

Project Report On
Daily Expense Tracker



By

Janmejaya Bai (18CSE051)

Ankit Tripathy (18CSE043)

Jasmin Patnaik (18CSE181)

In partial fulfillment of requirements for the award of the degree

Bachelor of Technology (Computer Science)

(2020)



Janmejaya Bai

Department of Computer Science

Certificate

This is to certify that Janmejaya Bai (18CSE051) and His/Her Friend, student of Bachelor of Technology, Fifth Semester, Department of Computer Science of GIET University, has pursued the Minor Project titled “Daily Expense Tracker” and the report has been submitted in partial fulfillment of requirements for the award of the degree, Bachelor of Technology in Computer Science in the Year 2021.

**External Examiner
Signature**

**Internal Examiner
Signature**

ACKNOWLEDGEMENT

We express our sincere regard and indebtedness to our teachers, for their valuable time, guidance, encouragement, support and cooperation throughout the duration of our project. We would sincerely like to thank CSE/IT Department for giving the opportunity to work on enhancing our technical skills while undergoing this project. This project helped in understanding the various parameters which are involved in the development of a web application and the working and integration of front end along with the back end to create a fully functional web application.

We would like to thank Dr. Sanjay K Kuanar, Head of Department and whole of department for their constant support.

Janmejaya Bai (18CSE051)

Ankit Tripathy (18CSE043)

Jasmin Patnaik (18CSE181)

ABSTRACT

In today's busy and expensive life we are in a great rush to make money. But at the end of the month we broke off. As we are unknowingly spending money on little and unwanted things. So, we have come over with the idea to track our earnings. Daily Expense Tracker (DET) aims to help everyone who are planning to know their expenses and save from it. DET is an website which users can execute in their mobile phones and update their daily expenses so that they are well known to their expenses. Here user can define their own categories for expense type like food, clothing, rent and bills where they have to enter the money that has been spent and also can add some information in additional information to specify the expense. User can also define expense categories. User will be able to see pie chart of expense. Although this website is focused on new job holders, interns and teenagers, everyone who wants to track their expense can use this app.

Every day Expense Tracker System is intended to monitor Income-Expense of a Housewife on an everyday premise. This System separates the Income-based daily expenses. On the off chance that you surpass day's cost, the system will cut it from your salary and will give new everyday cost permitted sum. On the off chance that that day's cost is less, the system will include it in reserve funds. Day by day cost following System will produce report toward the finish of month to demonstrate Income-Expense Curve. It will give you a chance to include the reserve funds sum, which you had put something aside for some specific Festivals or days like Birthday or Anniversary.

INDEX

S.No.	Index	Page No.
Chapter 1	INTRODUCTION	1-5
1.1	Introduction	1
1.2	Aim	1
1.3	Existing System	1
1.4	Proposed System	2
1.5	Feasibility Study	2-4
1.6	Project Work Schedule	4
1.7	Organisation of Report	5
Chapter 2	SOFTWARE REQUIREMENTS SPECIFICATION	6
2.1	Hardware Requirement	6
2.2	Software Requirement	6
Chapter 3	DESIGN & PLANNING	7-12
3.1	Software Development Life Cycle Model	7
3.2	GENERAL OVERVIEW	8
3.3	Flow Chart	8
3.4	ER Diagram	9
3.5	DFD Diagram	9-10
3.6	Use case diagram	11
3.7	Algorithm used	12
3.8	Screenshots	12
Chapter 4	IMPLEMENTATION DETAILS	13-25
4.1	FRONT END	13-14
4.2	BACK END	15-16
4.3	CODING	17-25

Chapter 5	TESTING AND IMPLEMENTATION	26-27
5.1	UNIT TESTING	26
5.2	INTEGRATION TESTING	27

Chapter 6	ADVANTAGES	28
Chapter 7	CONCLUSION	28
7.1	RECOMMENDATION	28
7.2	FUTURE USE	28
	BIBLIOGRAPHY & REFERENCES	29

CHAPTER 1 : INTRODUCTION

1.1 INTRODUCTION

Daily Expense Tracker System is designed to keep a track of Income-Expense of a Housewife on a day-to-day basis. This System divides the Income based on daily expenses. If you exceed day's expense, system will cut it from your income and will provide new daily expense allowed amount. If that day's expense is less, system will add it in savings. Daily expense tracking System

will generate report at the end of month to show Income-Expense Curve. It will let you add the savings amount, which you had saved for some particular Festivals or days like Birthday or Anniversary. Daily Expense Tracker System is a system which will keep a track of Income Expense of a House-Wife on a day to day basics, This System takes Income from House-Wife and divides in daily expense allowed, If u exceed that days expense it will cut if from your income and give new daily expense allowed Amt, and if that days expense is less it will add it in savings. Daily expense tracking System will generate report at the end of month to show Income-Expense Curve. It will let you add the savings amt which you had saved for some particular Festivals or day like Birthday or Anniversary.

1.2 AIM

Daily Expense Tracker (DET) aims to help everyone who are planning to know their expenses and save from it. DET is an android app which users can execute in their mobile phones and update their daily expenses so that they are well known to their expenses. Here user can define their own categories for expense type like food, clothing, rent and bills where they have to enter the money that has been spent and also can add some information in additional information to specify the expense. User can also define expense categories. User will be able to see pie chart of expense. Also, DET app is capable of clustering. Personal and administration clustering is possible by the use of Apriori algorithm. Although this app is focused on new job holders, interns and teenagers, everyone who wants to track their expense can use this app.

1.3 EXISTING SYSTEM

In existing, we need to maintain the Excel sheets, CSV etc. files for the user daily and monthly expenses. In existing, there is no as such complete solution to keep a track of its daily expenditure easily. To do so a person as to keep a log in a diary or in a computer, also all the calculations needs to be done by the user which may sometimes results in errors leading to losses.

