

A Project Report

on

Budget Spy

*Submitted in partial fulfillment of the
requirement for the award of the degree of*

B.Tech C.S.E



(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Name of Supervisor: Dr. Vipin Rai
Designation : Associate Professor

Submitted By

Umang Mundhra
18SCSE1010298

Anushka Priya
18SCSE1010220

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING /
DEPARTMENT OF COMPUTERAPPLICATION
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
DECEMBER, 2021**



**SCHOOL OF COMPUTING SCIENCE AND
ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA**

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled “**BUDGET SPY**” in partial fulfillment of the requirements for the award of the Bachelor of Technology in Computer Science and Engineering in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of July, 2021 to December and 2021, under the supervision of Dr Vipin Rai (Associate Professor), Department of Computer Science and Engineering, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Anushka Priya, 18SCSE1010220
Umang Mundhra, 18SCSE1010298

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Vipin Rai
Associate Professor

CERTIFICATE

The Final Project Viva-Voce examination of Anushka Priya,18SCSE1010220 and Umang Mundhra,18SCSE1010298 has been held on 20-12-2021 and his/her work is recommended for the award of Bachelor of Technology in Computer Science and Engineering and Bachelor of Technology in Computer Science and Engineering respectively

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Project Coordinator

Signature of Dean

Date: December, 2021

Place: Greater Noida

Abstract

We created a web application called "Spy Budget". As the name suggests, this project is a web application used to track user income and expenditure so that one knows what they are spending their money on and what things are bringing in the most money. It is similar to digital record keeping that keeps track of the costs incurred by the user. This app captures user revenue by speech or manual recognition, and manages its daily expenses for the user to save money. Budget Spy not only tracks budget tracking but also provides ways to analyze data with charts and graphs and intelligently predict future budgets and problems such as collapse.

The problems identified with the actual system being used by most people are as follows:

- Much detail cannot be kept since there is much rewrite to be done which makes the task boring.
- Involves much paperwork which:
 - occupy lots of space, can be easily lost or stolen, may become dirty or get damaged thus causing lost of information, may cause clarity problems thanks to handwriting problems.
- Searching and referencing are difficult and time-consuming.
- Analysis and comparison of data becomes difficult.
- Reports generation is a tedious process.
- The work being done isn't accurate.

A solution for the matter described above is to supply a web means to store and manage the financial data (budget) of those people and this may involve two-tier architecture:

the database tier where of these data are stored and therefore the application tier which the members will use to realize access to the system. The main reason why the proposed application should be online is that users can access their financial data from anywhere at any time via the web either from the browser of a computer or that of latest mobile phones.

The proposed system must be ready to support many users at an equivalent time.

It must provide users with an appropriate interface where they will store all their financial data for various months/years online.

We have used React for front-end, JavaScript, state management using react context API, local storage for saving the data, Material UI for creating a user-interface, Speechly for adding voice capabilities.

This user friendly web application is for keeping our day-to-day expenditures through voice recognition as well as manually (depending on user's choice) which will help us to keep record of our money daily.

- 1) It will have various options to stay records (for example Food, Travelling Fuel, Salary, etc.).
- 2) Automatically it'll keep it up sending notifications for our daily expenditure.
- 3) In today's busy and expensive life, we are during a great rush to form money, but at the top of the month we broke off. As we are unknowingly spending money on titles and unwanted things. So, we have come over with decide to follow our profit.

Here user can define their own categories for expense types like food, clothing, rent, and bills where they need to enter the cash that has been spend and likewise can add some data in extra data to indicate the expense.

Table of Contents

Title	Page No.
Candidates Declaration	I
Acknowledgement	II
Abstract	III-IV
Contents	V-VI
List Of Figures	VII
Acronyms	VIII
Chapter 1 Introduction	1
1.1 Introduction	1-2
Chapter 2 Literature Survey	3
2 Literature Survey	3
2.1 Rationale	4-5
2.1.1 Tangible Benefits	
2.1.2 Intangible Benefits	
2.2 Parent Objectives	6-7
2.3 Special Functionalities of the System	8-9
2.4 Learning Opportunities	10-11
2.5 Nature of Challenge	12
2.6 Domain Research	13
2.7 Critical Appraisal	14
2.8 Critical Evaluation of Analysis	15
2.9 Design	16-18
3.0 High Level Design	19
3.1 Data Flow Diagram	20
3.2 Daily Expense Tracker	21-22
3.3 Use Case of DET	23
3.4 Use Case of Reports	24-25
3.5 Use Case Future Prediction	26
3.6 Feasibility Study	27-28
3.7 Hardware Requirements	29
3.8 Software Requirements	30

Chapter 3	Existing System	31
Chapter 4	Proposed System	32
	4.1 Flow of Architecture	33
	4.2 Features of the Applications	34
	4.3 Uml Diagram	35
	4.4 Class Diagram of Web Model	36
	4.5 Block Diagram	37
	4.6 Working of the System	38-39
	4.7 Module	40
	4.8 Tools and Technology Used	41
	4.8.1 Vs Code	41
	4.8.2 React	42
	4.8.3 Node JS	43
	4.8.4 React Charts	44
	4.8.5 Speechly	45
	4.8.6 Code Snippet	46
	4.8.7 Implementation Snippets	47
Chapter 7	Result And Conclusion	48
	7.1 CONCLUSION AND FUTURE SCOPE	48
	Reference	49-50

List of Figures

S.No.	Title	Page No.
1	Table 1	17
2	High Level Design	19
3	Data Flow Diagram	20
4	Use Case of the whole system	22
5	Use Case of the Expense Tracker	23
6	Table 2	23
7	Use Case for Reports	24
8	Table 3	25
9	Use Case Future Prediction	26
10	Table 4	26
11	Flow of Architecture	33
12	UML	35
13	Class Diagram of Web Model	36
14	Block Diagram	37
15	VS Code Logo	41
16	React Logo	42
17	Node JS Logo	43
18	React Charts	44
19	Speechly	45
20	Code Snippets	46
21	Implementation Snippets	47

Acronyms

B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
BCA	Bachelor of Computer Applications
MCA	Master of Computer Applications
B.Sc. (CS)	Bachelor of Science in Computer Science
M.Sc. (CS)	Master of Science in Computer Science
SCSE	School of Computing Science and Engineering

CHAPTER-1

INTRODUCTION

Budget tracking involves recording and analyzing the income and expenses of an individual or organization over a while. Today, as we live in a fast-paced and socially prosperous world, many people look forward to more efficient ways of budgeting their time and money. Over the years, some research has been done on the budget system. It has been noted that in many cases, budget management is done mentally and never put on paper which makes budget compliance very difficult. This may be because most people do not know how to do so or not to have appropriate mechanisms in place to monitor budgeting and self-analysis. Creating a budget also requires looking ahead and acting responsibly for goals, for the future. By creating a budget, people can set goals to reach a certain level of income and look after their expenses. Many small business owners have realized that their increase in profit margins has not happened until they have a written intention and an effective way to do it. BUDGET SPY is developed using React, Context API, Local Storage, Material UI, Speechly. One of the biggest goals in building this app is to create a clear picture of the financial situation of users by recording and keeping track of costs with voice recognition, as it happens. This online web app will help reduce the existing budget issues (paperwork) and provide methods for budget analysis and provide a quick and accurate response.

We have also used several dependencies listed below:

1. @material-ui/core
2. @Material-ui/icons
3. @material-ui/lab
4. @speechly/react-ui
5. @speechly/react-client
6. chart.js
7. reactchartjs-2
8. uuid

CHAPTER 2

2. LITERATURE SURVEY

- The idea of this system came into developers mind when he was unable to expenses when his guardians asked him for the expense report. His guardians asked the details of expenses which he had done during a month. Developer was unable to remember all those expenses.
- There are plenty of apps available right now that tracks the budget like “Good Budget”^[1] or “Mint”^[2] but they are not supported by voice recognition features.
- There are some research papers also available on this topic like “Expense Tracker”^[3] by atiya kazi , praphulla s. kherade , raj s. vilankar , parag m. sawant

2.1 RATIONALE

This application provides benefits in many ways, like it allows users to save their daily expenses in a customized manner. It allows users to save their task in the system along with the position from where they saved the data. It allows users to save their wastage of money by providing notifications whenever required. It allows the users to know their future benefits by doing calculations based on their savings. The system to be implemented could bring about significant tangible and intangible benefits. Given below is a list of tangible and intangible benefits expected from the system.

2.1.1 Tangible Benefits

One time investment: Users will have to invest their money for just once to use this system on android mobile but to use the web services they will have to pay according to the plan.

Saving on Internet Data usage: there is no need to start the internet all the time when you want to save the expense details as this can be done offline. User can synchronize their data during their of time i.e. just once in a day or a week internet is needed. Hence wastage of data usage can be saved.

Accessibility of system: System can be accessible anywhere if the user has got the mobile. If mobile is not with him/her then the system can be accessible with the help of website if the user has a login id and password.

