[](http://www.ignou.ac.in/)**INDIRA GANDHI NATIONAL OPEN UNIVERSITY**

**MASTER OF COMPUTER APPLICATIONS (MCA)**

**(6TH SEMESTER)**

***SYNOPSIS***

**Expense Manager**

**SUBMITTED BY**

***SUKANYA HALDER***

# Table of Contents

[Table of Contents 2](#_Toc344418418)

[Introduction & Objective of the Project: 3](#_Toc344418419)

[Project Category: 4](#_Toc344418420)

[*Tools/Platforms used (Hardware/Software):* 5](#_Toc344418421)

[Analysis of the Project: 8](#_Toc344418422)

[2. Overview of the Project: 12](#_Toc344418423)

[Implementation of Security Mechanism at Various Levels 13](#_Toc344418424)

[3. Future Scope and Further enhancement of the project: 14](#_Toc344418425)

[Bibliography 14](#_Toc344418426)

# Introduction & Objective of the Project:

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.

# Project Category:

This software will follow Object Oriented Programming Paradigm and use below mentioned areas:

**Front End/ GUI Tools:** WPF (Windows Presentation Foundation), XAML

**Programming Language:** C#

**IDE:** Visual Studio 2010

**Development Environment:** .NET 4

**Backend:** MySQL

**IDE (For Database):** MySQL WorkBench

**Source Code Management Tool:** Github

**Internet Technologies:** HTML, JavaScript, PHP, Google App Framework

**Networking Technologies:** TCP/IP

**Wireless Technologies:** Bluetooth

**Operating Systems**: Windows XP, Windows 7, Windows Mobile, Symbian Mobile

**Applications**: Financial application

# Hardware and Software Specification

## Hardware Requirement

* ***Disc capacity :*** *10 MB of available hard disk space*
* ***RAM :*** *1 GB (32 Bit) or 2 GB (64 Bit)*
* ***Processor :*** *1.6GHz or faster*
* *DVD-ROM Drive / USB* ***Port***

## Software Requirement

* *Windows XP (x86) with Service Pack 3 / Windows Vista (x86 & x64) with*

*Service Pack 2 / Windows 7 (x86 & x64)*

* *Microsoft .NET 4.0*

# *Tools/Platforms used (Hardware/Software):*

Software Requirement Specification:

Desktop Application

|  |  |  |
| --- | --- | --- |
| PLATFORM | TOOLS | PURPOSE |
| Windows 7 | .NET Framework | Development Environment |
| Visual Studio 2008 | C# | Writing Program |
| MySQL |  | Database Design |
| MS Word | Documentation | Documentation |

Web Application

|  |  |  |
| --- | --- | --- |
| PLATFORM | TOOLS | PURPOSE |
| Windows 7 | .NET Framework |  |
|  | PHP | Writing Program |
| MySQL |  | Database Design |
| MS Word | Documentation | Documentation |

Mobile Application

|  |  |  |
| --- | --- | --- |
| PLATFORM | TOOLS | PURPOSE |
| Windows 7 | .NET Framework |  |
| NetBeans | JSP | Writing Program |
| MySQL |  | Database Design |
| MS Word | Documentation | Documentation |

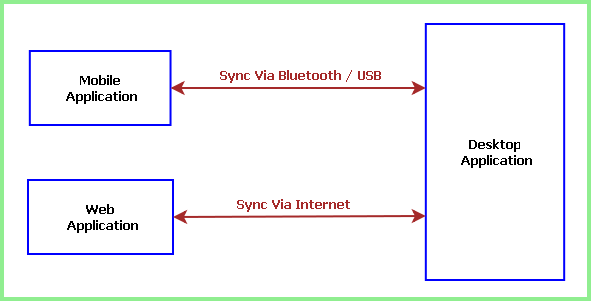
**Hardware Requirement Specification:**

|  |  |
| --- | --- |
| Processor | Pentium-5 (2.50 GHz) |
| RAM | 3GB |
| Hard Disk | 380GB |
| Mother Board | Intel |

# Analysis of the Project:

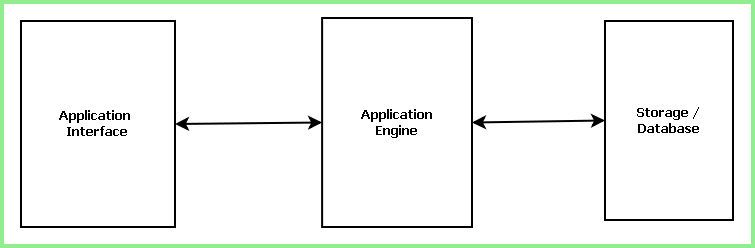
Expense manager software consists of three different applications:

* + 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.