1.4 PROPOSED SYSTEM

To reduce manual calculations, we propose an application which is developed by Android. This application allows users to maintain a digital automated diary. Each user will be required to register on the system at registration time, the user will be provided id, which will be used to maintain the record of each unique user. Expense Tracker application which will keep a track of Income-Expense of a user on a day to day basis. This application takes Income from user and divides in daily expense allowed. If u exceed that days expense it will cut if from your income and give new daily expense allowed amount, and if that days expense is less it will add it in savings.

Expense tracking application will generate report at the end of month to show Income-Expense via multiple graphs. It will let you add the savings amount which you had saved for some particular Festivals or day like Birthday or Anniversary.

1.5 FEASIBILITY STUDY

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it's worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

- 1) Operational Feasibility
- 2) Technical Feasibility
- 3) Economical Feasibility

1.5.1 OPERATIONAL FEASIBILITY

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes. To ensure success,

desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system

may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

1.5.2 TECHNICAL FEASIBILITY

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating in order to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible. The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

1.5.3 ECONOMICAL FEASIBILITY

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

1.6 Giant Chart

Activity	Time Frame					
	08/08/2020	11/08/2020	01/09/2020	01/10/2020	01/11/2020	08/11/2020
	To 10/08/2020	To 31/08/2020	To 29/09/2020	To 30/10/2020	To 07/11/2020	To 18/11/2020
Literature Survey & Planning						
Feature Finalisation With Mockup						
Front End Developments						
BackEnd Development						
Testing & Fault Detection						
Project Report						

1.7 ORGANISATION OF THE REPORT

1.7.1 INTRODUCTION

This section includes the overall view of the project i.e. the basic problem definition and the general overview of the problem which describes the problem in layman terms. It also specifies the software used and the proposed solution strategy.

1.7.2 SOFTWARE REQUIREMENTS SPECIFICATION

This section includes the Software and hardware requirements for the smooth running of the application.

1.7.3 DESIGN & PLANNING

This section consists of the Software Development Life Cycle model. It also contains technical diagrams like the Data Flow Diagram and the Entity Relationship diagram.

1.7.4 IMPLEMENTATION DETAILS

This section describes the different technologies used for the entire development process of the Front-end as well as the Back-end development of the application.

1.7.5 RESULTS AND DISCUSSION

This section has screenshots of all the implementation i.e. user interface and their description.

1.7.6 SUMMARY AND CONCLUSION

This section has screenshots of all the implementation i.e. user interface and their description.

CHAPTER 2 : SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Hardware Requirements

Number	Description
1	PC with 250 GB or more Hard disk.
2	PC with 2 GB RAM.
3	PC with Pentium 1 and Above.

2.2 Software Requirements

Number	Description	Type
1	Operating System	Windows XP / Windows
2	Language	JavaScript, HTML, CSS
3	Database	MySQL
4	IDE	Visual Code
5	Browser	Google Chrome