Reduces the stress: This system will directly reduce the stress from the user as the user will not have to calculate each and every thing ex. How much they have balance in their hand? Etc.

Data can be recovered: As the system database is online i.e. on remote server so if the user lost his/her device then they can get the desired data from the server again.

2.1.2 Intangible Benefits

Economic Benefits: with the help of this system we can save our lot of money which goes in waste land due to the lack of planning.

Save time: this system will save our time as we can do hours of work in a minute. E.g. if we need to search a person whom we have to give money from our notebook then it will take hours if there are thousands of record but with the help of this system it will take less than a minute.

2.2 PROJECT OBJECTIVES

The objective of this system is to design such a system for users which will resolve the problems related with daily expenses and save a lot of money by providing them necessary notifications and by showing their profit or loss based on daily expenses.

Functionalities of DET (Daily Expense Tracker) Functionalities of this system are divided into three categories namely Core, Enhanced and Special functionalities. These are given below.

Core Functionalities of the System Core feature of any project is the heart of that project or you can say the features without which system is incomplete. In this project, developer is developing a complete interactive, user friendly, effective, feature-rich, multimedia site. Core features are :

Save daily expenses: This is the one of the main functionality of this system which will save all the daily expenses which the user will like to save. Along with the expenses the location of the place using GPS technology will also be saved so that if the user wants to know from where they saved this detail they can easily know. User can save credits, incomes and debits detail as well in this system.

Profile management: Users can manage their profile using this functionality e.g. view their profile and make changes if required.

Putting limit on expense: With the help of this functionality users can put limit on their expense. If user crosses their budget of the month then a notification will be displayed to their cell phone i.e. you have Crossed your budget of this month.

Account management: Here we can manage our account e.g. customize the options, put limitations i.e. previous functionality will come in under this category, user can save the mobile no. on which the message will be sent.

Report Generation : Various types of reports will be generated within this section for the users. These reports are described in details in section

Searching of Expenses: With the help of this functionality users will be able to search the expenses they have done on a particular item.

Contact us: Users can contact with the developer with the help of this functionality whenever required.

Terms and Conditions : This functionality will display the terms and conditions of using the system.

Enhanced Functionalities of System: To make user understand the system better, enhance features are used.

Future Prediction: This functionality will predict the future savings based upon the current income and expense.

Synchronization of offline-online data : Using this functionality user can synchronize their data which is saved in mobile to the server. This will automatically be done, no need to click on any button.

Task Manager: This functionality will help the user to schedule their task to do in the mobile so that they can be up to date with their each and every task.

Saving location based on GPS: This functionality will save the location of that place in terms of latitude and longitude so that it nurture we want to know from where we have spent our money then we can easily know it.

2.3 SPECIAL FUNCTIONALITIES OF THE SYSTEM

Based on the research and if developer has the time and ability, these are the advanced features developer is going to attempt to implement this system.

Saving data on external database other than from any device: Data which is in the device should be synchronized with the server so that the user can access it from anywhere. Challenging part here is that if the user saves any data in mobile when there is no internet connection and after sometime user saves other data from web browser then it's very hard to make data consistency in the system.

Offline and online facility: providing both of the facility to the user so that user will feel free to work with this system. It was one of the most challenging part of this system. The problem was to execute the query placed in JSP page i.e. web application with the help of android device. Initially the developer was working with the localhost with port no. 8080 but he found error as server not found. Then after a lot of research work he found instead of writing localhost in android we have to write. This is the path to the localhost from emulator. This was just a single problem, more problems are described in implementation part.

Graphical data: Showing graphical data to the user so that user can understand his/her expenses easily. It was really a hard nut to crack for the developer as there is no tool in android to draw graphical reports. So developer did a lot of research on this part and after trying many API's developer had decided to work with Google graph.

System portability: making the system to be portable on various devices. This issue made the developer a lot to worry as many functions which are available in the android newer versions are not compatible with older version of android say API level 8 does not have isEmpty() function to check the string whether it is empty or not. So developer had to make his own function to do the same.

When the screen orientation is changed the data provided to the system is lost which has not been saved in android application hence managing this screen orientation to be compatible with the system is one of the challenge in developing this system. Managing reports and sending them on mail: various types of reports will be made and sent to the user's mail so that user can get a clear

and more detailed view of the reports. Apart from these one of the challenges is making the GUI follow HCI principles.

Future saving prediction: on the basis of saving of the week/month predicting the future savings is one of the challenges to implement.

Knowing the user: To understand the spending habits of people. The manner in which they spend the money, decisions they take before spending on something.

2.4 LEARNING OPPORTUNITIES

Following are the learning objectives while developing this system.

Learning and practices of XML and Web Technology As this system is going to be developed entirely on android technology so learning of android technology is obvious. And XML is one of the core features of android to create the layouts then its knowledge is also most required. Learning of synchronization of data between offline-online One of the toughest features of this system is to synchronize the data between mobile and server. So to achieve this target I need to learn the algorithm for the same.

Learning Web services: To implement the web-services developer has to learn the web services also. With the help of web-services this system will be able to synchronize the data between server and mobile.

Learning to integrate with web to develop a single system. After learning the web-services and android basic I will be able to implement the both to create a single integrated system. Learning and practices of the advance JSP-servlet and android to make graphs In order to provide graphical report I have to learn advance JSP-Servlet.

Learning of JSON (JavaScript Object Notation) In order to interchange the data from android mobile to the server JSON is required. So it should be studied to implement the web services Knowledge about DBMS or Databases To save the data in database knowledge of DBMS is required so that no duplication of data occurs in the system.

Concept of Human Computer Interaction HCI principles will make the system easy to be used by the user and hence it can play a vital role in the growth of the system in future. Learning for a development of Real time bug free system. In order to make the system bug free various testing approaches should be learn and implemented in this system example. unit testing **Learning and Practices of SQL Query** To insert the data and fetch it SQL queries should be learnt.

Learning algorithm for smart prediction based on inputs of a month In order to perform future prediction which is one of the functionality of DET i.e. EF1 one of the best algorithm should be implemented. So learning algorithms will help me out.

Learning of Google graph In order to produce graph in android as well as in web pages Google graph is very much efficient to be used.

Learning file handling In order to save small chunks of data in mobile and access it later on developer needs to learn file handling.

2.5 NATURE OF CHALLENGE

As described above the challenges are really going to make the developer a lot from this system. Some of the challenges are really tough to implement e.g. implementing the web- services in this system, showing graphical reports in order to make users understand their profit and loss etc. this system will be complete if and only if all the functionalities given in core are completed. This is also one of the major challenges i.e. to complete the system within given time frame.

Challenges in terms of Theory and Software

Challenges in terms of Software Challenges in terms of software are small as eclipse has given the developer the frame work, where the developer can work on this project. It has also provided the developer WDT (Web Development Toolkit) on which he has developed the system. And of course it is free of cost. No licensing is required for it. But yes the developer has to register himself to the Google MAP API to get a free developer key which has been used during the development of the system to use Google GPS services. Also while deploying the project developer need to register each individual for a specific key so that they can use the same. Apart from android software like Apache tomcat will be used which is also free for the developer point of view. For the backend process MYSQL database is used and to access it, developer has worked on Wampserver. Wampserver is a tool which provides MYSQL database. To debug the program developer has used emulator so that errors can be rectified as soon as possible.

Challenges in terms of Theory The main challenge comes in terms of theory for this system. Example. how to implement web-application with android device, how to use efficient graph theory for smart prediction, how to use graph, how to customize the options, etc. but yes these all problems are real which the developers enjoy.

2.6 DOMAIN RESEARCH

As this project is a web application. The development of this system needs a great research work. With research work done properly one can make out the success or failure of the project, as it provides complete exposure of knowledge, business, human networking, better insights and understanding of the required area. Domain research deals with the whole method of a Website building. Then the developer will include the study of J2EE, XML, Android, SQL Queries. To make the research better, developer has divided it in the following domains:

- 1. Web Research-** The developer should have knowledge of the Web trends and web-applications.
- 2. Site Development-** The primary domain research of project will be choosing the website development methodology as well as android application development. The entire methodology should be clear in mind so as to have clear picture of what next steps to follow for a successful completion of this project within specific time period.

2.7 CRITICAL APPRAISAL

There were many problems in the current system, which have been sorted out during the development of this system. All the challenges which were listed above were critical and have taken a lot of time of the developer. Not only had this, even to solve a single problem of the user developer had spent a lot of time. The most challenging part was to execute the query on JSP page with the help of emulator i.e. android device. A lot of errors were hampering the system to be completed. Most of them are defined in implementation part of the system. One of the major challenge was to get the data after clicking on the report button and then show them in List view (just like list in HTML). Initially developer was expecting the JSON Object to return all the values in list but it failed to return so developer had used File Handling for the same. Data from the server is saved in File in android and then it is fetched as per the requirement. This is the area on why developer has given a limited choice to the user to get less than 20-30 no. of records at a time.