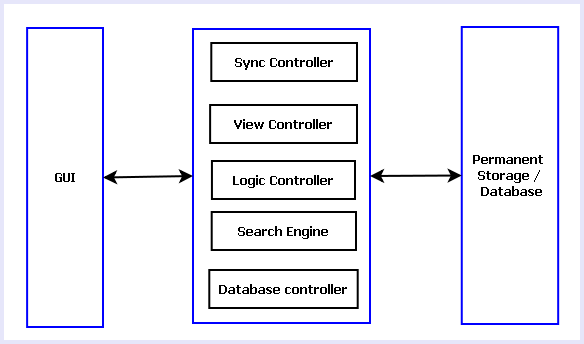


Fig: Modules of Desktop Application

**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.

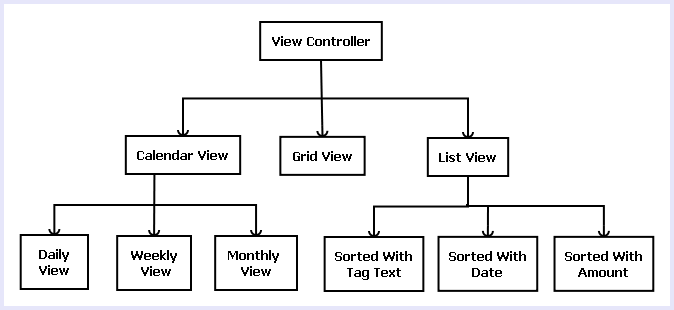


Fig: Various parts of View Controller

**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.

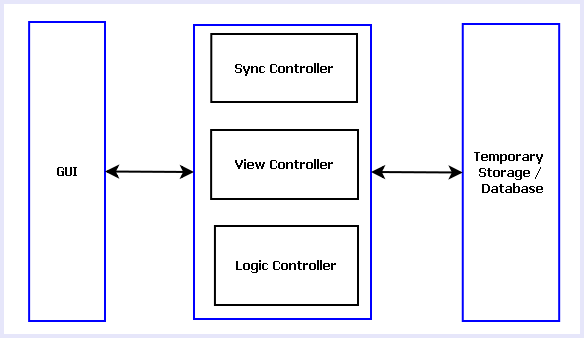


Fig: Modules of Mobile & Web Application

# Hardware and Software Specification

## Hardware Requirement

* ***Disc capacity :*** *10 MB of available hard disk space*
* ***RAM :*** *1 GB (32 Bit) or 2 GB (64 Bit)*
* ***Processor :*** *1.6GHz or faster*
* *DVD-ROM Drive / USB* ***Port***

## Software Requirement

* *Windows XP (x86) with Service Pack 3 / Windows Vista (x86 & x64) with*

*Service Pack 2 / Windows 7 (x86 & x64)*

* *Microsoft .NET 4.0*

Requirement and Analysis

Problem definition

Requirement Specification:

Technical Specification:

Planning and Scheduling:

Gantt chart:

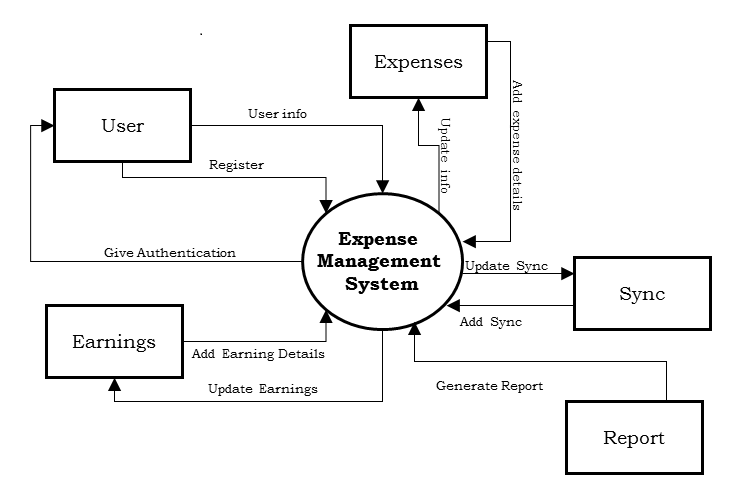
Tracking Gantt

Pert chart

# Scope of the Solution:

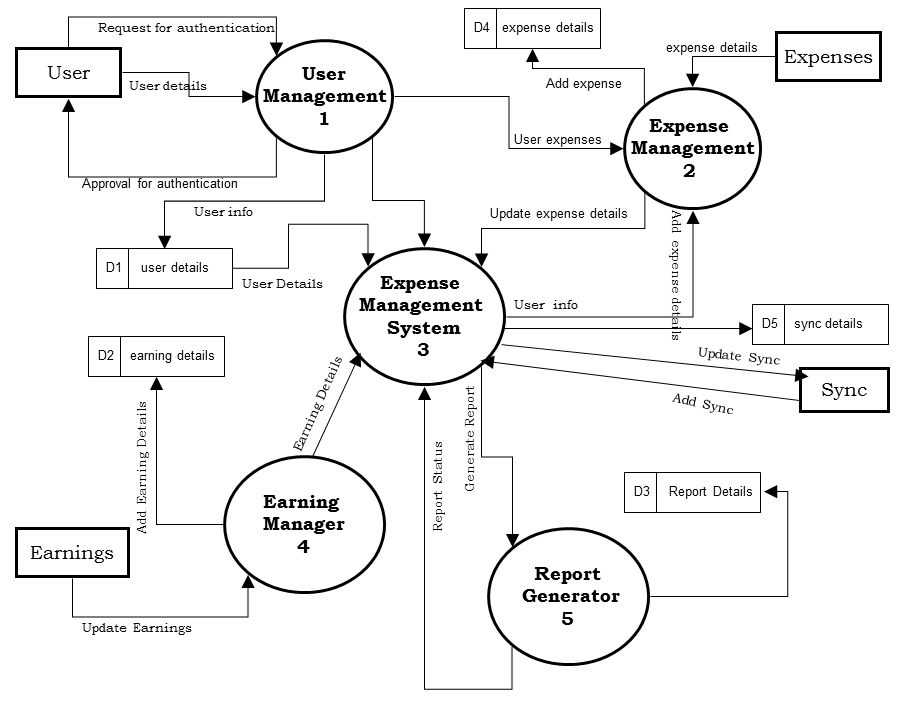
# Analysis:

## Context Diagram

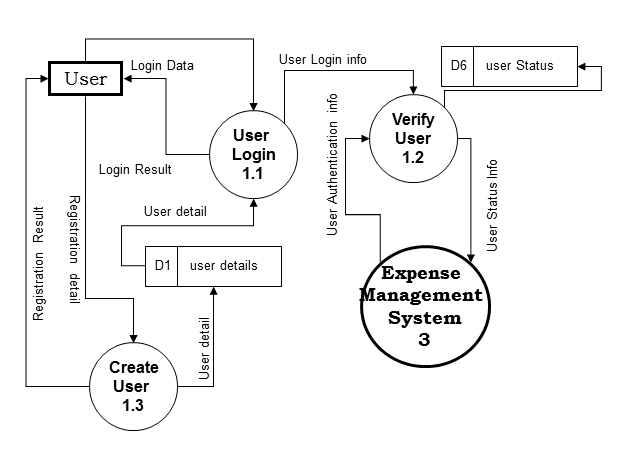


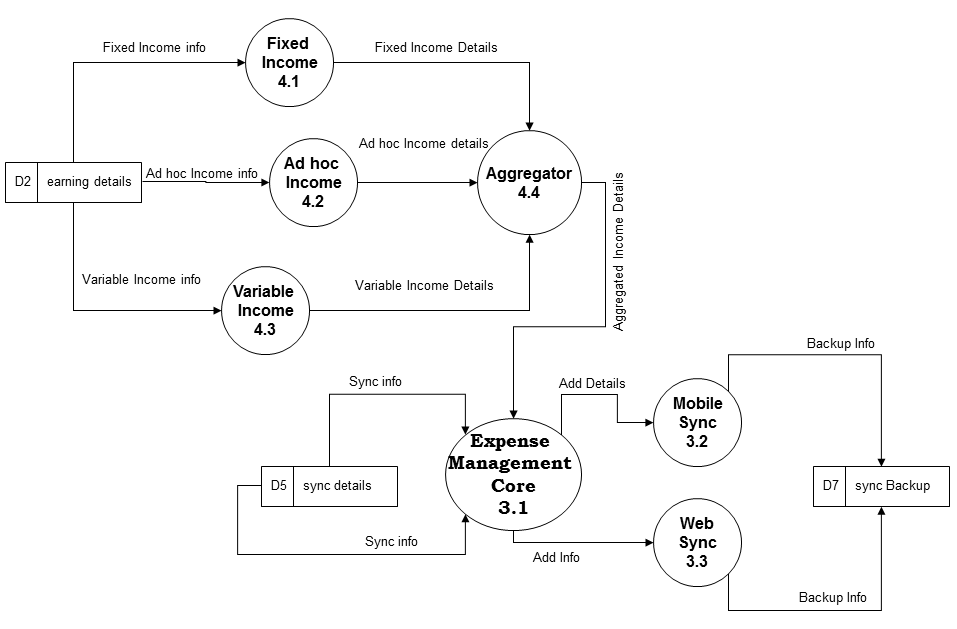
## DFD

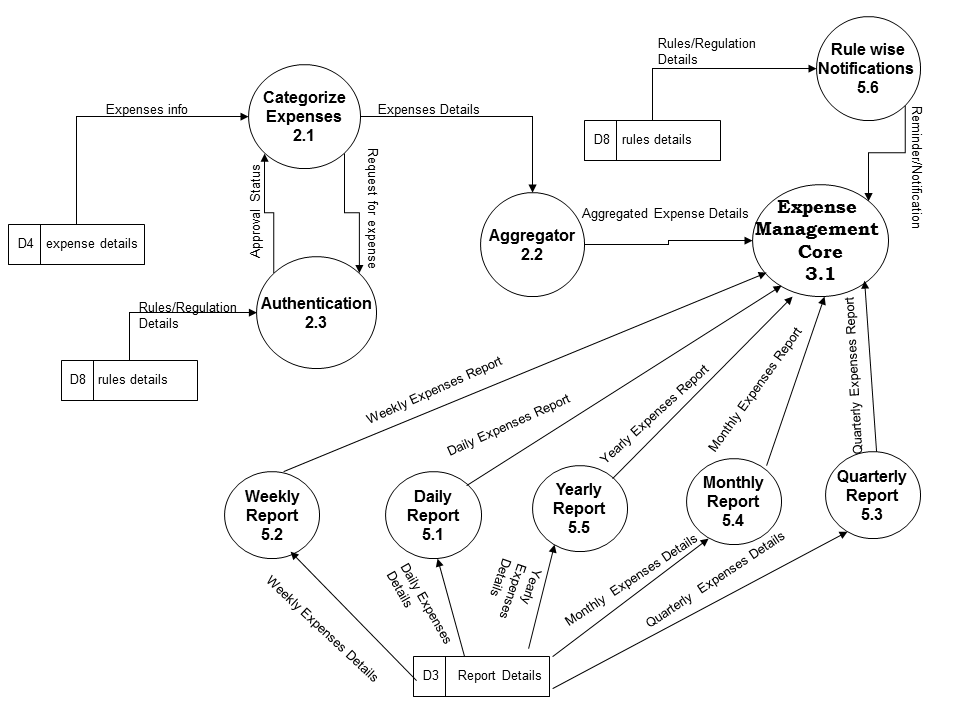
### Level 0



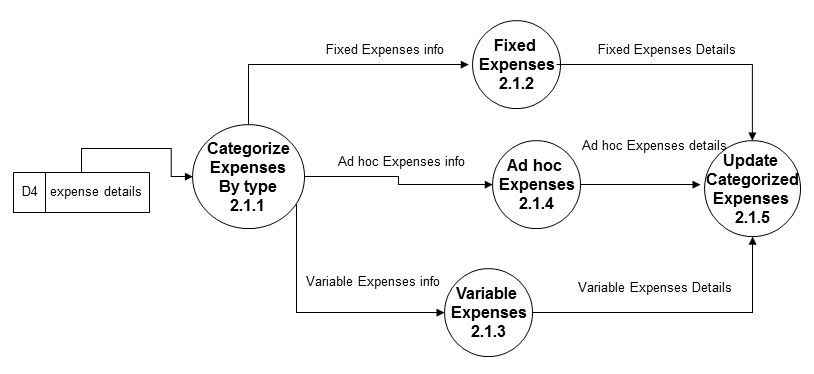
### Level 1





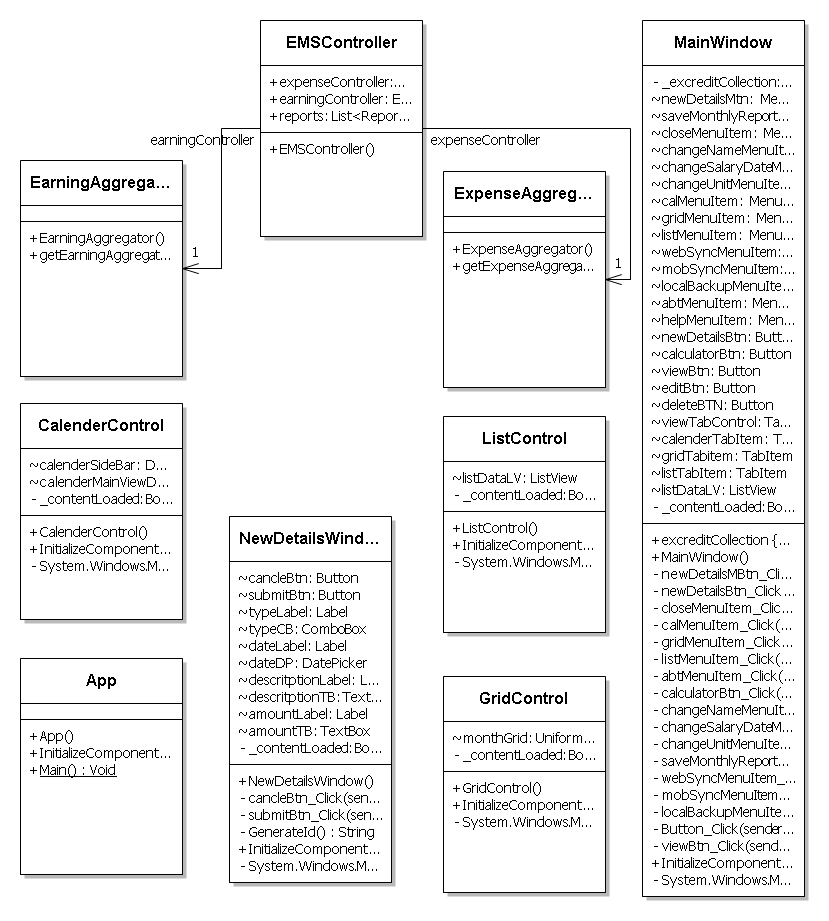


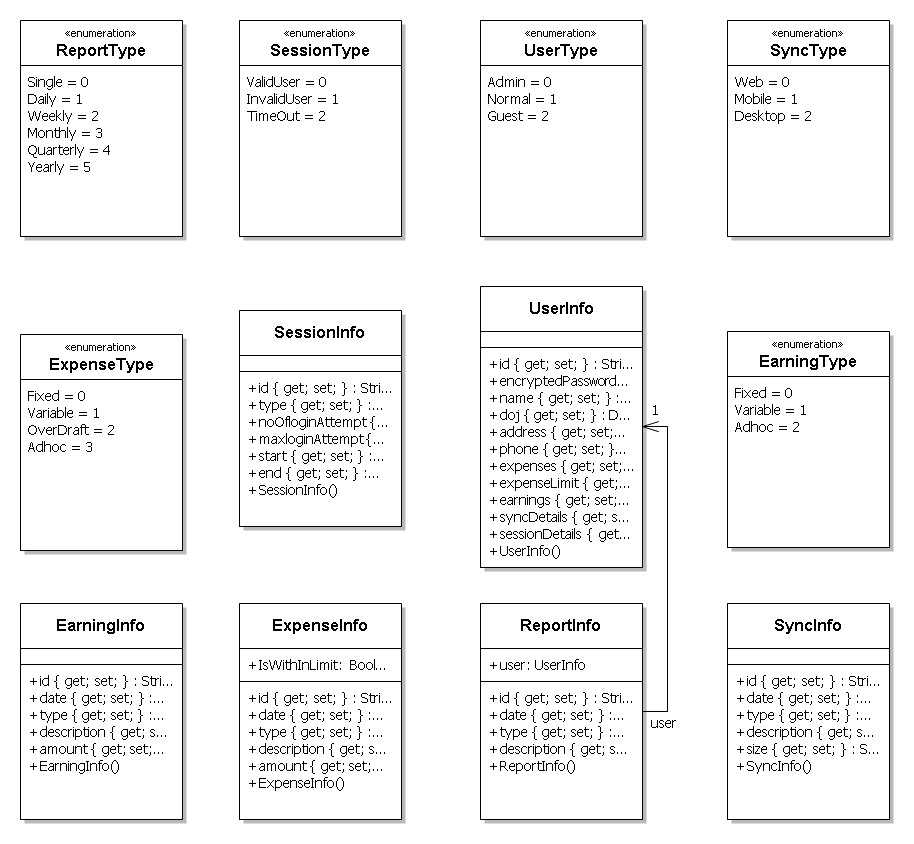
### Level 2



## ER Diagram

## Class Diagram





# Database and Table Details

# Complete structure:

## Module Description

## Estimation:

## Data Structure:

|  |
| --- |
| **ExpenseInfo** |
| public class ExpenseInfo  {  public string id { get; set; }  public DateTime date { get; set; }  public ExpenseType type { get; set; }  public string description { get; set; }  public double amount { get; set; }  public bool IsWithInLimit;  } |

|  |
| --- |
| **EarningInfo** |
| public class EarningInfo  {  public string id { get; set; }  public DateTime date { get; set; }  public EarningType type { get; set; }  public string description { get; set; }  public double amount { get; set; }  } |

|  |
| --- |
| **SyncInfo** |
| public class SyncInfo  {  public string id { get; set; }  public DateTime date { get; set; }  public SyncType type { get; set; }  public string description { get; set; }  public float size { get; set; }  } |

|  |
| --- |
| **UserInfo** |
