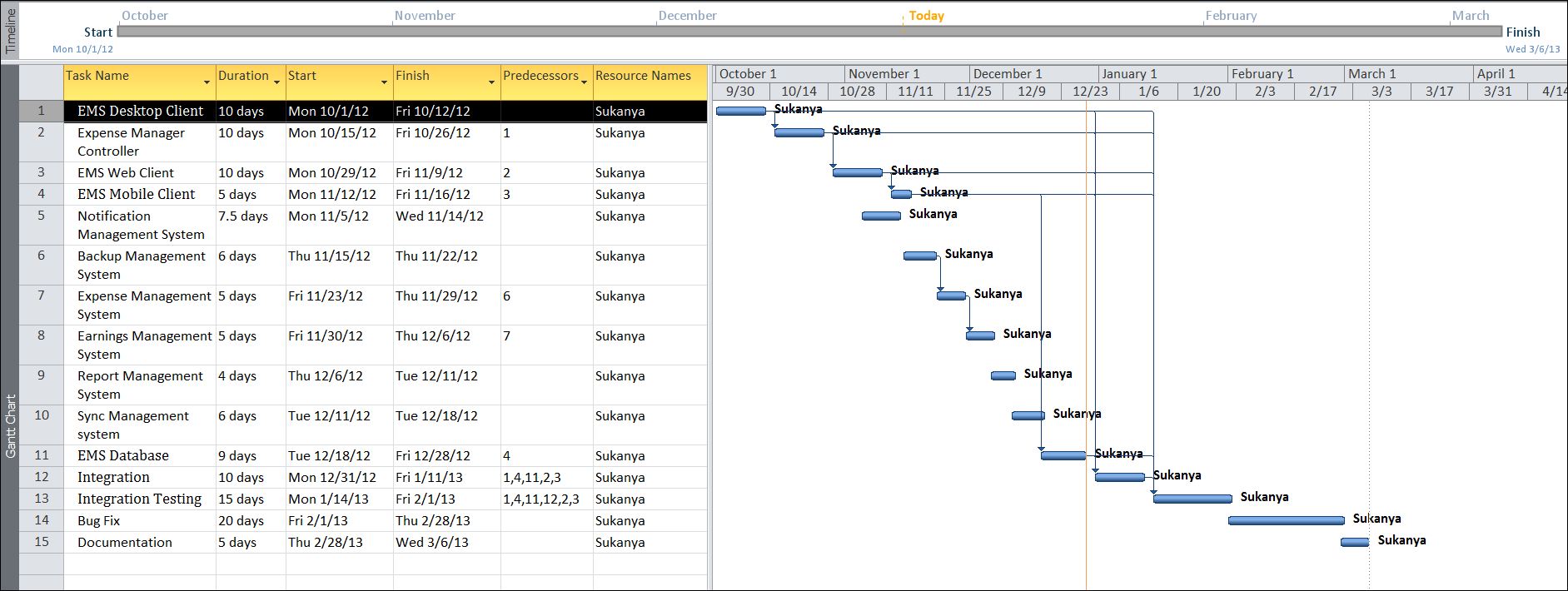
 Preliminary Investigation

 Feasibility Study

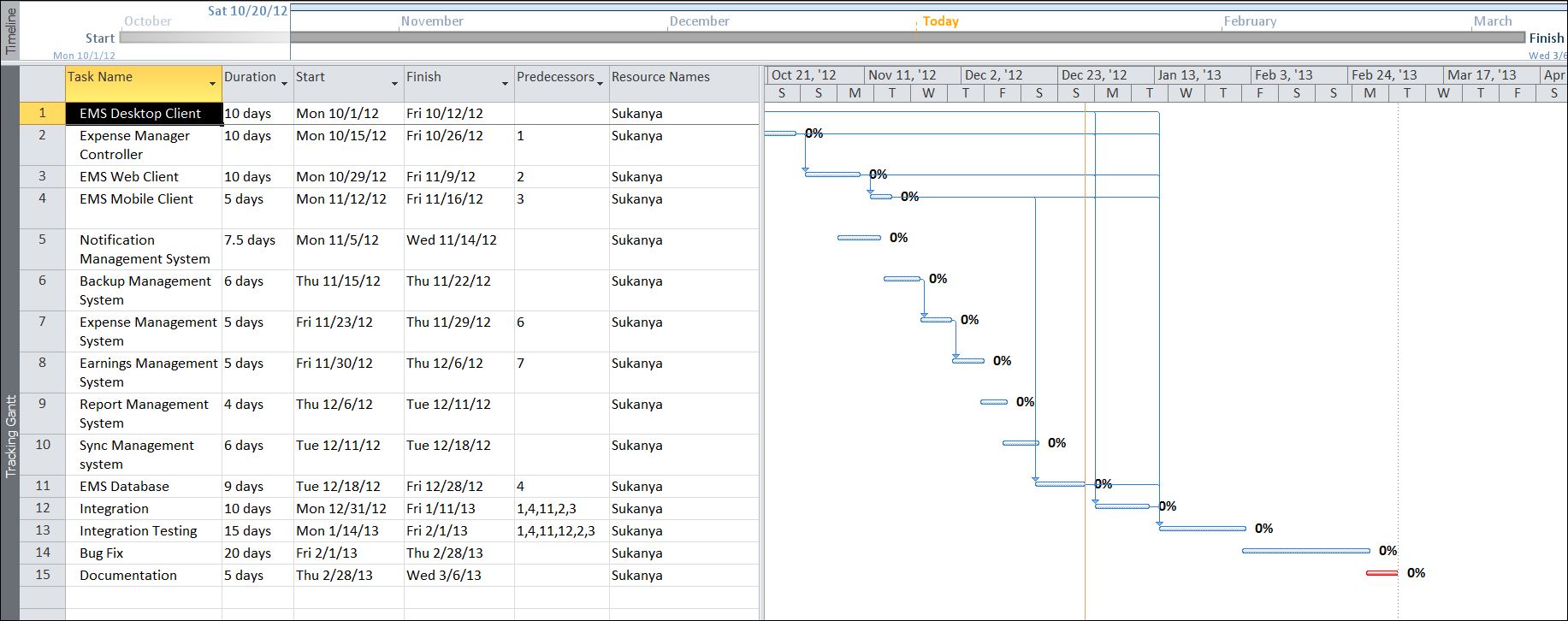
 Project Planning

 Project Scheduling (PERT Chart and Gantt Chart both)