2.8 CRITICAL EVALUATION OF ANALYSIS

After the analysis of the questionnaire the developer has come to the point that to make a system efficient and handy a suitable data gathering technique must be used. And hence after the analysis developer has found answer of many questions on which he was not sure about. Initially the developer had thought of the system which will consist of only saving and that too on webserver but according to the feedback of the users, he decided to make it for saving credits and debits as well and make the system for both offline as well as online. Developer has reached the conclusion because of the feedback of the users which has been gathered by the questionnaire and interview questions.

2.9 DESIGN

The design document that we develop during this phase is the blueprint of the software. It describes how the solution to the customer problem is to be built. Since solution to complex problems isn't usually found in the first try, iterations are most likely required. This is true for software design as well. For this reason, any design strategy, design method, or design language must be flexible and must easily accommodate changes due to iterations in the design. Any technique or design needs to support and guide the partitioning process in such a way that the resulting sub-problems are as independent as possible from each other and can be combined easily for the solution to the overall problem. Sub-problem independence and easy combination of their solutions reduces the complexity of the problem. This is the objective of the partitioning process. Partitioning or decomposition during design involves three types of decisions:- Define the boundaries along which to break; Determine into how many pieces to break; and identify the proper level of detail when design should stop and implementation should start. Basic design principles that enable the software engineer to navigate the design process suggest a set of principles for software design, which have been adapted and extended in the following list: A good designer should consider alternative approaches, judging each based on the requirements of the problem, the resources available to do the job. The design should be traceable to the analysis model. Because a single element of the design model often traces to multiple requirements, it is necessary to have a means for tracking how requirements have been satisfied by the design model. The design should not repeat the same thing. Systems are constructed using a set of design patterns, many of which have likely been encountered before. These patterns should always be chosen as an alternative to reinvention. Time is short and resources are limited! Design time should be invested in representing truly new ideas and integrating those patterns that already exist. The design should minimize the intellectual distance between the software and the problem as it exists in the real world. that 15, the structure of the software design should (whenever possible) mimic the structure of the problem domain. The design should exhibit uniformity and integration. A design is uniform if it appears that one person developed the entire thing. Rules of style and format should be defined for a design team before design work begins. A design is integrated if care is taken in defining interfaces between design components. The design activity begins when the requirements document for the software to be developed is available as is the case the waterfall model. Design is essentially the bridge between requirements The design of a

system is essentially a blueprint or a plan for a solution for the system. We consider a system to be a set of components with clearly defined behaviour that interacts with each other in a mixed defined manner to produce some behaviour or services for its environment. A component of a system can be considered a system, with its own components. In a software system, a component is a software module. The design process for software systems, often, has two levels. At the first level, the focus is on which modules are needed for the system, the specifications of these modules, and how the modules should be interconnected. This is what is called the system design or top-level design. In the second level, the internal design of the modules, or how the specifications of the module can be satisfied, is decided. This design level is often called detailed design or logic design. Detailed design essentially expands the system design to contain a more detailed description of the processing logic and data structures so that the design is sufficiently complete for coding. Because the detailed design is an extension of system design, the system design controls the major structural characteristics of the system. The system design has a major impact on the testability and modifiability of a system, and it impacts its efficiency. Much of the design effort for designing software is spent creating the system design. The input to the design phase is the specifications for the system to be designed. Hence, reasonable entry criteria can be that the specifications are stable and have been approved, hoping that the approval mechanism will ensure that the specifications are complete, consistent, unambiguous, etc. The output of the top-level design phase is the architectural design or the system design for the software system to be built. This can be produced with or without using a design methodology. Reasonable exit criteria for the phase could be that the design has been verified against the input specifications and has been evaluated and approved for quality.

After completing the analysis developer collects sufficient amount of data to model the system. Requirement models allow people to share a common vision of solving the system before the actual implementation. They confirm that the system is flexible enough to accommodate the level of quality of future business needs. Based on the needs of users and the detailed analysis of a new system, the new system has been designed. This is phase System Project, which is the most crucial step in developing a system. This developer will provide Structural Design System normally, the design proceeds in two stages Preliminary or general design and Structure or detailed design Preliminary or general design: In the preliminary or general design, the features of the new system are specified. The costs of implementing these features and the benefits to be derived are

estimated. If the project is still considered to be feasible, we move to the detailed design stage. Structure or Detailed design: In the detailed design stage, computer oriented work begins in earnest. At this stage, the design of the system becomes more structured. "Structure design is a blue print of a computer system solution to a given problem having the same components and inter- relationship among the same components as the original problem. Input output and processing specifications are drawn up in detail. (Anonymous, 2010). So Developer analyzes the requirements and problem and decided to provide Structural Diagram or Detailed Design for these functionalities.

Traceability matrix between analysis and design

Requirement Analysis	Design		
	Use Case	Activity Diagram	Sequence Diagram
Profile Management	UC-01	AC-01	
Registration	UC-02	AC-02	SD-02
DET	UC-03	AC-03	SD-03
Task Manager	UC-04	AC-01	
Report	UC-05	AC-01	
Change Wall Paper	UC-06	AC-01	
Future Prediction	UC-07	AC-01	
Send Report	UC-08		
View Route	UC-09		

Table 1

HIGH LEVEL DESIGN

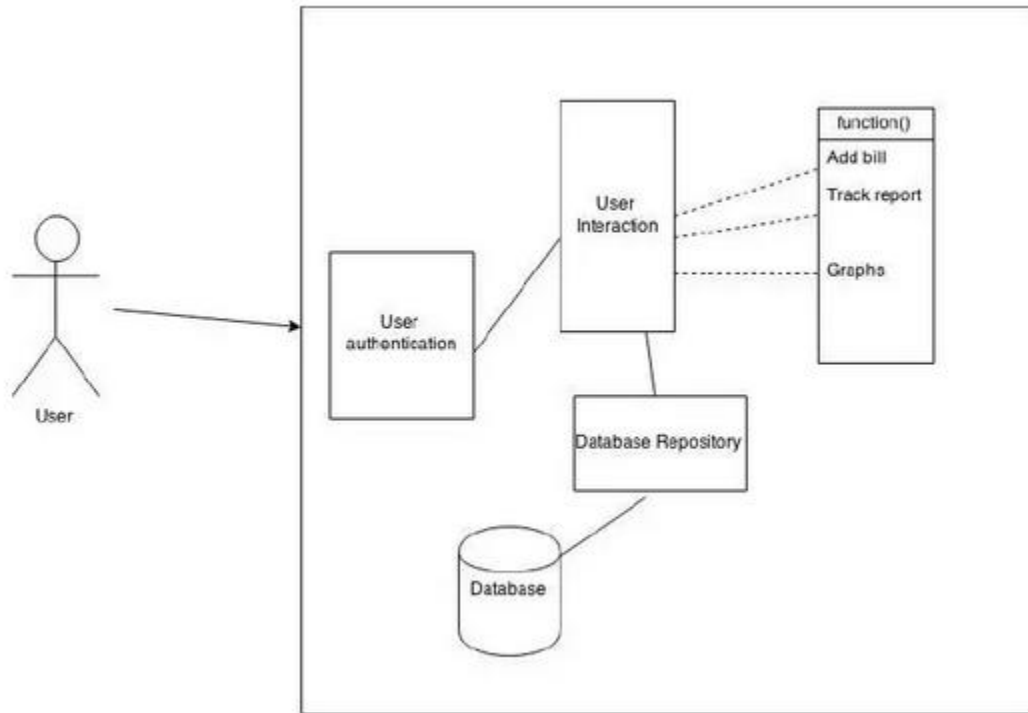


Fig: High level design

DATA FLOW DIAGRAM

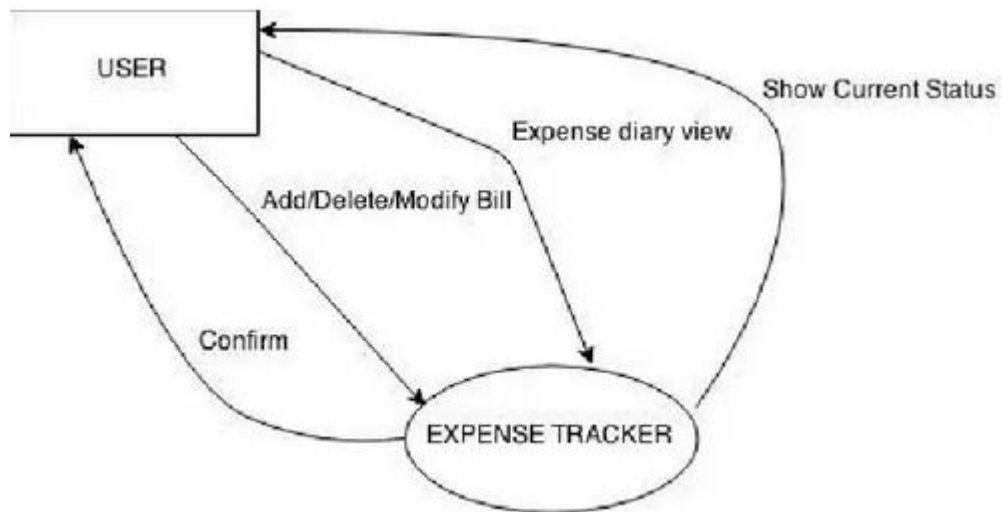


Fig: Data flow diagram

3.2 DAILY EXPENSE TRACKER

Design and Methodology Used

There are basically two types of design methodology we have: function-oriented and object oriented design. Here developer selects Object Oriented Design Concept

