SOFTWARE REQUIREMENTS SPECIFICATION

**For**

**BUDGET EXPENSE TRACKER**

**Prepared by:-**

*HARISH M*

*AATHITHYA R*

*ABISHEK K*

# Introduction

## Purpose

The main objective of this document is to illustrate the requirements of the project Budget expense tracker. The document gives the detailed description of the both functional and non-functional requirements proposed by the client .The purpose of this project is to provide a friendly environment to maintain the budget and expense of the client .This project describes the hardware and software interface requirements using ER diagrams and UML diagrams.

## Document Conventions

* + - Entire document should be justified.
    - Convention for Main title

Font face: Times New Roman Font style: Bold

Font Size: 14

* + - Convention for Sub title

Font face: Times New Roman Font style: Bold

Font Size: 12

* + - Convention for body

Font face: Times New Roman Font Size: 12

## Scope of Development Project

The main objective of this project is to make the tracking and monitering of budget and expenses of

A person. Modern men always find different and new way’s to find a way to make money but are very poor in tracking money that they earn. So the main motive is to help the people track the money they earn and help them save the money. For this project, we are using java for frontend development and DBMS for backend development.

## Definitions, Acronyms and Abbreviations

JAVA -> platform independence SQL-> Structured query Language ER-> Entity Relationship

UML -> Unified Modeling Language

IDE-> Integrated Development Environment SRS-> Software Requirement Specification

## References

* + - Books

 Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices (ACM Press) by Michael Jackson

Software Requirements (Microsoft) Second EditionBy Karl E. Wiegers

Software Engineering: A Practitioner’s Approach Fifth Edition By Roger S. Pressman

* + - Websites

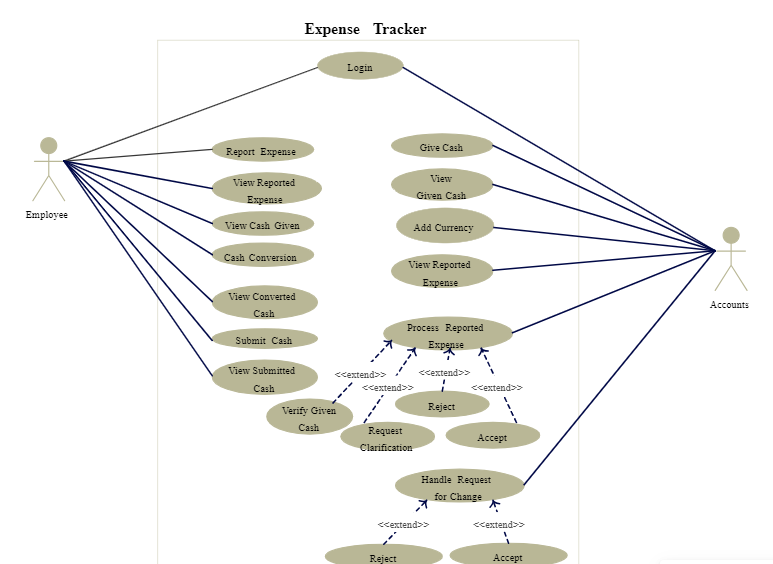
[**http://www.slideshare.net/**](http://www.slideshare.net/)

[**http://ebookily.net/doc/srs-budget-expense-tracker**](http://ebookily.net/doc/srs-budget-expense-tracker)

# Overall Descriptions

## Product Perspective

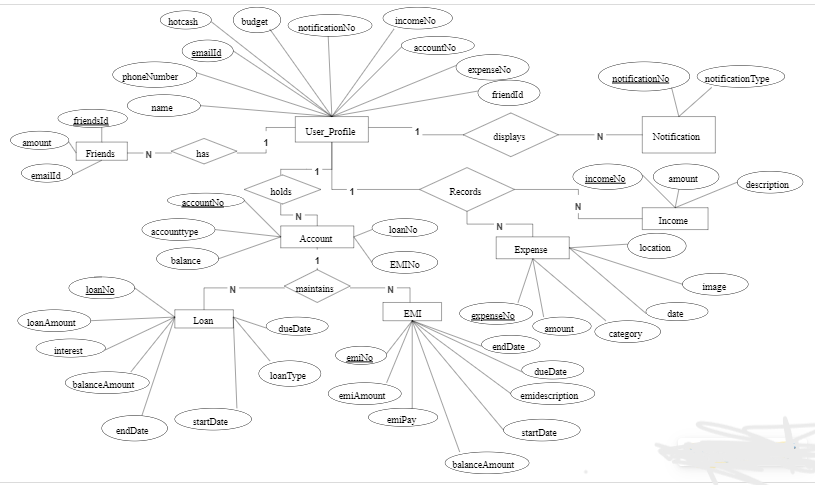
Use Case Diagram Budget expense tracker.