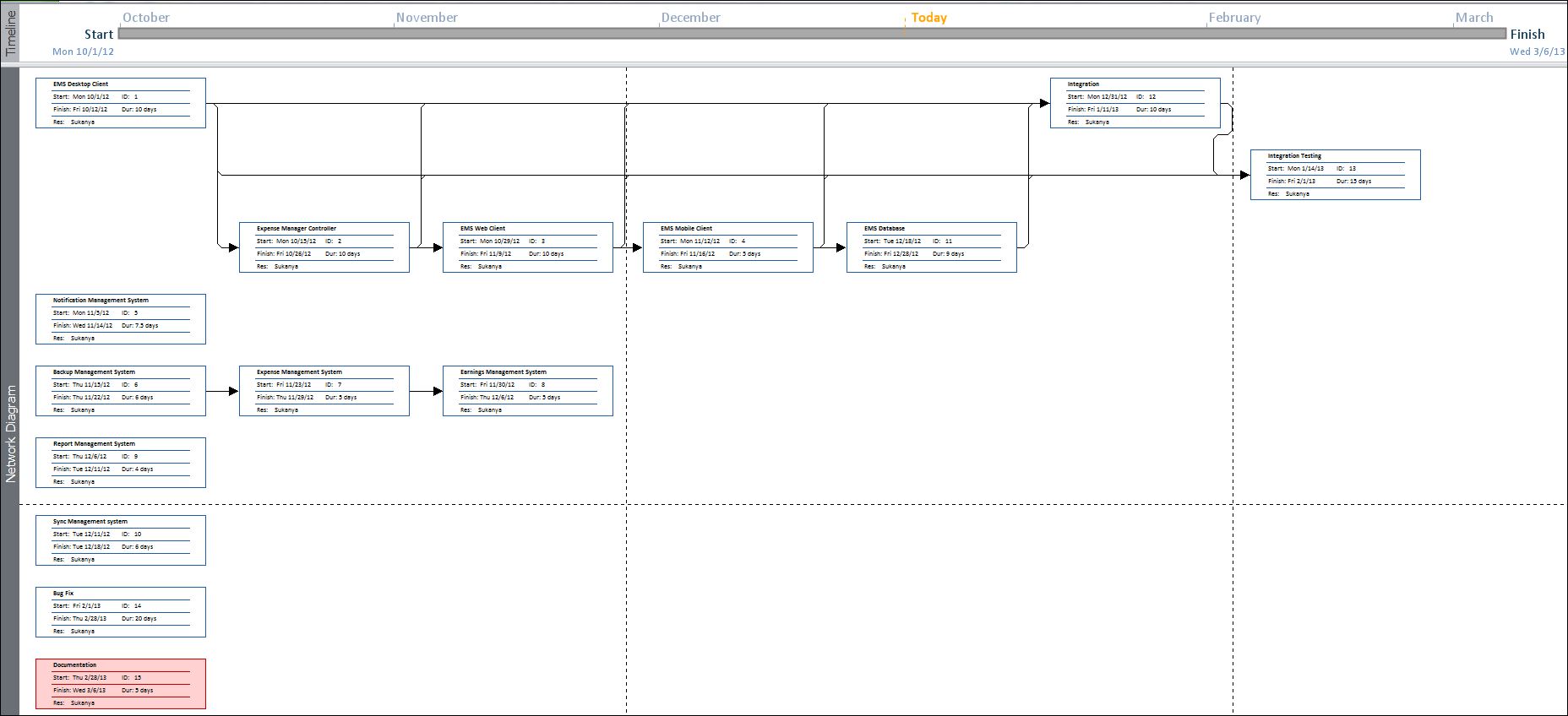
### Gantt chart



### Tracking Gantt



### Pert chart

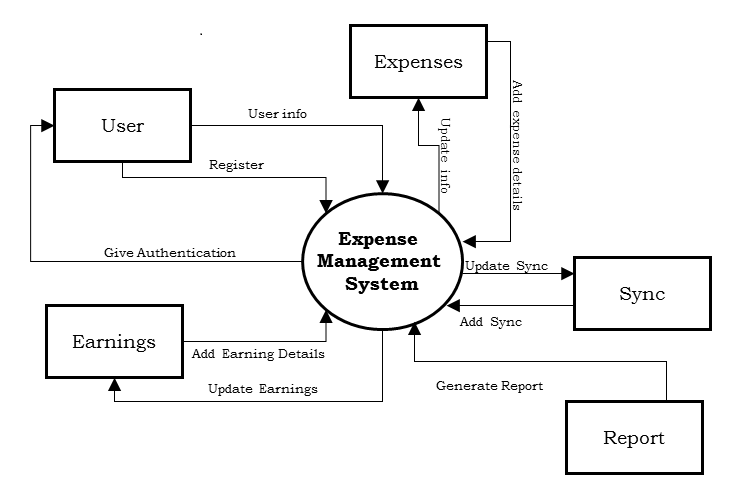


 Software requirement specifications (SRS)

 Software Engineering Paradigm applied

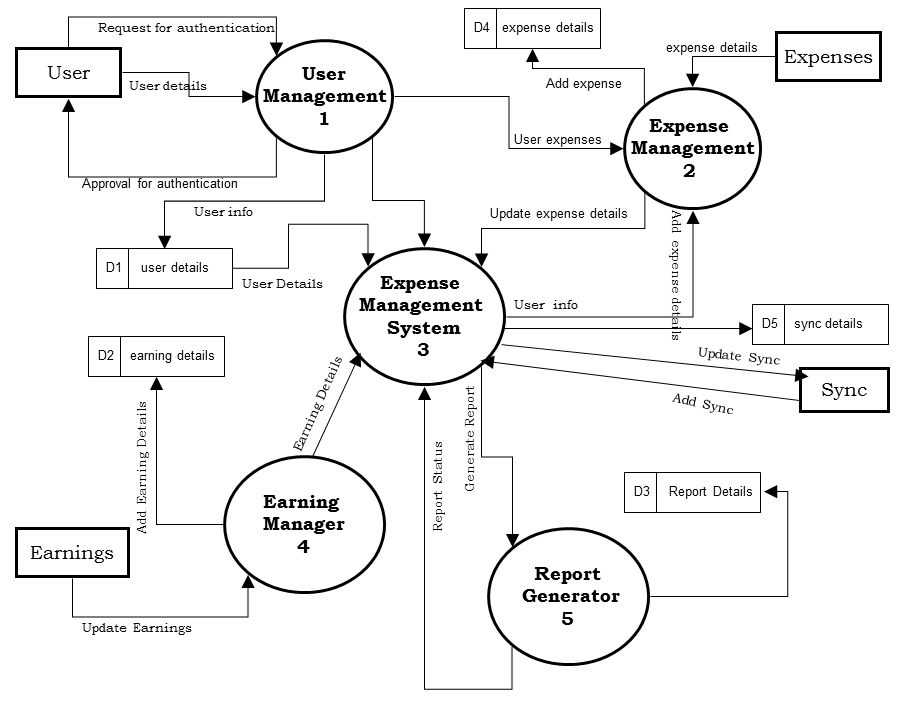
 Data models (like DFD), Control Flow diagrams, State Diagrams/Sequence diagrams, Entity Relationship Model, Class Diagrams/CRC Models/Collaboration Diagrams/Use-case Diagrams/Activity Diagrams depending upon your project requirements