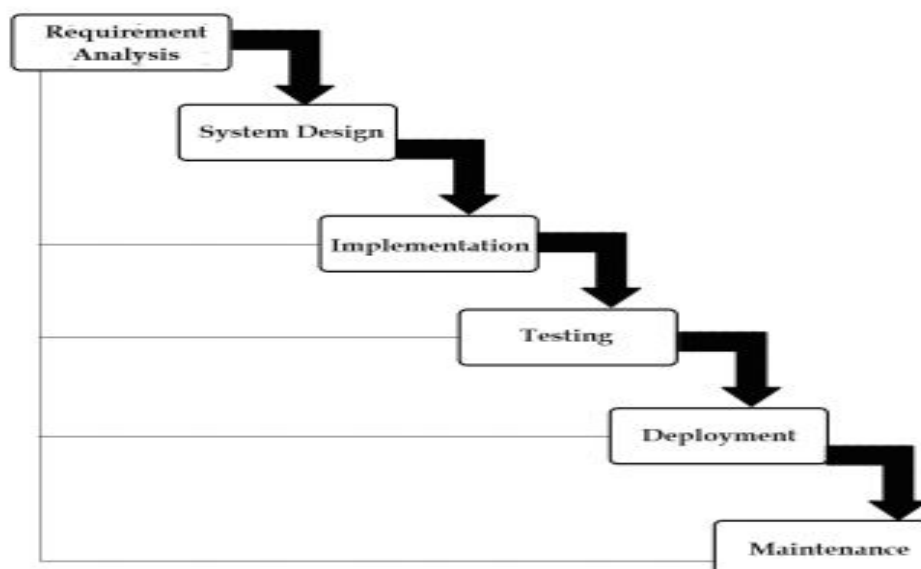
CHAPTER 3 : DESIGN & PLANNING

3.1 Software Development Life Cycle Model

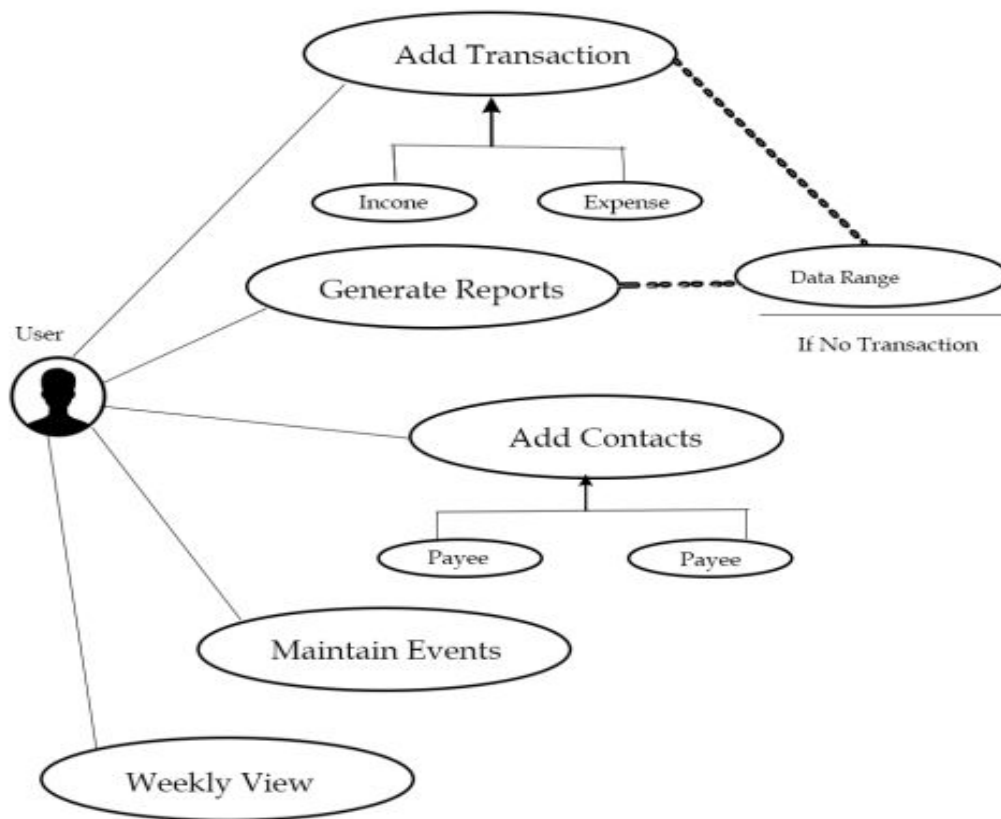
3.1.1 WATERFALL MODEL

The waterfall model was selected as the SDLC model due to the following reasons:

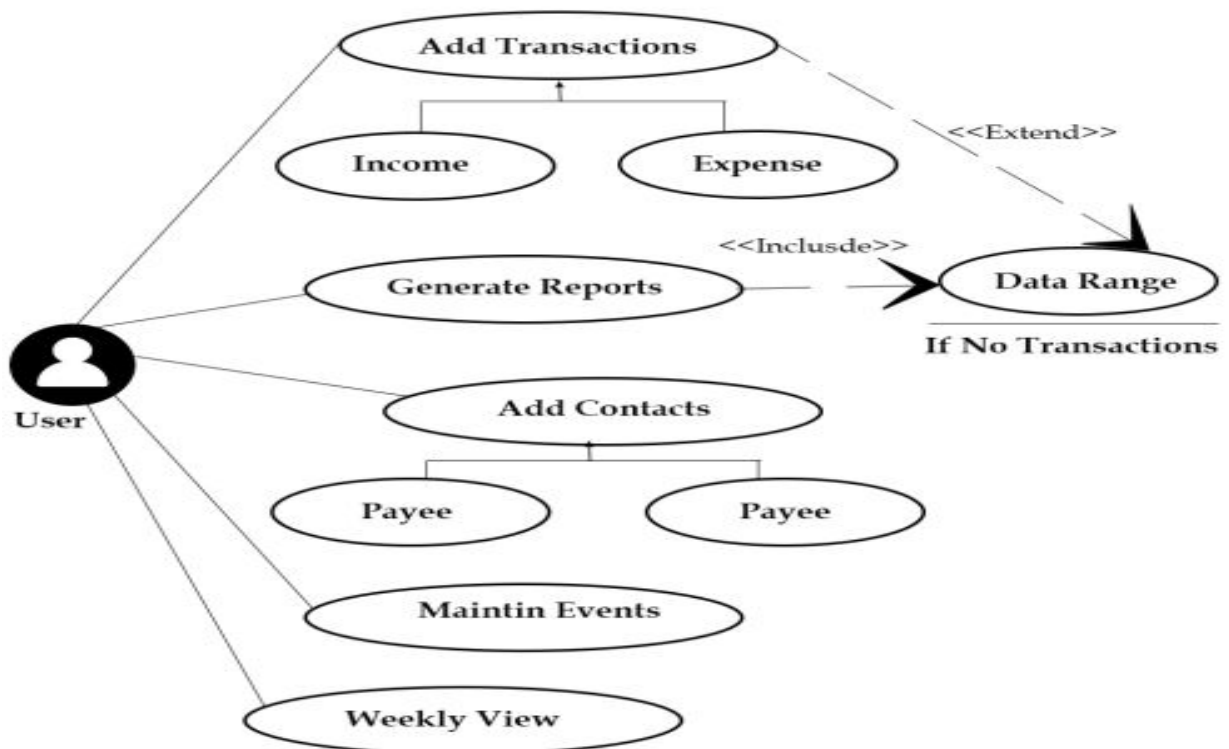
- Requirements were very well documented, clear and fixed.
- Technology was adequately understood.
- Simple and easy to understand and use.
- There were no ambiguous requirements.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.