This is a broad level diagram of the project showing a basic overview. The users can be anyone who is earning. This System will help you to track your expense . This algorithm will show you the best ways on how to save your money and when to save.

## Product Function

Entity Relationship Diagram of Budget expense tracker



The main objective of this project is to make the tracking and monitering of budget and expenses of

A person. Modern men always find different and new way’s to find a way to make money but are very poor in tracking money that they earn. So the main motive is to help the people track the money they earn and help them save the money. For this project, we are using java for frontend development and DBMS for backend development.

## User Classes and Characteristics

The features that are available to the Administrtators are:-

* + - Has full access to all features and functionalities.
    - Can manage user accounts.
    - May have additional privileges, such as generating system-wide reports.

The features that are available to the Members are:-

* + - Access to personal budget and expense tracker features.
    - Input , edit and delete personal budget limits and categories
    - Can Set up personal budget limits and categories.

## Operating Environment

The product will be operating in windows environment. The budget expense tracker is a website and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer,Google Chrome,and Mozilla Firefox.Also it will be compatible with the IE 6.0. Most of the features will be compatible with the Mozilla Firefox & Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection.

The hardware configuration include Hard Disk: 40 GB, Monitor: 15” Color monitor, Keyboard: 122 keys. The basic input devices required are keyboard, mouse and output devices are monitor, printer etc.

## Assumptions and Dependencies

The assumptions are:-

* + - The coding should be error free
    - The system should be user-friendly so that it is easy to use for the users
    - The system should have more storage capacity and provide fast access to the database
    - Users may access from any computer that has Internet browsing capabilities and an

Internet connection

* + - Users must have their correct usernames and passwords to enter into their online accounts and do actions

The dependencies are:-

* + - The specific hardware and software due to which the product will be run
    - On the basis of listing requirements and specification the project will be developed and run
    - The end users (admin) should have proper understanding of the product
    - The system should have the general report stored
    - The information of all the users must be stored in a database that is accessible by the tracking system

## Requirement

Software Configuration:-

This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database.

Operating System: Windows NT, windows 98, Windows XP Language: Java Runtime Environment, Net beans 7.0.1 (front end) Database: MS SQL Server (back end)

Hardware Configuration:- Processor: Pentium(R)Dual-core CPU Hard Disk: 40GB

RAM: 256 MB or more

## Data Requirement

The inputs consist of the query to the database and the output consists of the solutions for the query. The output also includes the user receiving the details of their accounts. In this project the inputs will be the queries as fired by the users like create an account, selecting books and putting into account. Now the output will be visible when the user requests the server to get details of their account in the form of time, date and which books are currently in the account.

# External Interface Requirement

## GUI

T Login/Registration Screen:

Fields for username and password.

"Forgot Password" link (if applicable).

Option to register a new account.

Dashboard:

Overview of budget status (current balance, remaining budget).

Recent transactions or a summary of expenses.

Quick links to different sections (expenses, income, reports).

Expense Entry Form:

Fields for expense amount, date, and category.

Dropdown or selection for expense categories.

Additional fields for notes or attachments (e.g., receipts).

Income Entry Form:

Fields for income source, amount, and date.

Additional fields for notes or attachments.

Budget Management:

Section to set up and manage monthly or yearly budgets.

Categorize budgets and set limits.

Reports and Analytics:

Visual representation of spending patterns (charts, graphs).

Options to filter and customize reports.

Expense Categories Management:

List of predefined and user-defined expense categories.

Option to add, edit, or delete categories.

Reminders and Notifications:

Section to set up reminders for upcoming bills or expenses.

Notifications for budget warnings.

Settings:

User preferences (currency, language).

Security settings (change password, enable two-factor authentication).

History and Audit Trail:

Log of past transactions and changes.

Option to filter and search transaction history.

User Profile:

View and edit user details.

Profile picture upload (optional).

Help and Support:

FAQ section.

Contact support or provide feedback.

Offline Mode Indicator:

Clear indication of the application's offline or synchronization status.