## Context Diagram

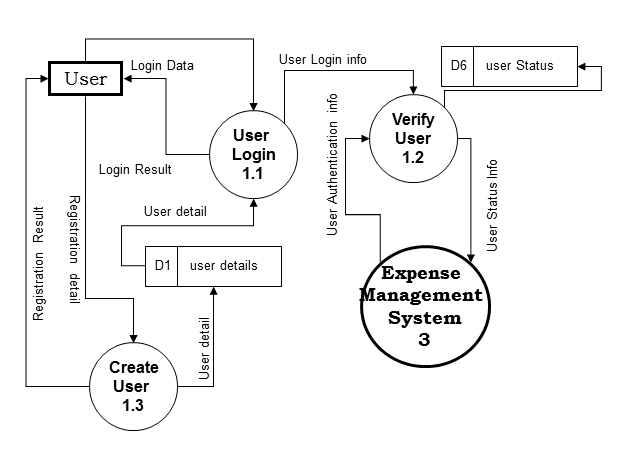


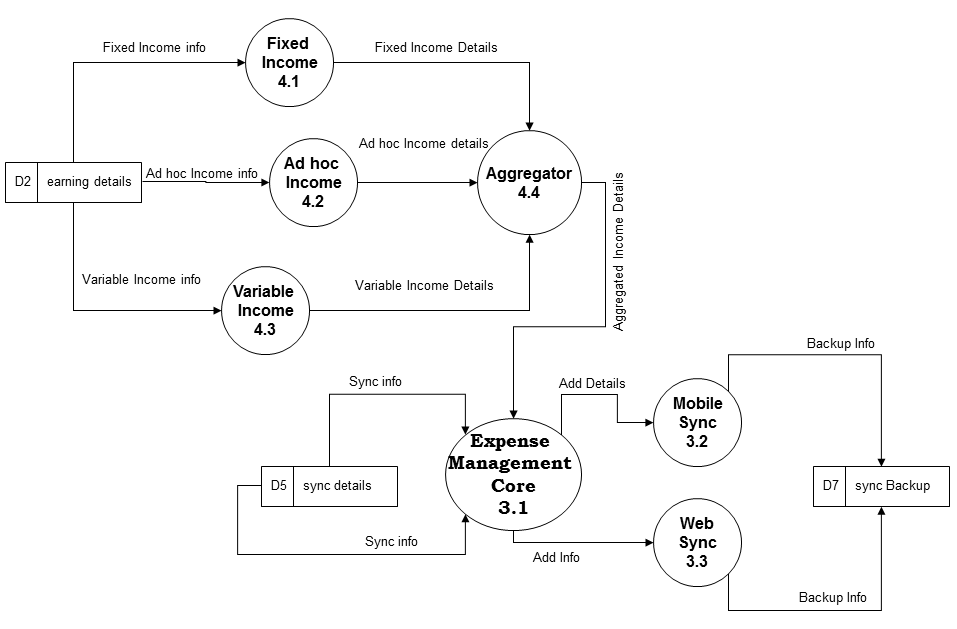
## DFD

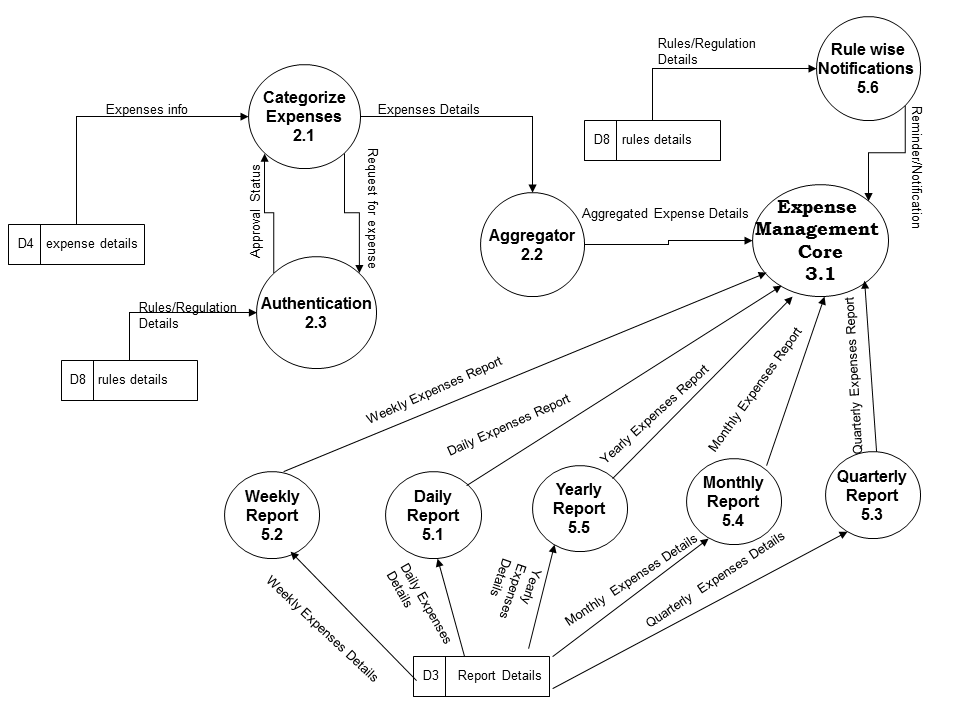
### Level 0



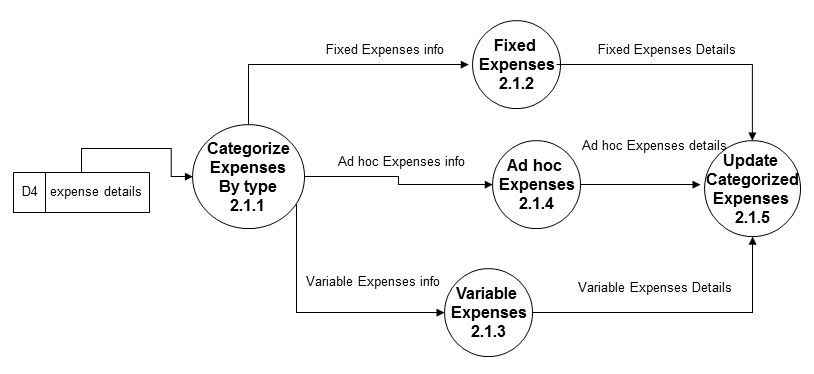
### Level 1







### Level 2



## ER Diagram

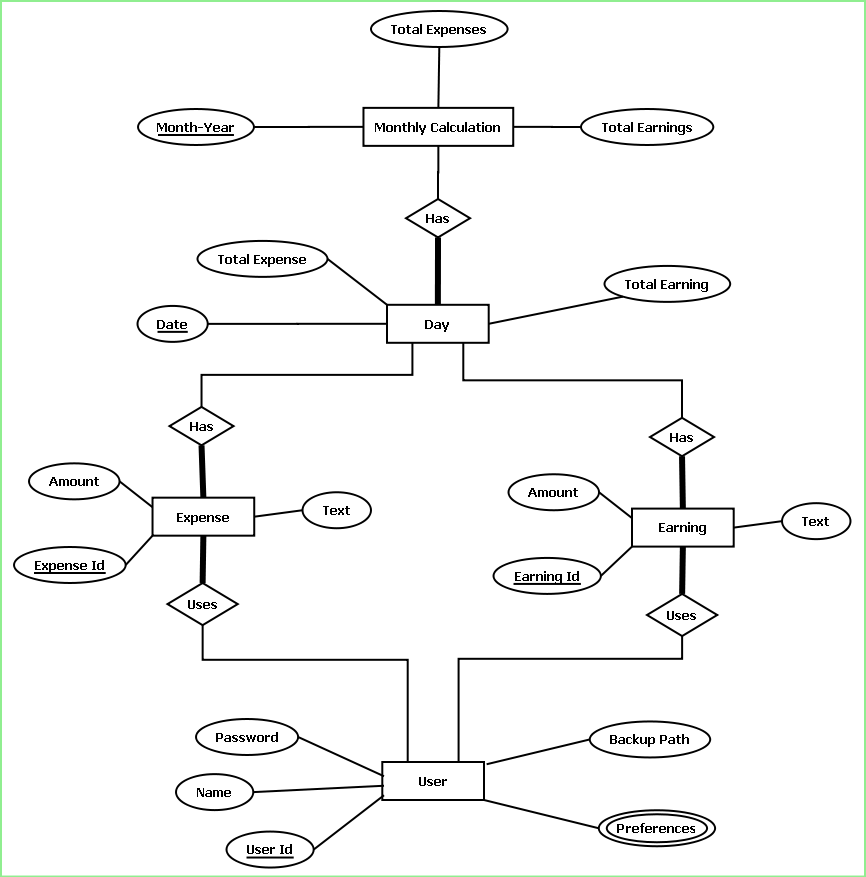
Expense Manager Database will be an optimized database which will save certain information about every expenses and earnings logged by User.