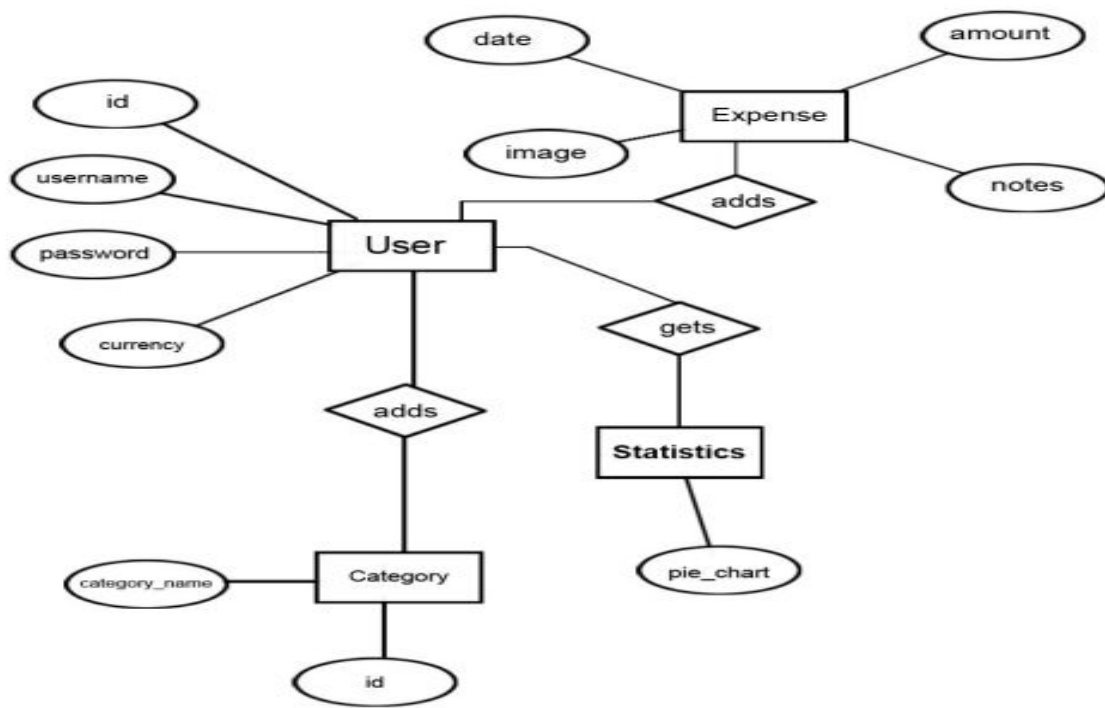
3.2 GENERAL OVERVIEW



3.3 Flow Chart



3.4 ER Diagram

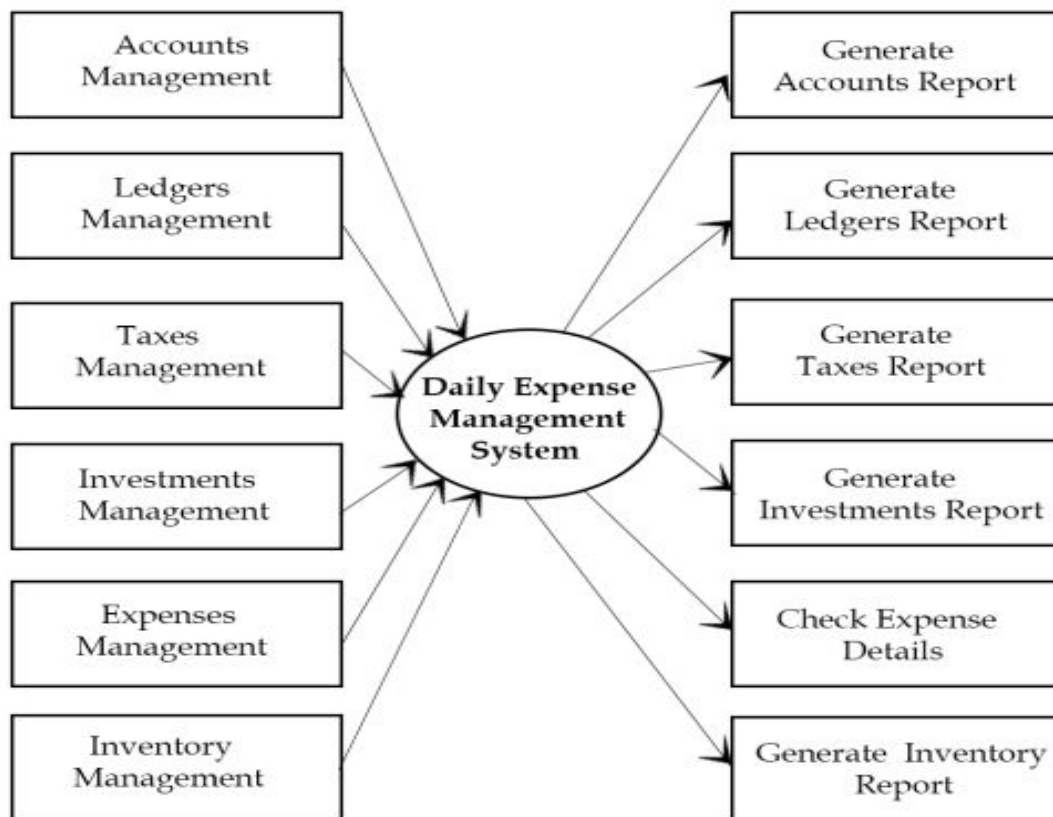


3.5 DFD Diagram

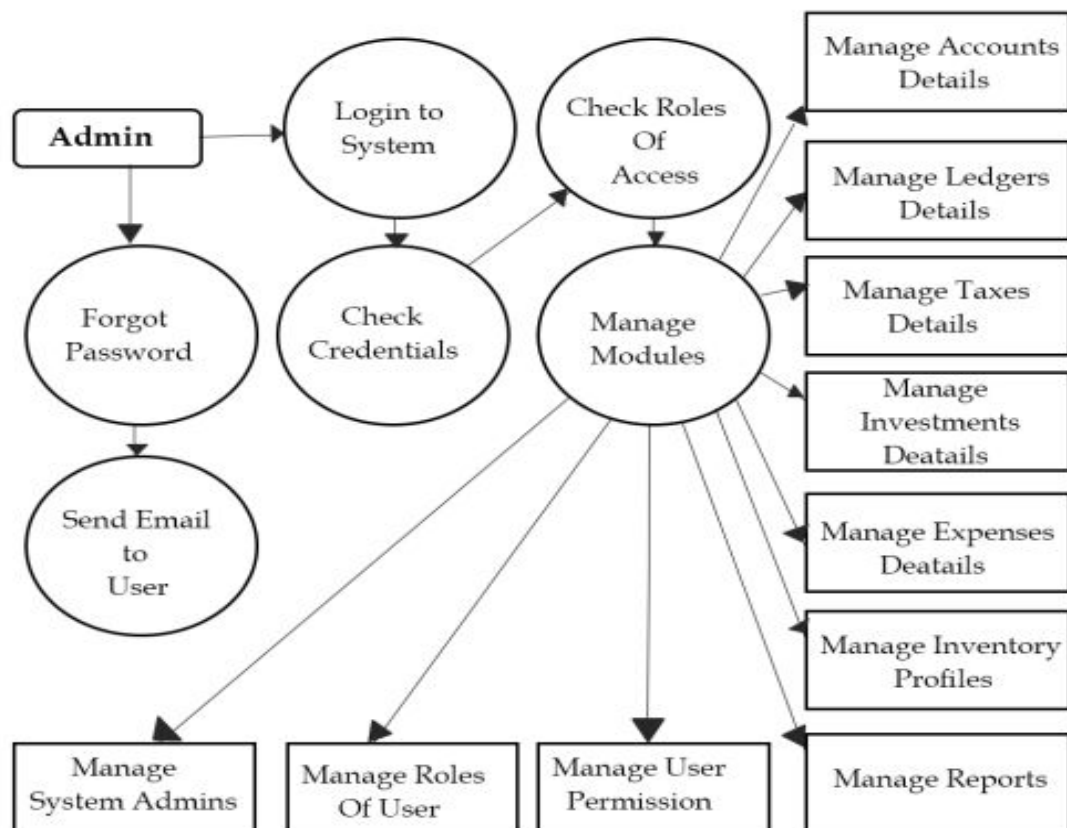
3.5.1 Zero-Level DFD Diagram



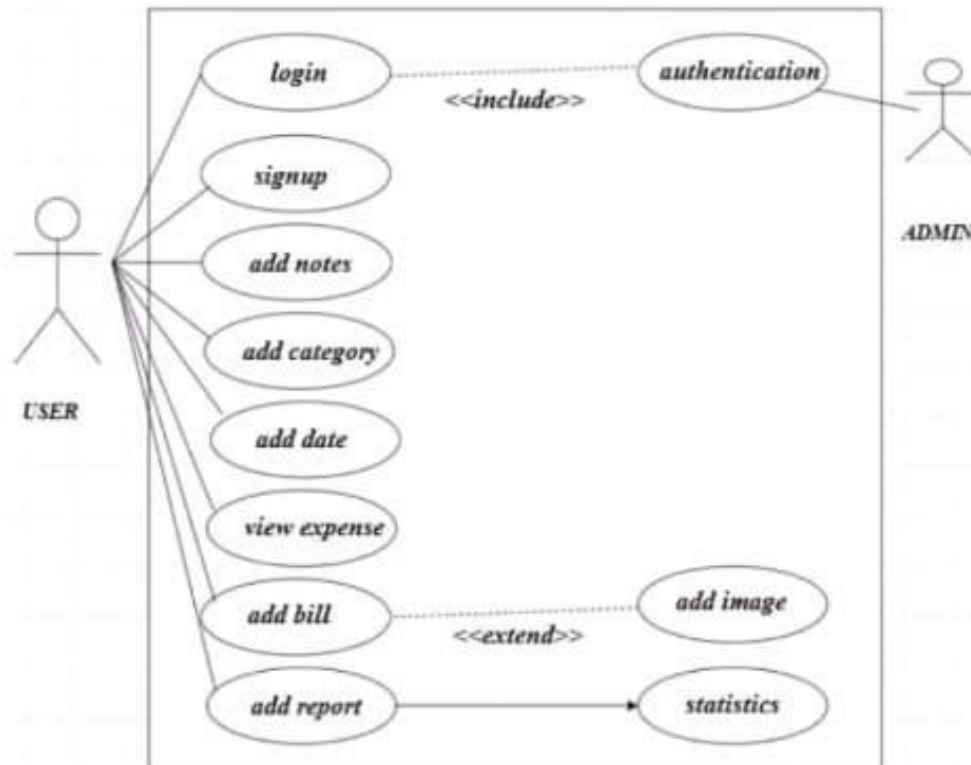
3.5.2 First-Level DFD Diagram



3.5.3 Second-Level DFD Diagram



3.6 Use-case Diagram