Justification of Selecting OO Design

The emphasis on defining software objects and how they collaborate to fulfill the requirements. As a developer uses an object-oriented language (Java) as a platform for this application so that glass is removed from the system needs to be modeled in the form of classes and objects. As a developer result concludes that the DET has to be designed using principles of object oriented design. To implement the project developer chose to make use of UML (Unified Modeling Language). Why use UML? Developer chooses UML because he has chosen Object Oriented

Design Methodology

Use Case Use case diagram is used to describe the functionality of the system, which mainly include the target users and perspective. It helps in the search for "who is using the system, what are the common scenarios and what they are trying to achieve your goals or" instead of how it will achieve. After analyzing the system, use case are used by developer to document the requirements of the proposed system. Use cases are divided according to different scenario that shows how the system interacts with the user. In the case of use-case we do not use any technical language, so they are easy to understand. Use cases are not used to represent the inner workings of the system, but showing the step that the user will take to perform any specific task. The system requirements will be evident after developing use cases of different modules. Below are additional reasons why you need to indicate use cases for project developers

1. Use Cases helps in finding out the challenges involved in making the Project Planner System and to decide the functionalities to be included in system.
2. Use cases will be helpful for non-technical person to understand the flow of system.

3. Moreover, to give the overview of the system it is necessary to include use case.

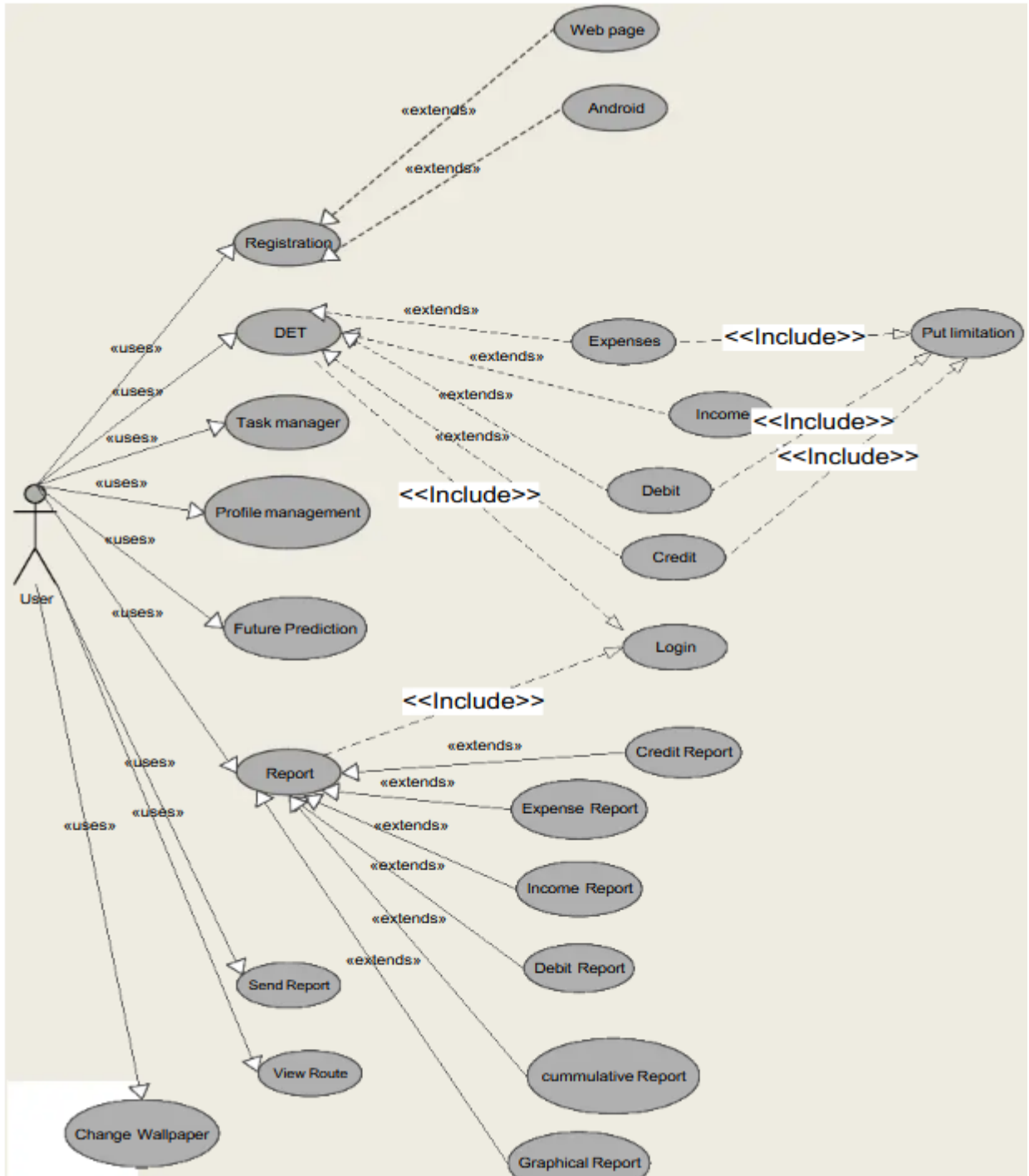


Fig: Use Case of the whole system

3.3 USE CASE OF DET

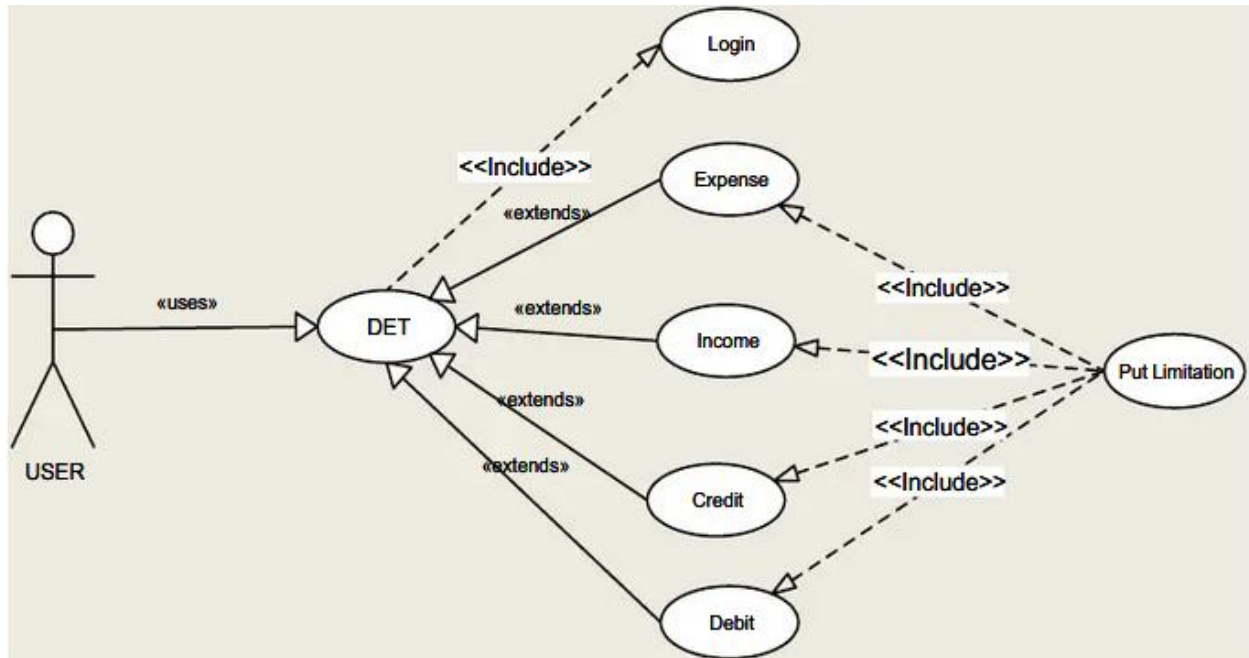


Fig: Use case of expense tracker

Use Case ID	UC-03
Name	DET
Goal	To manage users daily expenses
Description	It is used to save expenses and view reports.
Actor(s)	Application-User
Assumption	<ul style="list-style-type: none"> Users are using this application to manage their daily expenses.
Pre-Condition	<ul style="list-style-type: none"> User has logged in the system.
Post-Condition	<ul style="list-style-type: none"> User can save the records
Happy Pathway	<ol style="list-style-type: none"> User uploads the data(all their expenses) Admin runs the monitor application Admin analyzes the reports
Alternate Pathway(s)/Exception Pathway(s):	<ol style="list-style-type: none"> No user uploads the data means no one is using application User can enter all detail of their daily expenses.