We will design a RDBMS for Expense Manager. The entities and their attributes are listed below. Attributes in Bold letter is the unique key.

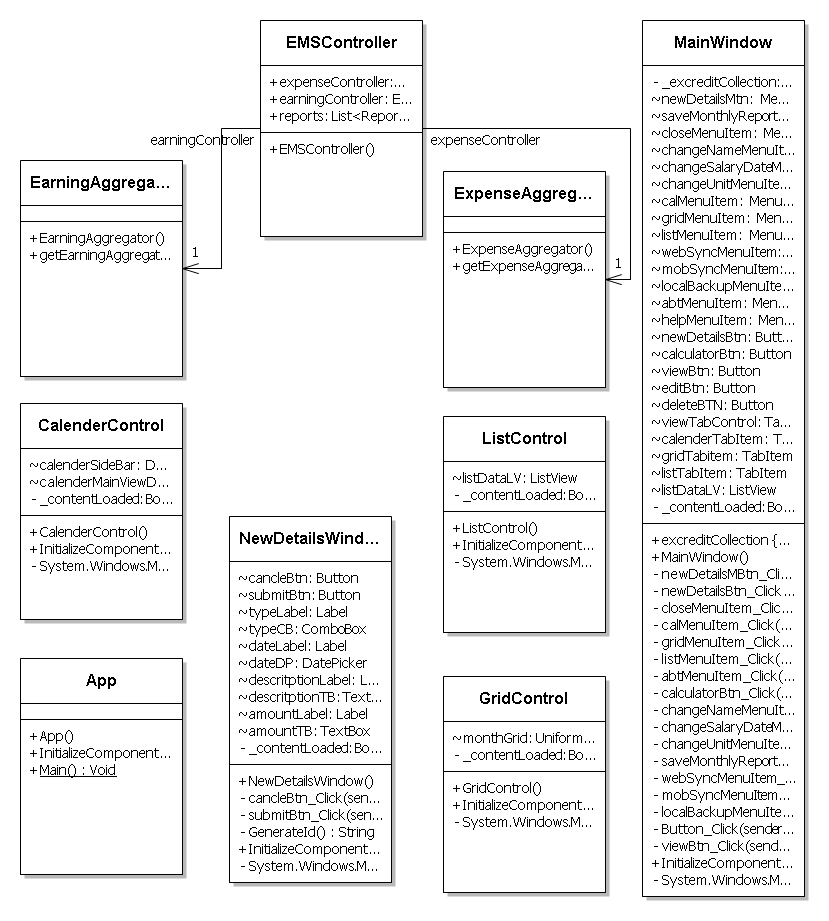
|  |  |
| --- | --- |
| **Entities** | **Attributes** |
| Expenses | **Expense Id,** Amount, Text |
| Earnings | **Earning Id,** Amount, Text |
| Monthly Calculation | **Month-Year**, Total Expense, Total Earning |
| Day | **Date**, Total Expense, Total Earning |
| User | **User Id**, Name, password, Backup Path, Preferences. |