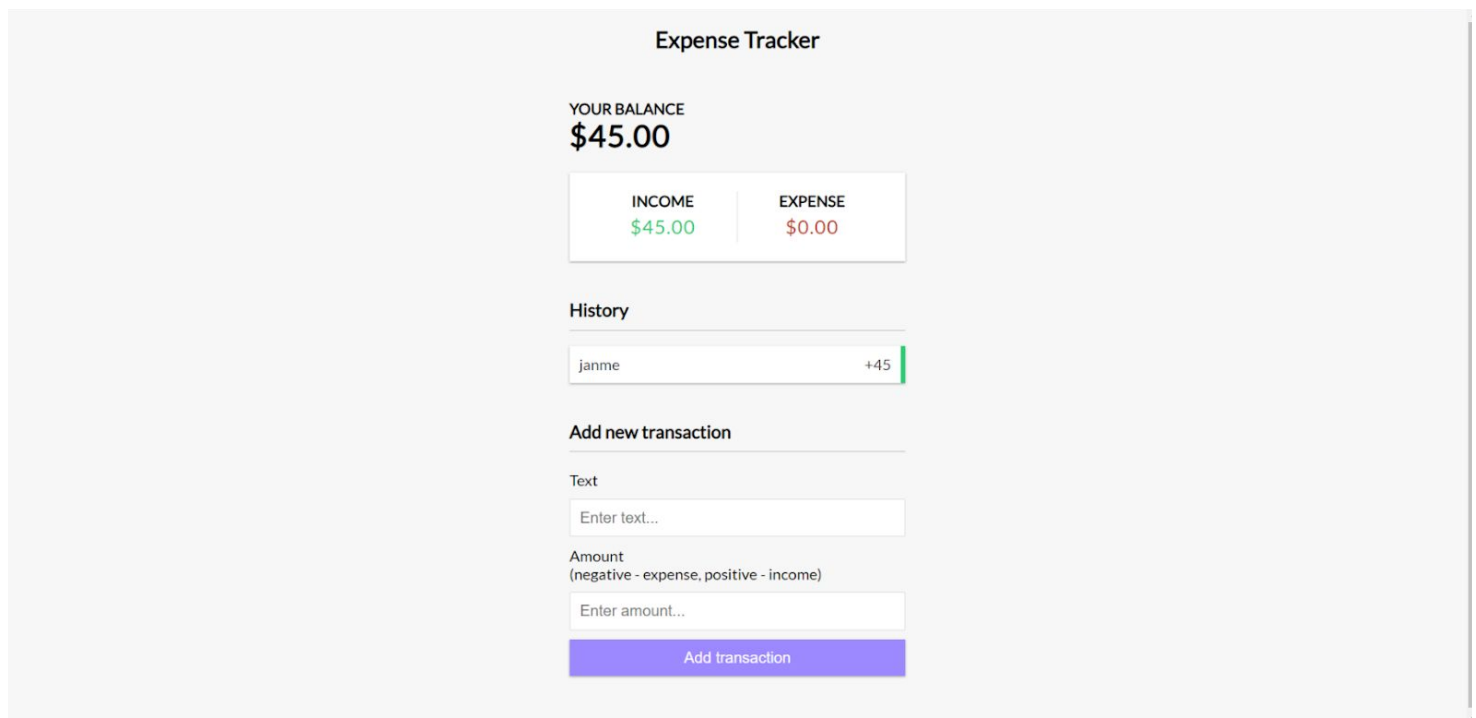
Name	Description
Initializing actor	User
Log-in	To provide identification details to the system.
Authentication	To validate the details provided by the user
View expenses	Provide the updated log od expenses
Add bill	To add new expenses
Track report	Display the report
Add notes	To add further information
Add Date	T add the date of the expenditure

