Software Requirements Specification

for

Daily Expense Tracker

Version 1.0 approved

Prepared by
Muskan Khasturia 1902073
Chirag Khatri 1902076
Isha Khatri 1902077
Jenil Kothari 1902081

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

We are developing an Web application named as “Daily Expense Tracker System” and this application is used to manage the application user‘s daily expenses in a more efficient and manageable way. By using this application we can reduce the manual calculations for their daily expenses and keep the track of the expenditure. In this application, user can provide his/her expense to calculate his/her total expenses per day and these results will stored for unique user.

1.2 Document Conventions

Important points have been underlined to provide emphasis. Headings and Subheadings have been written in bold font to provide emphasis. The points in all sections have been written in the order of their priority, from higher priority points to lower priority points, so that important points are not missed out. Abbreviations are used in some places which will be understood by the developers of the application

1.3 Intended Audience and Reading Suggestions

The intended audience is the team of developers who will be designing and implementing the Daily Expense Tracker System. Also, the document is to be utilized by the testing team who will be testing and evaluating the performance and design of the application. The document consists of all the necessary information that will be required by the team of software engineers who will be working on the project**.**

1.4 Product Scope

This application can take a good market as it is usable by anyone who are willing to manage their expenses and aiming to save for the future investments and many more. There is not any range criteria or any kind of profession or gender are focused, it be will used hugely.

1.5 References

PHP : https://www.php.net/docs.php

Bootstrap Framework: https://getbootstrap.com/Document

MySQL : https://dev.mysql.com/doc/

JavaScript : https://developer.mozilla.org/en-US/docs/Web/JavaScript

2. Overall Description

2.1 Product Perspective

Daily Expense Tracker System (DETS) aims to help everyone who are planning to know their expenses and save from it.
We are developing an Web application named as “Daily Expense Tracker System” and this application is used to manage the application user‘s daily expenses in a more efficient and manageable way.
In this application, user can provide his/her expense to calculate his/her total expenses per day and these results will stored for unique use.

2.2 Product Functions

1. **Dashboard:** In this section, user can briefly view expenses on a daily basis, monthly basis and yearly basis.
2. **Expenses:** In this section user can manage the expenses(add/delete).
3. **Expense Report**: In this section, user can view expenses on day wise basis, month wise basis and year wise basis according to periods of time.
4. **Profile**: In this section, user can update his/her profile.
5. **Change Password**: In this section, user can change his/her passwords
6. **Logout**: Through this button, user can log out.

2.3 User Classes and Characteristics

Daily Expense Tracker has only one module that is User. In this web application user has to register so they can use all the functionalities of the system . While using the application , User can manage their profile and make changes required. User can also add and manage expenses. With the help of this functionality, User can generate daily , monthly and yearly reports . There is also an option to generate Category Wise reports.

2.4 Operating Environment

Since the application is a web application it can work on any device having a browser. Devices: Mobile Phone, Computer, Laptops, Tablets. Operating System: Windows, Linux distributions, MacOS, Android, IOS RAM: 128 MB or more Disk Space: 20 MB or more. Browsers: Mozilla Firefox 30+, Google Chrome 27.0+, Microsoft Edge. Other browsers can also be used. Internet connection: Strong internet connection with speed of at least 3 Mbps for best experience.

2.5 Design Implementation Concepts

CO-1: The time allotted for this project is at most 3 months.
CO-2: The front end of the application will be made using HTML, CSS, Bootstrap and JavaScript.
CO-3: PHP will be used as the language for the backend of the application and SQL will be used for the database of the application. Xampp was used as the server.
CO-4: The website will be in English language. Users who do not know English will have to change the language of the websites.

2.6 User Documentation

Appropriate instructions will be provided at every step in the application to ensure the users do not face any difficulties while using the application. In future, we can add a review system.

2.7 Assumptions and Dependencies

AS-1: The application supports only English language. We assume the users of the application will be well versed with English.

AS-2: The users of the application should have basic knowledge of using web applications.

DE-1: The application will require SQL as dependency, since we use PHP as the backend language.

DE-2: Bootstrap Framework will be used for the front end of the application.

DE-3: For dates and piecharts we have used datepicker and easypiechart.