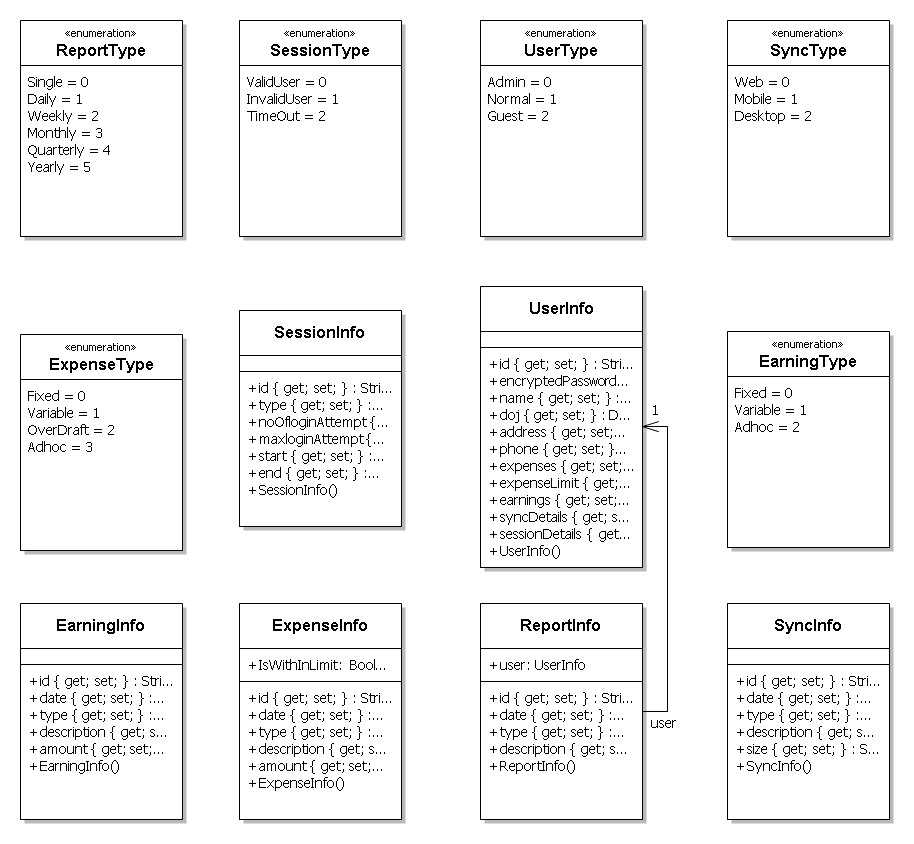
**Relationship between Entities:**

* Monthly Calculation has Day 🡪 1 : N
* In a **Day** happens **Expenses**🡪 1 : N
* In a **Day** happens **Earnings**🡪 1 : N
* **User** does **Expenses**🡪 M : N
* **User** does **Earnings**🡪 M : N



## Class Diagram





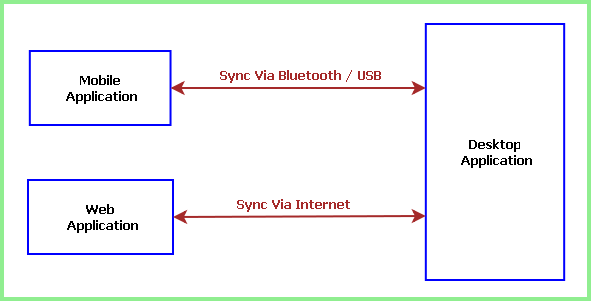
 System Design

 Modularisation details

## Module Description

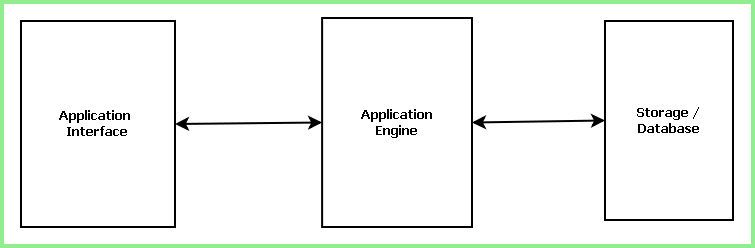
Expense manager software consists of three different applications:

* + Desktop Application
  + Mobile Application
  + Web Application



Each application is internally divided into three main modules such as:

* Application Interface
* Application Engine
* Storage / Database



Desktop Application is the full featured application which contains the Permanent storage or bigger database where as Mobile and Web application has a small & temporary storage. People can note down their expenses while roaming, at their mobile using expense manager. They can later sync and take the backup of their expenses and get a final report. They can sync with the web interface and store the data in the online database from where their important data would never be lost. Thus using three different interface and database data will not only be easy to maintain but also secure forever.

**Desktop Application Module:** It consists of three main parts, namely the GUI module, Engine/controller module and storage module.

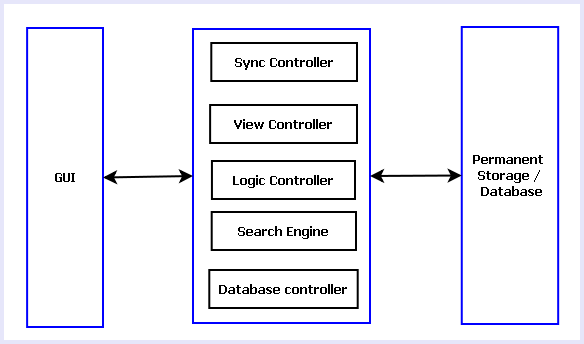


Fig: Modules of Desktop Application

**Desktop Application GUI:** The GUI for the desktop application will be designed using WPF (Windows Presentation Foundation) and XAML (Extensive Advanced Markup Language). The GUI will have several views like Calendar view, List view, Grid View. It will have options foe adding new expenses and earnings, searching for expenses/earnings, adding remainder for future expenses, syncing with mobile/web application.