Low level design

3.7 Algorithms used

The Apriori algorithm is an influential algorithm for mining frequent itemsets for boolean association rules. Apriori uses a “bottom up” approach”, where frequent subsets are extended one item at a time (a step known as candidate generation, and groups of candidates are tested against the data). Apriori is designed to operate on database containing transactions ie: collections of items bought by the customer.

3.8 Screen-shot diagram



The screenshot displays a web application titled "Expense Tracker". At the top, it shows "YOUR BALANCE" as "\$45.00". Below this, a summary table lists "INCOME" as "\$45.00" and "EXPENSE" as "\$0.00". A "History" section contains a single entry: "janme" with a value of "+45", accompanied by a green progress bar. The "Add new transaction" section includes a "Text" input field with the placeholder "Enter text...", an "Amount" input field with the placeholder "Enter amount.." and a note "(negative - expense, positive - income)", and a blue "Add transaction" button.

Expense Tracker	
YOUR BALANCE	
\$45.00	
INCOME	EXPENSE
\$45.00	\$0.00
History	
janme	+45
Add new transaction	
Text	
Enter text...	
Amount	
(negative - expense, positive - income)	
Enter amount..	
Add transaction	

CHAPTER 4 : IMPLEMENTATION DETAILS

In this Section we will do Analysis of Technologies to use for implementing the project.

4.1 : FRONT END

4.1.1 HTML

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

4.1.2 JavaScript

JavaScript is a high-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object orientation, and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it, and major web browsers have a dedicated JavaScript engine to execute it. As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

The terms Vanilla JavaScript and Vanilla JS refer to JavaScript not extended by any frameworks or additional libraries. Scripts written in Vanilla JS are plain JavaScript code. Google's Chrome extensions, Opera's extensions, Apple's Safari 5 extensions, Apple's Dashboard Widgets, Microsoft's Gadgets, Yahoo! Widgets, Google Desktop Gadgets, and Serence Klipfolio are implemented using JavaScript.

4.1.3 Css

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

CSS information can be provided from various sources. These sources can be the web browser, the user and the author. The information from the author can be further classified into inline, media type, importance, selector specificity, rule order, inheritance and property definition. CSS style information can be in a separate document or it can be embedded into an HTML document. Multiple style sheets can be imported. Different styles can be applied depending on the output device being used; for example, the screen version can be quite different from the printed version, so that authors can tailor the presentation appropriately for each medium. The style sheet with the highest priority controls the content display. Declarations not set in the highest priority source are passed on to a source of lower priority, such as the user agent style. The process is called cascading.

One of the goals of CSS is to allow users greater control over presentation. Someone who finds red italic headings difficult to read may apply a different style sheet. Depending on the browser and the web site, a user may choose from various style sheets provided by the designers, or may remove all added styles and view the site using the browser's default styling, or may override just the red italic heading style without altering other attributes.

4.2 : BACK END

4.2.1 Java

Java is a general-purpose programming language that is class-based, object-oriented[15] (although not a pure object-oriented language, as it contains primitive types , and designed to have as few implementation dependencies as possible. It is intended to let application developers write once, run anywhere (WORA),meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them. As of 2019, Java was one of the most popular programming languages in use according to GitHub,particularly for client-server web applications, with a reported 9 million developers.

Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GNU General Public License. Meanwhile, others have developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

The latest versions are Java 12, released in March 2019, and Java 11, a currently supported long term support (LTS) version, released on September 25, 2018; Oracle released for the legacy Java 8 LTS the last free public update in January 2019 for commercial use, while it will otherwise still support Java 8 with public updates for personal use up to at least December 2020. Oracle (and others) highly recommend that you uninstall older versions of Java, because of serious risks due to unresolved security issues.Since Java 9 (and 10) is no longer supported, Oracle advises its users to immediately transition to Java 11 (Java 12 is also a non-LTS option).

4.2.2 MySQL

MySQL is an open-source relational database management system (RDBMS) based on Structured Query Language (SQL). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is pretty easy to master in comparison with other database software like Oracle Database, or Microsoft SQL Server. MySQL can run on various platforms UNIX, Linux, Windows, etc. You can install it on a server or even in a desktop. Besides, MySQL is reliable, scalable, and fast. The official way to pronounce MySQL is My Ess Que Ell, not My Sequel. However, you can pronounce it whatever you like, who cares?

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Youtube, Twitter and so on.

