**TITLE**

Smart Expense Ledger

**1. INTRODUCTION**

**1.1 Problem Definition**

The Smart Expense Ledger is A Mobile Application. This project has to implement and gather information for the daily expense in a more manageable way. This application can be used to reduce the manual calculations for daily expenses. One can use this application to create their expense that occurs on daily, monthly, yearly basis. More than that, one can share expenses in a group environment, for their day today needs as a group they spend and account the expenses. This application is especially useful for college students where they form smaller invisible groups to share their expenses in a closed circle and settle on need basis. This app can capture all the expenses in a group share each expense on a percentage or a specific amount to others. The uniqueness of this application is to capture the debts one has with others as well credits as reports.

**1.2 Purpose and Scope of the Problem**

The purpose of the document is to describe the major think is Expense Ledger for group of friends, Group of member etc. before the college students want to check the details manually. The Scope of The document is to describe the Smart Expense Ledger. The group of people can search their expense with the help of like. It is very useful to approach requirements is defined in the System Requirement Specification. The Group of members can views the details about the reporting talking expenses automatically.

2. **PROJECT MILESTONE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Development phase | 5 Months | | | | | Duration |
| 2 Weeks | 4 Weeks | 5 Weeks | 5 Weeks | 4 Weeks |
| Requirement Gathering &  Analysis |  | | | | | 14 Days |
| Designing |  | | | | | 28 Days |
| Coding |  | | | | | 35 Days |
| Testing |  | | | | | 35 days |
| Implementation |  | | | | | 21 Days |
| Documentation |  | | | | | Parallel |
| Total(Months/Days) | 5 Months | | | | |  |

**3. PROJECT OVERVIEW**

The project will be using a smart approach to manage expenses. The college students want to see the Smart Expense Ledger in his or her into login only, once this app using group of members. This application can be used to reduce the manual calculations for daily expenses. One can use this application to create their expense that occurs on daily, monthly, yearly basis. More than that, one can share expenses in a group environment, for their day today needs as a group they spend and account the expenses. This application is especially useful for college students where they form smaller invisible groups to share their expenses in a closed circle and settle on need basis. This app can capture all the expenses in a group share each expense on a percentage or a specific amount to others. The uniqueness of this application is to capture the debts one has with others as well credits as reports.

**4. SYSTEM REQUIREMENT SPECIFICATION**

**4.1 Functional Requirement**

* User Registration Module
* Expense Module
* Expense Reporting

**Module Description**

**Module Name:** Registration

**Purpose**

The purpose of Smart Expense Ledger conduct fields Group Details, Adding members to group, member details, group member details preparation and reports.

**Response Sequence**

The Expense Ledger it responsible for this all detail of college student

**Sub Module Name:** Group Details

**Purpose**

The purpose of group details adding the group id and group name using this table content

**Response Sequence**

The Expense Ledger it responsible for this all detail of college student

**Input**

We can add the group information

* Group Id
* Group Name

**Output**

Click the Add button it stored the document.

**Sub module Name:** List of group Members

**Purpose**

The purpose of List of group Members many useful for the group of members and friends easy to handling this way.

**Response Sequence**

The group as responsible for this expense ledger

**Entry Criteria**

The adding group members details is stored in the database

**Input**

* Member id
* Member Name
* First Name
* Last Name
* Email id
* Password
* Phone No

**Output**

Click add button it stored the database

**Sub Module Name:** Members

**Purpose**

The purpose of the members details stored on the adding group members.

**Response Sequence**

The member details responsible for this group

**Entry Criteria**

The member details stored in the database

**Input**

* Member Id
* First Name
* Last Name
* Email Id
* Password
* Phone no

**Output**

Click the Add the button it stored the database

**Module Name:** Expense Details

**Purpose**

The purpose of the expense more people spend the time food, trip, petrol many way this way is expense ledger stored in easy way in the life . This expense way of calculation is very easy and useful for more people.

**Response Sequence**

The expense is responsible for this group member

**Entry** **Criteria**

The expense details stored on the database

**Input**

* Expense id
* Expense Description
* Expense Date
* Expense Amount
* List of Sharing

**Output**

Click the Add button stored in the database

**Module Name:** Expense Sharing

**Purpose**

The settled out is used in the group of members spend expense is stored on the online expense stored in all member but particular friends not spend but one expense food all friends send expense after in the expense this way is used in the settled out process line used credit, debit and balance full details stored on the expense automatically.

**Response Sequence**

The settled out details stored on the expense members is responsible for the group of expense process

**Entry Criteria**

The settled out details is stored on the database

**Input**

* Member Id
* Expense id
* Expense Percentage

**Output**

Click the button stored on the full details of settled out database

**4.2 User Requirement Specification**

Login

Primary Objects

Smart Expense Ledger used to maintain a Group details, List of members details, Expense details, Amount Sharing details.

The Admin login into the system is entering the page.

Secondary Object

The Admin will be entering a correct Mail Id, Password.

* Mail Id
* Password

1. **Group Details**

Primary Objects

We will maintain Group information on this page.

Secondary Objects

Add

We can add this Group information.

* Group id
* Group Name

Update

We can easily modify the information.