Table 2

3.4 USE CASE FOR REPORTS

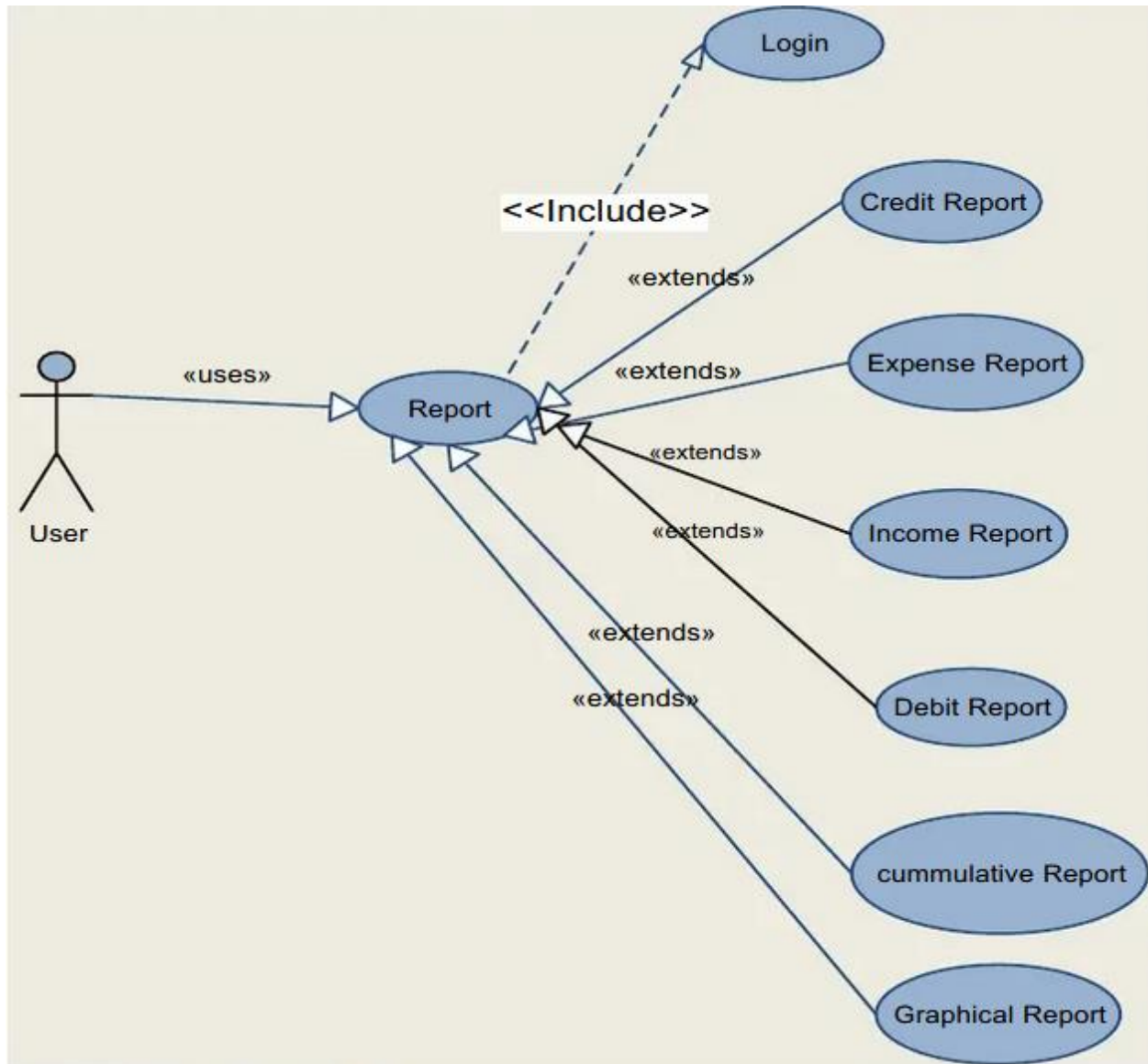


Fig: Use case for reports

Use Case ID	UC-05
Name	Report
Goal	User can view reports.
Description	User can view all the reports of DET e.g. credit report, debit report.
Actor(s)	User of the app
Assumption	<ul style="list-style-type: none"> • User has already inserted some records in that month.
Pre-Condition	<ul style="list-style-type: none"> • User will have to login to the system.
Post-Condition	<ul style="list-style-type: none"> • User views the list of expenses and detailed expense report of a particular item.
Happy Pathway	1. User views the expense, income, and credit and debit report.

Table 3

3.5 USE CASE-FUTURE PREDICTIONS

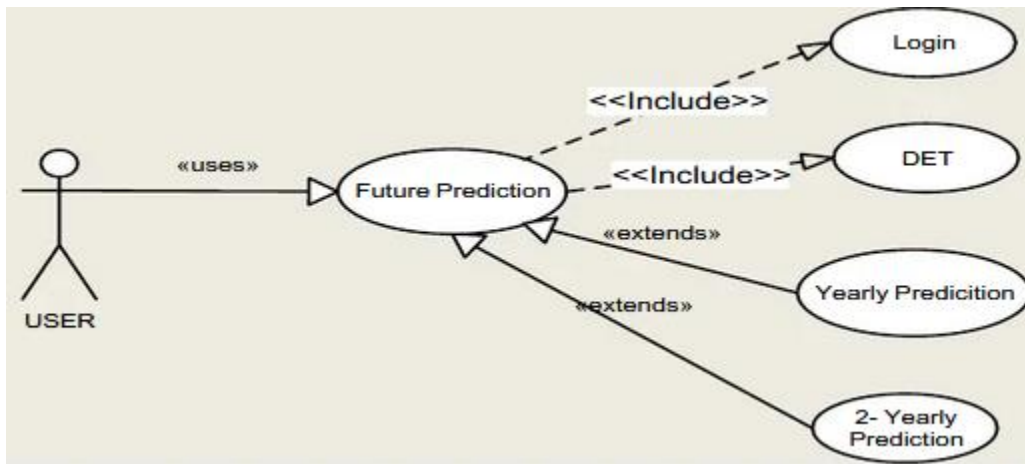


Fig: Use Case Future Prediction

Use Case ID	UC-07
Name	Future Prediction
Goal	To view the future savings.
Description	User can view the expected future savings
Actor(s)	User of the app
Assumption	<ul style="list-style-type: none"> • User has saved some records in the system. • User has access of internet.
Pre-Condition	<ul style="list-style-type: none"> • User has logged in the system • User has saved some records in the system.
Post-Condition	<ul style="list-style-type: none"> • User has viewed the future prediction.
Happy Pathway	1. User is able to view the future prediction.
Alternate Pathway(s)/Exception Pathway(s):	User can calculate the future prediction manually.

Table 4

3.6 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

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.

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.

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.

3.7 HARDWARE REQUIREMENTS

- Processor: Pentium 4 (or equivalent)
- 4 GB RAM
- Hard disk space: 20 GB
- A projecting device (for the instructor only)
- A connection to the internet
- Keyboard and mouse or other pointing device
- Processor: Pentium 4 (or equivalent)

3.8 SOFTWARE REQUIREMENTS

- Operating System: Windows 10 version 1507
- Node.js (<https://nodejs.org/en/>)
- Supported Internet browser: Chrome - Latest version, or the penultimate version

CHAPTER-3 EXISTING SYSTEM

Existing system does not use the smart concept which are used now a days. 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.

DISADVANTAGES OF EXISTING SYSTEM:

- The existing system is not user friendly because data is not maintained efficiently.
- But this project will not have any reminder to remain a person in a specific date, so that is the only drawback in which the remainder is not present.
- This project will be an unpopulated data because it has some disadvantages by not alerting a person for each and every month. But it can used to perform calculation on income and expenses to overcome this problem we have the proposed system.

CHAPTER - 4

PROPOSED SYSTEM

This new Online Income and Expense Tracker will eliminate all the demerits which are found under the existing system. To reduce manual calculations, we propose an application which is developed by php. 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 project which will keep a track of Income-Expense of a user on a day to day basis. This project takes Income from user and divides in daily expense allowed. If you exceed that day's expense it will cut it from your income and give new daily expense allowed amount, and if that day's expense is less it will add it in savings. Expense tracker will generate report at the end of month to show Income-Expense via multiple graphs. Expense tracking application system can generate report at the end of week or month to show Income-Expense via multiple graphs. It will let you add the savings amount which you had saved for some particular Festivals like Diwali, Birthdays. If we exceed the target of our budget it is automatically generate the notification that will sent via E-mail. An email will be sent to the user at the end of each month giving a brief summary of the monthly expenditure.