4.3 Coding

4.3.1 JavaScript code

```
const balance = document.getElementById('balance');
const money_plus = document.getElementById('money-plus');
const money_minus = document.getElementById('money-minus');
const list = document.getElementById('list');
const form = document.getElementById('form');
const text = document.getElementById('text');
const amount = document.getElementById('amount');

// const dummyTransactions = [
//   { id: 1, text: 'Flower', amount: -20 },
//   { id: 2, text: 'Salary', amount: 300 },
//   { id: 3, text: 'Book', amount: -10 },
//   { id: 4, text: 'Camera', amount: 150 }
// ];

const localStorageTransactions = JSON.parse(
  localStorage.getItem('transactions')
);

let transactions =
  localStorage.getItem('transactions') !== null ? localStorageTransactions : [];

// Add transaction
function addTransaction(e) {
  e.preventDefault();

  if (text.value.trim() === '' || amount.value.trim() === '') {
    alert('Please add a text and amount');
  } else {
    const transaction = {
      id: generateID(),
      text: text.value,
      amount: +amount.value
    };

    transactions.push(transaction);

    addTransactionDOM(transaction);

    updateValues();

    updateLocalStorage();

    text.value = '';
    amount.value = '';
  }
}

// Generate random ID
function generateID() {
  return Math.floor(Math.random() * 1000000000);
}
```

```
// Add transactions to DOM list
function addTransactionDOM(transaction) {
  // Get sign
  const sign = transaction.amount < 0 ? '-' : '+'
```

4.3.2 HTML code

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <meta http-equiv="X-UA-Compatible" content="ie=edge" />

  <link rel="stylesheet" href="style.css" />

  <title>Expense Tracker</title>

</head>

<body>

  <h2>Expense Tracker</h2>

  <div class="container">

    <h4>Your Balance</h4>

    <h1 id="balance">$0.00</h1>

    <div class="inc-exp-container">

      <div>

        <h4>Income</h4>

        <p id="money-plus" class="money plus">+$0.00</p>

      </div>

      <div>
```



```

<h4>Expense</h4>

<p id="money-minus" class="money minus">-$0.00</p>

</div>

</div>

<h3>History</h3>

<ul id="list" class="list">

  <!-- <li class="minus">

    Cash <span>-$400</span><button class="delete-btn">x</button>

  </li> -->

</ul>

<h3>Add new transaction</h3>

<form id="form">

  <div class="form-control">

    <label for="text">Text</label>

    <input type="text" id="text" placeholder="Enter text..." />

  </div>

  <div class="form-control">

    <label for="amount">Amount <br />

    (negative - expense, positive - income)</label>

    <input type="number" id="amount" placeholder="Enter amount..." />

  </div>

  <button class="btn">Add transaction</button>

</form>

```

```
</div>
```

```
<script src="script.js"></script>
```

```
</body>
```

```
</html>
```

4.3.3 CSS code

```
@import url('https://fonts.googleapis.com/css?family=Lato&display=swap');
```

```
:root {
```

```
  --box-shadow: 0 1px 3px rgba(0, 0, 0, 0.12), 0 1px 2px rgba(0, 0, 0, 0.24);
```

```
}
```

```
* {
```

```
  box-sizing: border-box;
```

```
}
```

```
body {
```

```
  background-color: #f7f7f7;
```

```
  display: flex;
```

```
  flex-direction: column;
```

```
  align-items: center;
```

```
  justify-content: center;
```

```
  min-height: 100vh;
```

```
  margin: 0;
```

```
  font-family: 'Lato', sans-serif;
```

```
}
```

```
.container {
```

```
margin: 30px auto;

width: 350px;

}

h1 {

letter-spacing: 1px;

margin: 0;

}

h3 {

border-bottom: 1px solid #bbb;

padding-bottom: 10px;

margin: 40px 0 10px;

}

h4 {

margin: 0;

text-transform: uppercase;

}

.inc-exp-container {

background-color: #fff;

box-shadow: var(--box-shadow);

padding: 20px;

display: flex;

justify-content: space-between;

margin: 20px 0;

}
```

```
.inc-exp-container > div {  
  
  flex: 1;  
  
  text-align: center;  
  
}  
  
.inc-exp-container > div:first-of-type {  
  
  border-right: 1px solid #dedede;  
  
}  
  
.money {  
  
  font-size: 20px;  
  
  letter-spacing: 1px;  
  
  margin: 5px 0;  
  
}  
  
.money.plus {  
  
  color: #2ecc71;  
  
}  
  
.money.minus {  
  
  color: #c0392b;  
  
}  
  
label {  
  
  display: inline-block;  
  
  margin: 10px 0;  
  
}  
  
input[type='text'],
```

```
input[type='number'] {  
  
    border: 1px solid #dedede;  
  
    border-radius: 2px;  
  
    display: block;  
  
    font-size: 16px;  
  
    padding: 10px;  
  
    width: 100%;  
  
}  
  
.btn {  
  
    cursor: pointer;  
  
    background-color: #9c88ff;  
  
    box-shadow: var(--box-shadow);  
  
    color: #fff;  
  
    border: 0;  
  
    display: block;  
  
    font-size: 16px;  
  
    margin: 10px 0 30px;  
  
    padding: 10px;  
  
    width: 100%;  
  
}  
  
.btn:focus,  
  
.delete-btn:focus {  
  
    outline: 0;  
  
}
```

```
.list {  
  
  list-style-type: none;  
  
  padding: 0;  
  
  margin-bottom: 40px;  
  
}  
  
.list li {  
  
  background-color: #fff;  
  
  box-shadow: var(--box-shadow);  
  
  color: #333;  
  
  display: flex;  
  
  justify-content: space-between;  
  
  position: relative;  
  
  padding: 10px;  
  
  margin: 10px 0;  
  
}  
  
.list li.plus {  
  
  border-right: 5px solid #2ecc71;  
  
}  
  
.list li.minus {  
  
  border-right: 5px solid #c0392b;  
  