**Desktop Application Engine:** Desktop Application Engine is the heart of the application. It controls the GUI interactions, logical calculations and database queries. It consists of 5 sub modules, such as:

**View Controller:**

It controls the look and feel of GUI. As mentioned earlier, the GUI will have three different views: List View, calendar View and Grid View.

List view will display data as a list with columns for date, tag text and amount. List view can be sorted by the columns.

In calendar view, user can select any date and data associated with the date will be displayed. Calendar view has three varieties like daily, weekly and monthly views.

Grid view will display the available data in contiguous grids. Each grid will display tag text, amount and date. The components of view controller are shown in the diagram below.

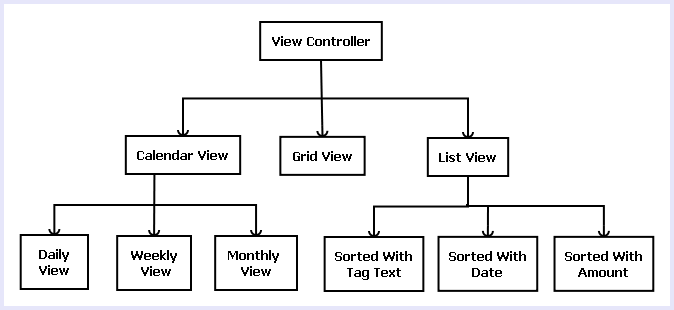


Fig: Various parts of View Controller

**Sync Controller:**

Sync controller handles the synchronization of data with mobile and web application. Sync controller receives data from other applications, processes data and saves data for future use.

**Logic Controller:**

Logic controller manages all the modules of application engine. It handles the interaction between other modules. The instances of all other module are created in logic controller so that it can control them. The application logics are written in this module.

**Search Engine:**

Search engine helps the user to search available data. It will have options for searching by tag text, amount and date. Search engine will form a query depending on the user input and fetch the result from database.

**Database Controller:**

Database controller handles the database interaction. It takes care of database addition, modification, deleting and retrieval of data from storage/ database.

**Desktop Application Storage:** Desktop application storage is the permanent storage/ database of the expense manager software. This module will be implemented using MySQL. The module sill store all the data related to this application. Users will be able to add, modify, erase and fetch/ view data on runtime.

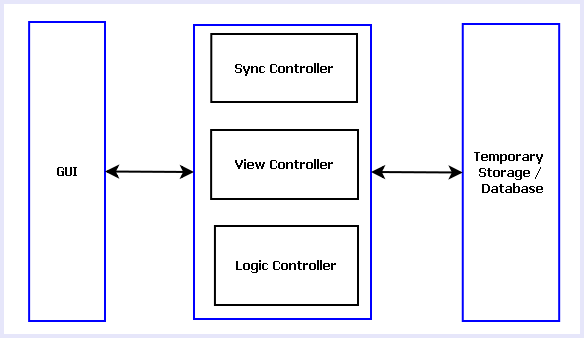
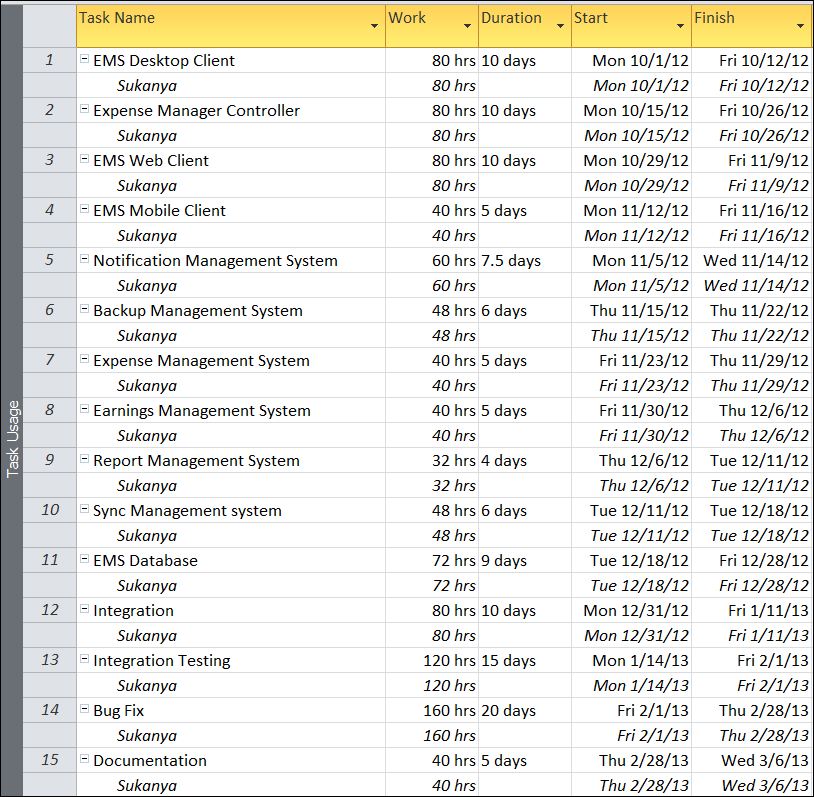