Delete

We can delete the Group information.

View

We can easily view on the Group information.

1. **List of Members Details**

Primary Objects

We will maintain a List of Members Details on this page.

Secondary Objects

Add

We can add the List of Members Full information.

* Member id
* Member Name
* First Name
* Last Name
* Email id
* Password
* Phone No

Update

We can easily modify the List of Member information.

Delete

We can delete the List of member full information.

View

We can easily view on the Members information.

1. **Expense Details**

Primary Objects

We will maintain a Expense details on this page.

Secondary Objects

Add

We can add the Expense full information.

* Expense id
* Expense description
* Expense date
* Expense Amount
* List of sharing

Update

We can easily modify the information.

Delete

We can delete the Expense full information.

View

We can easily view on this Expense full information.

1. **Amount Sharing**

Primary Objects

We will maintain Amount Sharing information.

Secondary Objects

Add

We can add the Amount Sharing information.

* Member id
* Expense id
* Expense Percentage

Update

We can easily modify the information.

Delete

We can delete the Amount Sharing information

View

We can easily view on this Amount Sharing full information.

**4.3 Non Functional Requirements**

|  |  |
| --- | --- |
| **Category** | **Subcategory** |
| **Usability** | **Accessibility -**This application support all Android Mobiles. |
| **Reliability** | **Availability -**This application available in 24/7. |
| **Robustness -** For every invalid data from the user, the error message will be displayed and show the correct format data to enter. |
| **Accuracy -**System shall provide the correct results. |
| **Safety -**required to take the database backup. |
| **Performance** | **Throughput -** The database shall be able to accommodate a minimum of 1,000 records of student payment details. |
| **Response Time –**This application take average response time is less than 10 seconds. |
| **Startup/Shutdown time -**This application take an operation within 1 second to server startup. |
| **Capacity -** This application is simultaneously multiple user can use at a time. |
| **Supportability** | **Maintainability -**Everyday trace critical errors to contain error log. |
| **Scalability -** Generally the system can adjust 100 simultaneously users. |
| **Portability -** In future can change the operating system shall not need to rewrite the application logic. |
| **Interface Requirement** | **User Interface -**Refer screenshot |
| **Hardware Interface -**N/A |
| **SUPP40 Software Interface**  1.Database : SQL  2.Programming tool:  Visual Studio  3.Windows versions: Windows 10 |
| **Communication Interface -**N/A |

**4.4 External Interface Requirements**

**Hardware Interface**

* System : Intel (R) Pentium(R) CPU A1018 @ 2.10GHz
* Hard Disk : 1 TB
* Ram : 2 GB (1.90 GB usable)

**Software Interface**

* Operating System : Windows 10
* Coding Language : ASP.NET
* IDE : Visual studio
* Back End : MYSQL
* Server : XAMPP

**5.SYSTEM DESIGN**

**5.1 Table Structure and its relation**

A table is a collection of related of data held in a table format within a database. It consists of columns and rows. Besides the actual data rows, tables generally have associated with them some metadata, such as constraints on the table or on the values within particular columns.

**Table Name:** Group

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **Field Name** | **Type** | **Constraints** |
| 1 | Group Id | Number(5) | Primary Key |
| 2 | Group Name | Varchar2(50) | Not Null |

**Table Name:** List of Group Members Details

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **Field Name** | **Type** | **Constraints** |
| 1 | Group id | Number(5) | Primary Key |
| 2 | Member id | Number(5) | Not Null |
| 3 | Member Name | Varchar2(50) | Not Null |
| 4 | Mail id | Varchar2(30) | Not Null |
| 5 | Password | Varchar2(8) | Not Null |
| 6 | First Name | Varchar2(50) | Not Null |
| 7 | Last Name | Varchar2(50) | Not Null |
| 8 | Phone No | Number(10) | Not Null |

**Table Name:** Expense Details

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **Field Name** | **Type** | **Constraints** |
| 1 | Expense Id | Number(5) | Primary Key |
| 2 | Expense Description | Number(5) | Not Null |
| 3 | Expense Date | Varchar2(10) | Not Null |
| 4 | Expense Amount | Number(7,2) | Not Null |

**Table Name:** Expense Sharing Details

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **Field Name** | **Type** | **Constraints** |
| 1 | Member id | Number(5) | Not Null |
| 2 | Expense id | Number(5) | Not Null |
| 3 | Expense Percentage | Number(7,2) | Not Null |

**5.2 Architecture Design**



5.2 ER Diagram



5.3 Data Flow Diagram



5.4 Use Case Diagram



5.5 Class Diagram



5.6 Process Flow Diagram



5.7 User Interface Design

**6. IMPLEMENTATION (TO DO)**

6.1 Sample coding

6.2 I/O Screen Shot

**7. TESTING**

7.1 Testing Strategies and selecting the suitable strategy for the Project.

7.2 Test Cases [highlighting the Purpose, Required Output, Expected Result].

**8. CONCLUSION**

**9. BIBILOGRAPHY**

* <https://www.youtube.com/watch?v=I0D3JIYFljE>
* <https://www.youtube.com/watch?v=vcjj5pT0bSQ>