}  
  
.delete-btn {  
  
  cursor: pointer;
```

background-color: #e74c3c;

border: 0;

color: #fff;

font-size: 20px;

line-height: 20px;

padding: 2px 5px;

position: absolute;

top: 50%;

left: 0;

transform: translate(-100%, -50%);

opacity: 0;

transition: opacity 0.3s ease;

}

.list li:hover .delete-btn {

opacity: 1;

}

CHAPTER 5 : TESTING AND IMPLEMENTATION

The term implementation has different meanings ranging from the conversion of a basic application to a complete replacement of a computer system. The procedures however, are virtually the same.

Implementation includes all those activities that take place to convert from old system to new. The new system may be totally new replacing an existing manual or automated system or it may be major modification to an existing system. The method of implementation and time scale to be adopted is found out initially. Proper implementation is essential to provide a reliable system to meet organization requirement.

5.1 Unit testing

The unit testing part of a testing methodology is the testing of individual software modules or components that make up an application or software.

Test case for installation

SN	Test case id	Test Description	Input test data	Expected Result	Actual result	Remarks
1	TC-INS-01	Install DET app	Transfer DET app	Open application in homepage	Application Executed in Homepage	Pass

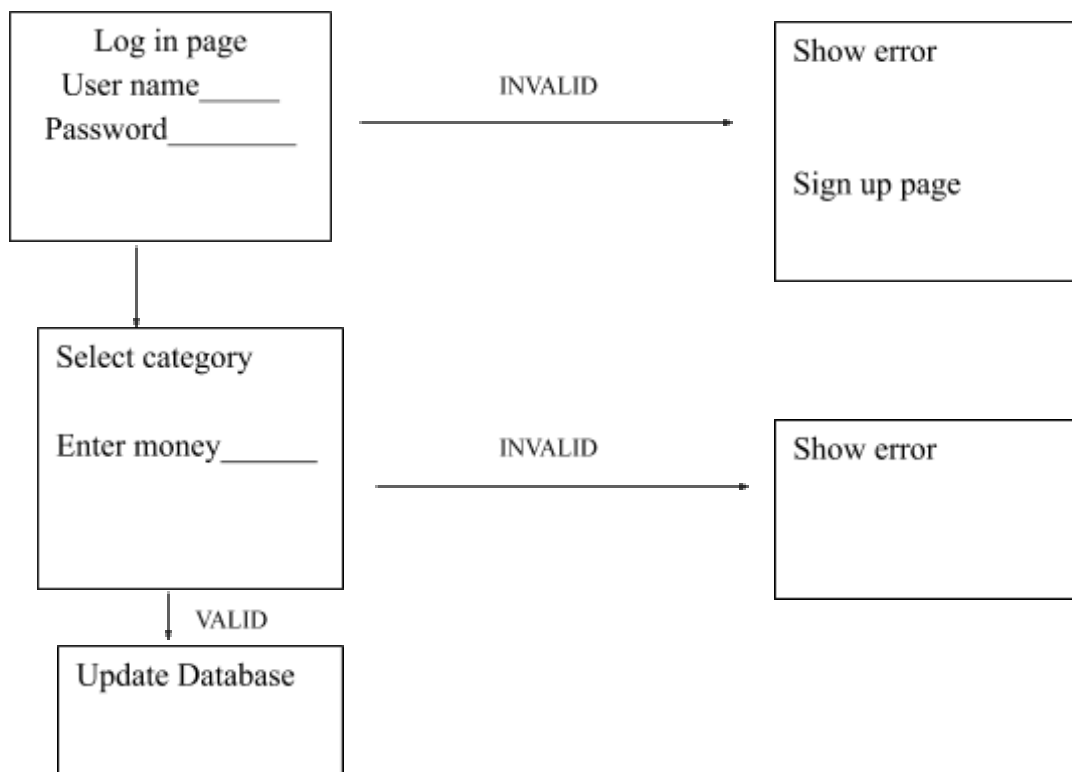
Test case for data entry

SN	Test case id	Test description	Input test data	Expected result	<i>Actual result</i>	<i>remarks</i>
1	TC-DT-01	Enter expense with numeric value	1500	Update table with 1500	Updated table with 1500	pass
2	TC-DT-02	Enter non-numeric value	leela	Show error	Printed "Invalid value"	pass
3	TC-DT-03	Enter decimal value	155.65	Update table with 155.65	Updated table with 155.65	pass
4	TC-DT-04	Enter	1860	Update	Can not	fail

		expense value without any category		default category with 1860	update table	
5	TC-DT-05	Enter numeric value in name field	13	Update table with 13	Can not update table for numeric	fail
6	TC-DT-06	Enter future date for expense	2021/02/16	Show error for maintaining future date	Updated table for future date	fail

5.2 Integration Testing

The integration testing part of a testing methodology is the testing of the different modules that have been successfully unit tested when integrated together to perform specific task and activities. The test is often done on both the interfaces between the components and the larger surface being constructed, if its equality property can not be accessed from its components. After integrating the requirements we tested it, it was fine and satisfactory.



CHAPTER 6 : ADVANTAGES

- This system helps house wives to reduce their expenses.
- Easy to use
- You'll have better insight into your spending habits
- provides a better overview and comprehensive analysis

CHAPTER 7 : CONCLUSION

After making this application we assure that this application will help its users to manage the cost of their daily expenditure. It will guide them and aware them about there daily expenses. It will prove to be helpful for the people who are frustrated with their daily budget management, irritated because of amount of expenses and wishes to manage money and to preserve the record of their daily cost which may be useful to change their way of spending money. In short, this application will help its users to overcome the wastage of money.

7.1 Recommendation

DET app is usable by anyone who are willing to manage their expenses and aiming to save for their future expenses. This app has no range criteria or any kind of profession or any kind of gender is focused so, it can be used by any person.

7.2 Future scope

In future days, there will be mail and paymode embedded wit this app. Also, backup details will be updated to cloud.

Bibliography

- 1) N.Mehta, “android application expense manager”, 2014.
- 2) A. Bamne, “Expense Tracker (UML)”,2014.
- 3) L.T.Hong, “Expenses with geo-location tracking.”, University Tunku Abdul Rehman, UTAR, 2015.

Reference

- 1) Wikipedia.org
- 2) Javatpoint.com
- 3) W3schools.com