| public class UserInfo  {  public string id { get; set; }  public string encryptedPassword { get; set; }  public string name { get; set; }  public DateTime doj { get; set; }  public string address { get; set; }  public string phone { get; set; }  public List<ExpenseInfo> expenses { get; set; }  public double expenseLimit { get; set; }  public List<EarningInfo> earnings { get; set; }  public List<SyncInfo> syncDetails { get; set; }  public SessionInfo sessionDetails { get; set; }  } |

|  |
| --- |
| **ReportInfo** |
| public class ReportInfo  {  public string id { get; set; }  public DateTime date { get; set; }  public ReportType type { get; set; }  public string description { get; set; }  public UserInfo user;  } |

|  |
| --- |
| **SessionInfo** |
| public class SessionInfo  {  public string id { get; set; }  public SessionType type { get; set; }  public int noOfloginAttempt { get; set; }  public int maxloginAttempt { get; set; }  public DateTime start { get; set; }  public DateTime end { get; set; }  } |

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

|  |
| --- |
| **SessionType** |
| public enum SessionType  {  ValidUser,  InvalidUser,  TimeOut  } |

|  |
| --- |
| **UserType** |
| public enum UserType  {  Admin,  Normal,  Guest  } |

|  |
| --- |
| **SyncType** |
| public enum SyncType  {  Web,  Mobile,  Desktop  } |

|  |
| --- |
| **EarningType** |
| public enum EarningType  {  Fixed,  Variable,  Adhoc  } |

|  |
| --- |
| **ExpenseType** |
| public enum ExpenseType  {  Fixed,  Variable,  OverDraft,  Adhoc  } |

## Implementation Methodology

* Object Oriented Programming methodology will be adopted and Java will be used as programming language.
* Apache tomcat web server will be used to implement the server
* User interface development will be done in MVC architecture using SWT (Standard Widget Toolkit).
* Relational DBMS MySQL will be used to implement & execute SQL query to database.
* Agile Software Development model will be used while developing this software.

## List of Reports

List of reports that are likely to be generated in this software are given below:

* Results & Grade report can be generated
* Class toppers will be given a certificate as a token of appreciation
* List of students can be generated
* List of teachers can be generated
* List of courses can be generated
* Course details can be generated
* Fund details can be generated
* Yearly donation report can be generated
* Salary slips can be created

# Implementation of Security Mechanism at Various Levels

* 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.

# 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*