ADVANTAGES OF PROPOSED SYSTEM:

- After logging into the system, a user can add the bills with an option to attach the image of the bill or not.
- The option to attach a bill helps the user to remember when and where the payment was made.
- The user can also add the information about how the payment was made i.e. via check, card or cash.
- There is also an option to view owe and lend expenses which adds or gets deducted from the overall budget according without bothering the user.

4.1 FLOW OF ARCHITECTURE

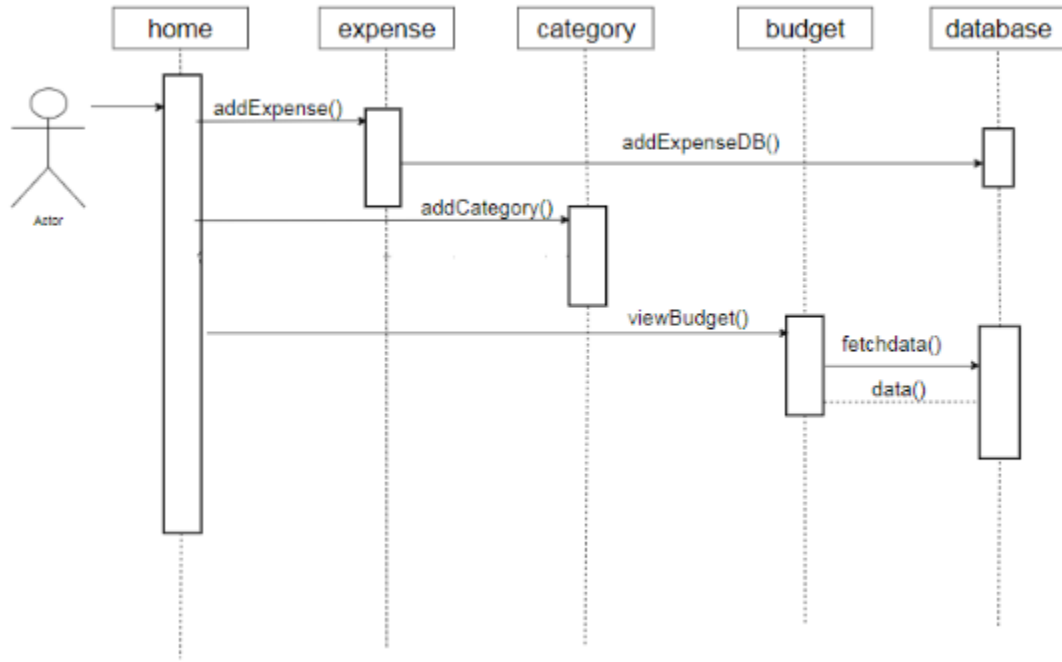


Fig: Flow of Architecture

4.2 FEATURES OF THE APPLICATION

1.Main Window

A Starting page with welcome note appears at the first page and the window where user can add the expense.

2.Add Category

The system shall allow the user to add category as per his/her choice.

3.Category Combo Box

This Combo Box contains all the category added by the user.

4.Remove Button

This Button delete any entry of last 20 days expenses.

5.Add Category Window

This window helps the user to easily interact with the category, where user can add and remove the category as per his choice.

6.Calender

The system shall allow users to add the date to their expenses

7.View Expense Day wise/Category wise Window

This window allows the user to interact with the expense according to day wise as well as category wise.

8.Menu option

This allows user to interact with other pages by just clicking on the menu option.

4.3 UML CLASS DIAGRAM

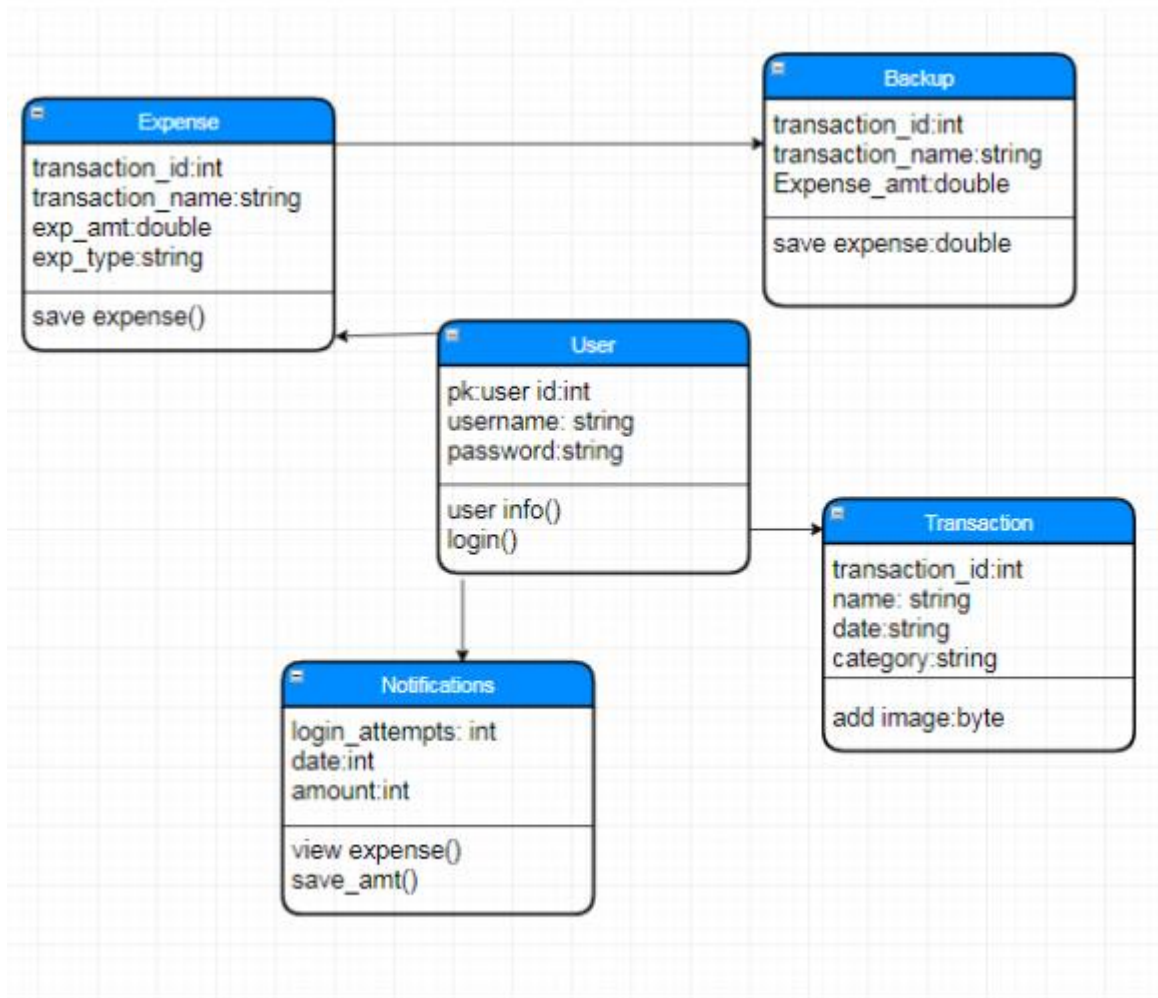
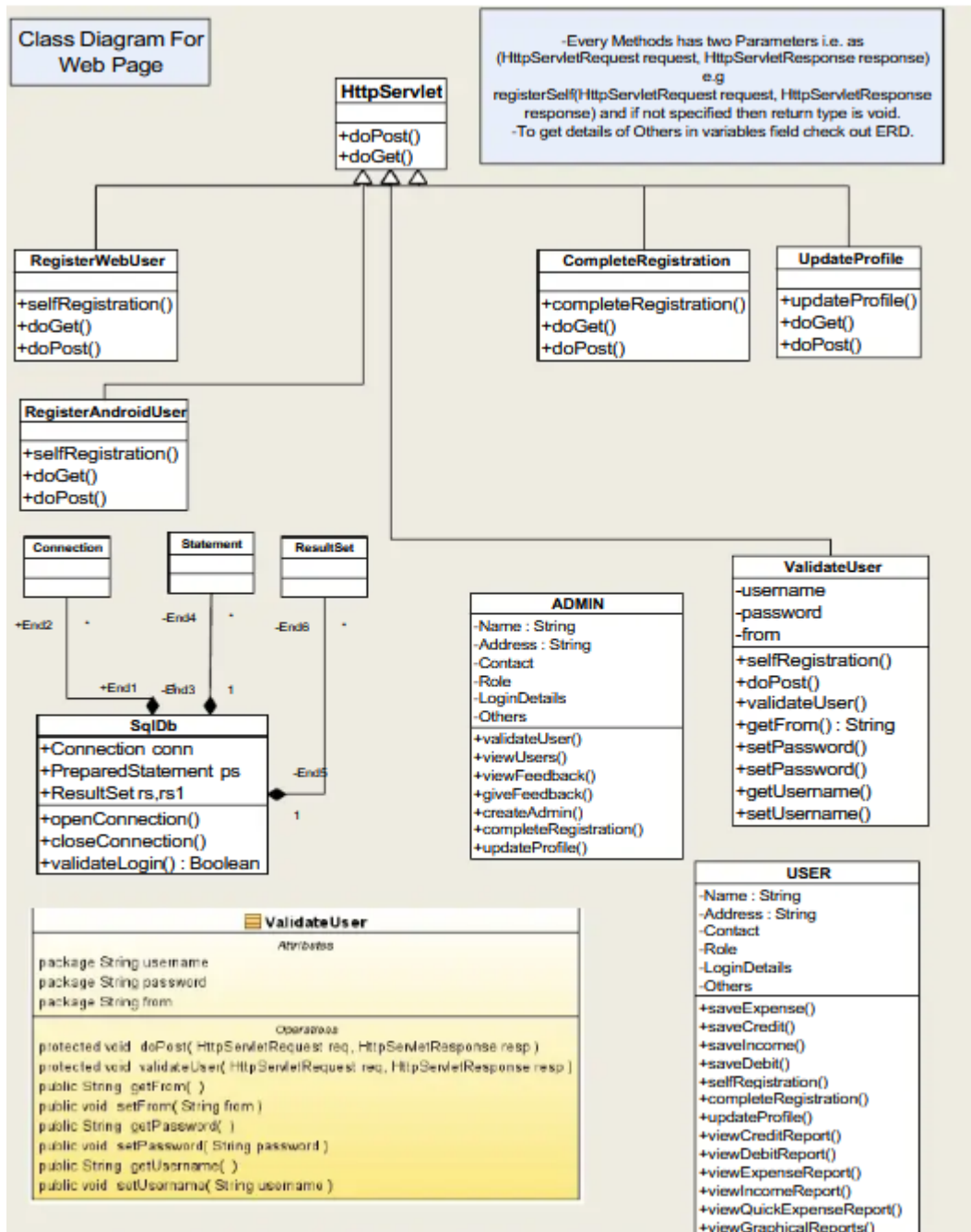