3. External Interface Requirements

3.1 User Interfaces

UI-1:
The website will start with a landing page. The landing page will have all information about the web application.
UI-2:
Their will be a sidebar on left hand side which contains Dashboard,Expenses,Expense Report,Profile,Change Password and Log out.
UI-3:
There will be alerts and pop ups which appear in case the user makes a mistake while using the application.
UI-4:
The interface will be responsive for all screen sizes as much as possible to provide the users a seamless experience.
UI-5:
User can briefly view expenses on day wise basis,month wise basis and yearly basis on dasboard section.

3.2 Hardware Interfaces

The web app will be compatible on almost any web browser without any need of too much RAM. The app is easy to use and doesn't require too many high end specifications.

3.3 Software Interfaces

- Browsers: Mozilla, Google Chrome, IE8, OPERA
- Operating System: Android, Windows 7, 8, 10, Mac OS, Linux distributions.

3.4 Communications Interfaces

- The application will be using HTTPS protocol.
- We will be storing user emails to update them with latest Updates and improved security patches.

4. System Features

4.1 Authentication and Authorization

4.1.1 Description and Priority

The application will be having multiple users and so authentication becomes a high priority system feature. The application will be using PHP and MySQL for authentication functionality.In Daily Expense Tracker System we use PHP and MySQL database. This is the project which keeps records of daily expenses. DETS has one module i.e. user. When the user creates a new account on the application, they will have to provide their email address and password. The password must be at least 8 characters long. The passwords in the system will be hashed and stored so that no other person can get to know the password.

4.1.2 Stimulus/Response Sequences

Once the user registers in the application, they will be guided to a login page where they will have to enter their email address and password to login. After successful login, the user will be redirected to the landing page of the application. There will also be a logout button on the navigation bar. On clicking the logout button, the user will be logged out.

4.2 Add / Delete Expenses

4.2.1 Description and Priority

Feature to add or delete expenses on the website will be provided to the user.

The User will have to register first and then login after that only user will be redirected to dashboard where user can add / delete expenses. The feature is of high priority as the application is based on adding /deleting expenses.

4.2.2 Response Sequence

The user will have to provide all the necessary information about the Item purchased such as Date of expense, Item Purchased and Cost of Item. On filling out all the information the user will be get the pop-up as expense added and on the Manage Expense page user can simply Delete the Expense by clicking on delete in Action column and then pop-up will be displayed as Record has been Deleted.

4.2.3 Functional Requirements

In order to display the table of Expenses added SQL will be used.

4.3 View Expenses

4.3.1 Description and Priority

The user gets to View all the expenses which he might need to calculate his budget. This is again a high priority feature. The user gets the option to View Expenses on Daywise, Monthwise, Yearwise basis using the sidepanel by selecting Expense Report**.**

4.3.2 Response Sequence

The user gets the option to View Expenses on Daywise, Monthwise, Yearwise basis using the sidepanel**.** To View Expenses on different basis user have to provide Start Date and End Date and after that user will be able to see daywise expenses here user can see expense of every day within the range user has provided along with the Expense Amount of each day. In monthwise expense report user can see the total amount he has spend in the given month. In yearwise expense report user will get the facility to see the total expense of the entire year.

4.3.3 Functional Requirements

To View Expenses SQL will be used.

4.4 Update Profile

4.4.1 Description and Priority

The user gets to update the profile. The user will simply have to click on the Profile section on the sidepanel. The priority of the feature is moderate.

4.4.2 Response Sequence

In this section, user can change Name and Mobile no by just clicking on update button. On performing the necessary changes they are successfully added to the database and the user will get the message as Profile Updated.

4.4.3 Functional Requirements

To Update Profile SQL will be used.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Scalability: The application should be scalable and should perform without any interruption for all the users.

5.2 Safety Requirements

• Backup power supply should be present for server, so that it does not stop functioning in case of power failure.

5.3 Security Requirements •

The passwords of the users are hashed and then stored in the database so that no person can access the passwords of the users.

• The passwords should be at least 8 characters long .

• The website should HTTPS protocol for security.

• User should have unique email.

5.4 Software Quality Attributes

5.4.1 Usability: The user interface should be simple to use and not cluttered with a lot of information .

5.4.2 Availability: The system should be always available. The system should be reliable and there should be no loss of data in case the server breaks down when operations are going on.

5.4.3 Maintainability: The code for the application should be written cleanly and should be well documented. The code should contain comments to help new programmers and developers make changes in the application.

5.4.4 Testability: The code should be written with proper test cases to be tested upon so that no errors during production take place.

5.5 Business Rules

The administrator of the application has full permission of controlling the system.

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>