Fig: Modules of Mobile & Web Application

## Estimation



 Data integrity and constraints

## Data Structure

|  |
| --- |
| **ExpenseInfo** |
| publicclassExpenseInfo  {  publicstring id { get; set; }  publicDateTime date { get; set; }  publicExpenseType type { get; set; }  publicstring description { get; set; }  publicdouble amount { get; set; }  publicboolIsWithInLimit;  } |

|  |
| --- |
| **EarningInfo** |
| publicclassEarningInfo  {  publicstring id { get; set; }  publicDateTime date { get; set; }  publicEarningType type { get; set; }  publicstring description { get; set; }  publicdouble amount { get; set; }  } |

|  |
| --- |
| **SyncInfo** |
| publicclassSyncInfo  {  publicstring id { get; set; }  publicDateTime date { get; set; }  publicSyncType type { get; set; }  publicstring description { get; set; }  publicfloat size { get; set; }  } |

|  |
| --- |
| **UserInfo** |
| publicclassUserInfo  {  publicstring id { get; set; }  publicstringencryptedPassword { get; set; }  publicstring name { get; set; }  publicDateTimedoj { get; set; }  publicstring address { get; set; }  publicstring phone { get; set; }  publicList<ExpenseInfo> expenses { get; set; }  publicdoubleexpenseLimit { get; set; }  publicList<EarningInfo> earnings { get; set; }  publicList<SyncInfo>syncDetails { get; set; }  publicSessionInfosessionDetails { get; set; }  } |

|  |
| --- |
| **ReportInfo** |
| publicclassReportInfo  {  publicstring id { get; set; }  publicDateTime date { get; set; }  publicReportType type { get; set; }  publicstring description { get; set; }  publicUserInfo user;  } |

|  |
| --- |
| **SessionInfo** |
| publicclassSessionInfo  {  publicstring id { get; set; }  publicSessionType type { get; set; }  publicintnoOfloginAttempt { get; set; }  publicintmaxloginAttempt { get; set; }  publicDateTime start { get; set; }  publicDateTime end { get; set; }  } |

|  |
| --- |
| **ReportType** |
| publicenumReportType  {  Single,  Daily,  Weekly,  Monthly,  Quarterly,  Yearly  } |

|  |
| --- |
| **SessionType** |
| publicenumSessionType  {  ValidUser,  InvalidUser,  TimeOut  } |

|  |
| --- |
| **UserType** |
| publicenumUserType  {  Admin,  Normal,  Guest  } |

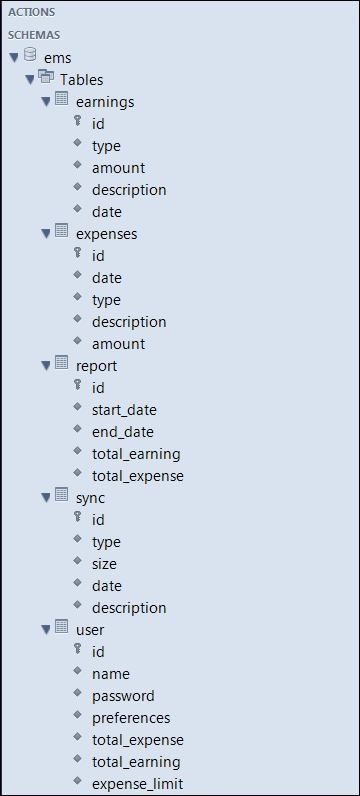
|  |
| --- |
| **SyncType** |
| publicenumSyncType  {  Web,  Mobile,  Desktop  } |

|  |
| --- |
| **EarningType** |
| publicenumEarningType  {  Fixed,  Variable,  Adhoc  } |

|  |
| --- |
| **ExpenseType** |
| publicenumExpenseType  {  Fixed,  Variable,  OverDraft,  Adhoc  } |

 Database design, Procedural Design/Object Oriented Design

The database used for this software is called **emsdb**. A screenshot from the MySQl workbench is given below. It shows the tables and its columns. The first row is the primary key.



 User Interface Design

 Test Cases (Unit Test Cases and System Test Cases)

 Coding

 Complete Project Coding

 Comments and Description of Coding segments

 Standardization of the coding

 Code Efficiency

 Error handling

 Parameters calling/passing

 Validation checks

 Testing

 Testing techniques and Testing strategies used

 Testing Plan used

 Test reports for Unit Test Cases and System Test Cases

 Debugging and Code improvement

 System Security measures (Implementation of security for the project developed)

* + This software requires a valid password to login and then it allows using any of its features.
  + The login password will be saved in encrypted format in database.
  + This software will use Google open-id authentication for web interface.
  + A backup and restore feature has been used in case of loss of data due to database crash and other problems.

 Database/data security

 Creation of User profiles and access rights

 Cost Estimation of the Project along with Cost Estimation Model

 Reports (sample layouts should be placed)

 Future scope and further enhancement of the Project

* + To support UNIX / Linux based operating systems.
  + To Support Mobile operating systems for Windows Mobile, Nokia, Blackberry.
  + To port it on handheld device like iPad, Galaxy Tab & netbooks.
  + To enhance the web interface

 Bibliography

* + http://en.wikipedia.org
  + http://msdn.microsoft.com/en-us/
  + http://www.microsoft.com/en-us/default.aspx
  + http://www.codeplex.com/
  + http://stackoverflow.com/
  + http://www.codeguru.com/
  + http://www.w3schools.com
  + www.mysql.org

 Appendices (if any)

 Glossary.