Fig: Uml diagram

4.4 CLASS DIAGRAM OF WEB MODEL



4.5 BLOCK DIAGRAM

In the figure given below, the rough project design is shown.

Once the user opens the web application, there would be a button for voice capabilities and it is totally up to you if you want to enter the details manually.

First scenario, if the details are entered through voice, speechly provides a stream of annotated transcript, expense or income is saved in local storage and finally graphs are updated with recent transaction.

Second scenario, if the details are entered manually, expense or income is saved in local storage and hence the graphs will be updated with recent transactions.

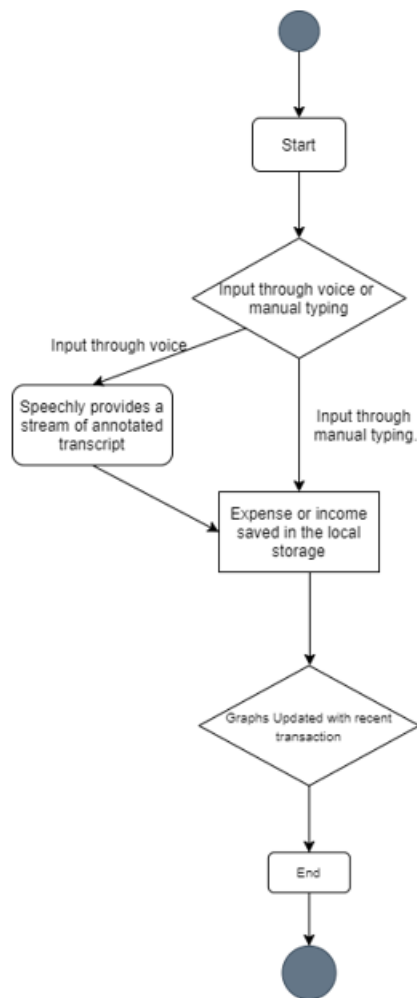


Fig: Block Diagram

4.6 WORKING OF THE SYSTEM

The name of our project is “BUDGET SPY” which is basically an expense tracker. It works on the top of latest technology stack and is as interactive and friendly as it is named. You have to provide the input manually or with the help of your speech (thanks to speechly for making it happen easily). The moment you fill in the input manually and click the "CREATE" button or the moment you provide your voice command , the app stores the data with the help of context that makes your app faster and the information is easily flowed between the various components of the app. Then the information is populated in a list which is displayed just below the input form . The list holds all the transaction be it any expenditure or income for your reference so that any time you want to have a look at your previous activities you could easily have it.

As we know , the technology is moving from alphabetical data to the images as they are easy to read , understand and are attractive. So our budget spy is not behind in this trend also. The graph of income and expenses updated automatically as soon as any transaction is added to the list. The next thing that is more important is all the information is stored in the local storage so that any time you want to clean the tabs or even shut down the browser , and the next time you open it you have all you transactions saved just like before. The possibility of human error is always estimated while creating any app so the Budget Spy also has the options to delete any transaction that you typed incorrectly so that your budget sheet is always very precise and accurate.

We have used certain dependencies :

1. @material-ui/core
2. @Material-ui/icons
3. @material-ui/lab
4. @speechly/react-ui
5. @speechly/react-client
6. chart.js
7. react-chartjs-2
8. uuid

There are several **npm** packages that allow charts to be created in React; **chart.js** is one of the packages which makes the creation of charts and graphs very easy.

MUI^[6] provides a robust, customizable, and accessible library of foundational and advanced components, enabling you to build your own design system and develop React applications faster.

Advantages of Material UI

Beautifully Designed

We can start our own projects with Google's Material Design or build our own designs using the sophisticated theming features.

Easily Customized

We can easily enjoy the power of our components without sacrificing the styles we want. We can also tweak on how our components render down to the very last class.

Superb Documentation

Our docs were shaped throughout the years with the help and experience of the open source contributors. Everything is just there.

Accessible in mind

We improve accessibility for all of our components constantly, helping us to reach the largest audience possible.

There are several **npm**^[7] packages that allow charts to be created in React; **chart.js** is one of the packages which makes the creation of charts and graphs very easy.

4.7 MODULE

A modularization consists of well-defined manageable units with well-defined interfaces among the units. Desirable property of modular system include

- a) Each module is a well-defined sub-system.
- b) Single, well – defined purpose of each module.
- c) Modules can be separately compiled and stored in a library.
- d) Modules can use another module.
- e) Modules should be easier to use than to build.
- f) Modules should be simpler from outside then from inside.

The project can be decomposed in following modules:

- a) Welcome module: This module is a starting page of this application.
- b) Main window module: This module is responsible for input the data and store in the database.
- c) Add amount: This module is responsible to enable the user to add amount in the database.
- d) Delete expense: This module is responsible for deleting the previous 20 days expenses.
- e) View Expense: This module is responsible for viewing all the expenses in detail added to the log by the user earlier.
- f) Add Module: This module is responsible for editing the pre-defined category.
- g) Categories module: This module is responsible for various options. In this app users have options of selecting various basic expense categories according to their choice.

4.8 TOOLS AND TECHNOLOGY USED

4.8.1 VS CODE

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. It is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++. It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services). Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis, and add code linters using the Language Server Protocol. Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software. Visual Studio Code allows users to set the code page in which the active document is saved, the newline character, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language.



Fig: VS Code

4.8.2 REACT

React is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications.

Declarative

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes. Declarative views make your code more predictable and easier to debug.

Component-Based

Build encapsulated components that manage their own state, then compose them to make complex UIs. Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

Learn Once, Write Anywhere

We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.

React can also render on the server using Node and power mobile apps using [React Native](#).



Fig: React Logo

4.8.3 NODE JS

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.

Node.js is primarily used to build network programs such as Web servers. The most significant difference between Node.js and PHP is that most functions in PHP block until completion (commands execute only after previous commands finish), while Node.js functions are non-blocking (commands execute concurrently or even in parallel, and use callbacks to signal completion or failure).

Platform architecture

Node.js brings event-driven programming to web servers, enabling development of fast web servers in JavaScript.¹ Developers can create scalable servers without using threading, by using a simplified model of event-driven programming that uses callbacks to signal the completion of a task Node.js connects the ease of a scripting language (JavaScript) with the power of Unix network programming.

Industry support

There are thousands of open-source libraries for Node.js, most of them hosted on the npm website. There are multiple developer conferences and events that support the Node.js community, including NodeConf, Node Interactive, and Node Summit as well as a number of regional events.



Fig: Node JS Logo

4.8.4 REACT CHARTS

The React Chart is a well-crafted charting component to visualize data. It contains a rich UI gallery of 30+ charts and graphs, ranging from line to financial that cater to all charting scenarios. Its high performance helps render large amounts of data quickly.

Time is short for front-end developers as it is, so having a charting system that is great out of the box, declarative, succinct and requires as little imperative scripting as possible not only helps you keep moving forward but lets you express your data visualization needs at the speed of your creativity.