Logout Button:

Clearly visible button to log out.

Responsive Design:

Ensure the GUI is responsive for different screen sizes (desktop, tablet, mobile).

Accessibility Features:

Consider accessibility features for users with disabilities.

Color Scheme and Theming:

Use a visually appealing color scheme.

Provide options for different themes.

Dashboard Widgets (Optional):

Customizable widgets for quick access to important information.

# Other Non-functional Requirements

## Performance Requirement

The proposed system that we are going to develop will be used as the Chief performance system within the different campuses of the university which interacts with the university staff and students. Therefore, it is expected that the database would perform functionally all the requirements that are specified by the organization.

## Safety Requirement

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost. Proper UPS/inverter facility should be there in case of power supply failure.

## Security Requirement

* + - System will use secured database
    - Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
    - System will have different types of users and every user has access constraints
    - Proper user authentication should be provided
    - No one should be able to hack users’ password
    - There should be separate accounts for admin and members such that no member can access the database and only admin has the rights to update the database.

## Requirement attributes

* + - There may be multiple admins creating the project, all of them will have the right to create changes to the system. But the members or other users cannot do changes
    - The project should be open source
    - The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database
    - The user be able to easily download and install the system

## Business Rules

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data.This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols. Neither admin nor member should cross the rules and regulations.

## User Requirement

The users of the system are members and Librarian of the university who act as administrator to maintain the system. The members are assumed to have basic knowledge of the computers and internet browsing. The administrators of the system should have more knowledge of the internals of the system and is able to rectify the small problems that may arise due to disk crashes, power failures and other catastrophes to maintain the system. The proper user interface, user manual, online help and the guide to install and maintain the system must be sufficient to educate the users on how to use the system without any problems.

The admin provides certain facilities to the users in the form of:-

* + - Backup and Recovery
    - Forgot Password
    - Data migration i.e. whenever user registers for the first time then the data is stored in the server
    - Data replication i.e. if the data is lost in one branch, it is still stored with the server
    - Auto Recovery i.e. frequently auto saving the information
    - Maintaining files i.e. File Organization
    - The server must be maintained regularly and it has to be updated from time to time

# Other Requirements

## Data and Category Requirement

There are different categories of users namely teaching staff, Librarian, Admin, students etc. Depending upon the category of user the access rights are decided.It means if the user is an administrator then he can be able to modify the data,delete, append etc. All other users except the Librarian only have the rights to retrieve the information about database. Similarly there will be different categories of books available. According to the categories of books their relevant data should be displayed. The categories and the data related to each category should be coded in the particular format.

## Appendix

A: Admin, Abbreviation, Acronym, Assumptions; B: Books, Business rules; C: Class, Client, Conventions; D: Data requirement, Dependencies; G: GUI; K: Key; L: Library, Librarian; M: Member; N: Non-functional Requirement; O: Operating environment; P: Performance,Perspective,Purpose; R: Requirement, Requirement attributes; S: Safety, Scope, Security, System features; U: User, User class and characteristics, User requirement;

## Glossary

The following are the list of conventions and acronyms used in this document and the project as well:

* + - Administrator: A login id representing a user with user administration privileges to the software
    - User: A general login id assigned to most users
    - Client: Intended users for the software
    - SQL: Structured Query Language; used to retrieve information from a database
    - SQL Server: A server used to store data in an organized format
    - Layer: Represents a section of the project
    - User Interface Layer: The section of the assignment referring to what the user interacts with directly
    - Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed
    - Data Storage Layer: The section of the assignment referring to where all data is recorded
    - Use Case: A broad level diagram of the project showing a basic overview
    - Class diagram: It is a type of static structure diagram that describes the structure of a system by showing the system’s cases, their attributes, and the relationships between the classes
    - Interface: Something used to communicate across different mediums
    - Unique Key: Used to differentiate entries in a database

## Class Diagram

A class is an abstract, user-defined description of a type of data. It identifies the attributes of the data and the operations that can be performed on instances (i.e. objects) of the data. A class of data has a name, a set of attributes that describes its characteristics, and a set of operations that can be performed on the objects of that class. The classes’ structure and their relationships to each other frozen in time represent the static model. In this project there are certain main classes

which are related to other classes required for their working. There are different kinds of relationships between the classes as shown in the diagram like normal association, aggregation, and generalization. The relationships are depicted using a role name and multiplicities.