React Chart's API's goal is to remove the necessity of knowing how to write and manipulate SVG elements. While powerful, SVG elements can often create a new layer of indirection for accomplishing simple customization tasks like tooltips, labels, annotations, etc. React Charts make this a breeze!

We believe data visualization is all about **effectively conveying information** to the users, and not about building *new* and *"exciting"* methods of indirection or *"art"* for them to ponder and decipher.



Fig: React Charts

4.8.5 SPEECHLY

Speechly is a tool for building real-time voice functionalities that integrate seamlessly to existing touch or web user interfaces. We don't think smart speakers or "voice-only" solutions is the best way to use voice and rather advocate multimodality and real-time visual feedback.

We don't believe that smart speakers and voice assistants are the best use case for voice, but voice should be thought of as an add-on to current mobile applications' and websites' user interface. Voice is a modality, not a complete user interface.

Touch screen user interfaces definitely need improvements: while selecting from a few options is easy, selecting for example 30 items from an inventory of 20.000 is pretty cumbersome.

Typing is notoriously hard, too. Most humans speak about three times faster with less errors than they type. In short, voice is a great solution for information heavy tasks. While there are good solutions for speech recognition, there's really no tools that would enable developers build the kind of user interfaces we've envisioned for voice

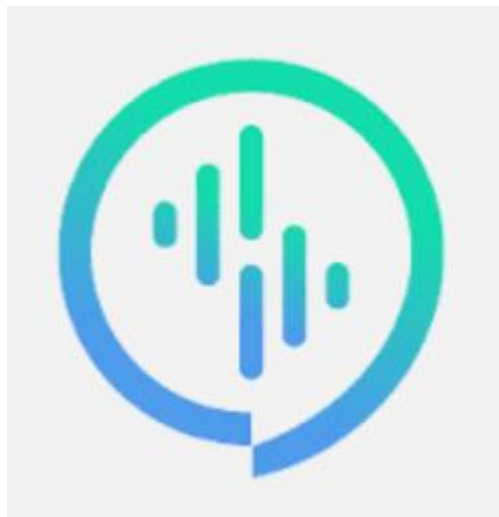


Fig: Speechly Logo

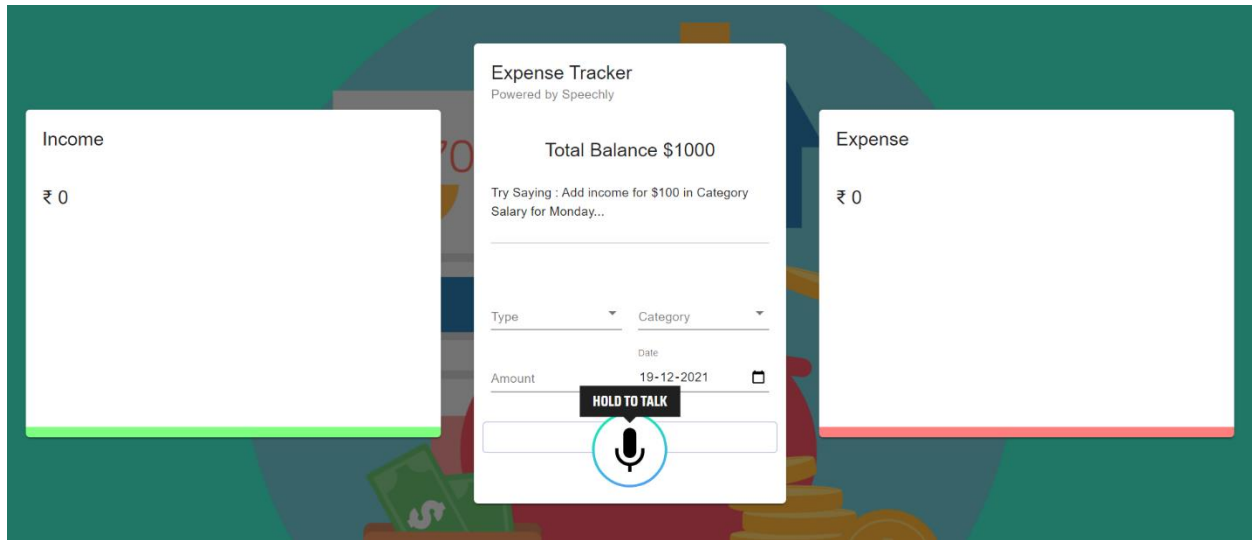
4.8.6 CODE SNIPPET

```
1 import React from 'react';
2 import Main from './components/Main/Main';
3 import Details from './components/Details/Details';
4 import { Grid } from '@material-ui/core';
5 import useStyles from './styles';
6 import { PushToTalkButton, PushToTalkButtonContainer, ErrorPanel } from '@speechly/react-ui';
7
8 const App = () => {
9   const classes = useStyles();
10  return (
11    <div>
12      <Grid className={classes.grid} container spacing={0} alignItems="center" justify="center" style={{ height: '100vh'}}>
13        <Grid item xs={12} sm={4}>
14          <Details title="Income" />
15        </Grid>
16        <Grid item xs={12} sm={3}>
17          <Main />
18        </Grid>
19        <Grid item xs={12} sm={4}>
20          <Details title="Expense" />
21        </Grid>
22        <div style={{ marginTop: '15rem' }} />
23        <PushToTalkButtonContainer>
24          <PushToTalkButton />
25          <ErrorPanel />
26        </PushToTalkButtonContainer>
27      </Grid>
28    </div>
29  );
30 }
31 export default App;
```

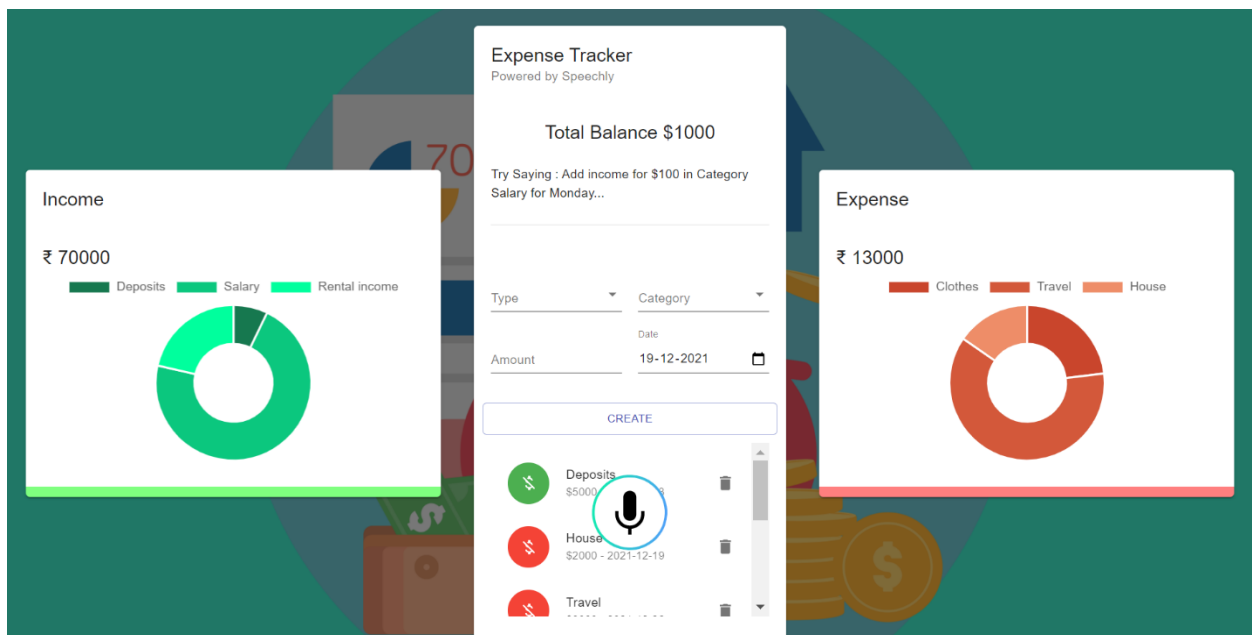
```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import { SpeechProvider } from '@speechly/react-client';
4
5 import { Provider } from './context/context';
6 import App from './App';
7 import './index.css';
8
9 ReactDOM.render(
10   <SpeechProvider appId="56a42060-25ad-42c5-9e3a-f7dfaa08f96f" language="en-US">
11     <Provider>
12       <App />
13     </Provider>
14   </SpeechProvider>,
15   document.getElementById('root'));
```

4.8.7 IMPLEMENTATION SNIPPETS

Home page



When we entered the details and the graph is shown on the either sides.



CONCLUSION AND FUTURE SCOPE

Monitoring our everyday expenses can set aside our cash, yet it can likewise help you set your monetary objectives for what's to come. On the off chance that you know precisely where your sum is going much of a stretch see where a few reductions and bargains can be made. Budget Spy project is for keeping our day-to-day expenditures using our own voice or manually will helps us to keep record of our money daily. The project what we have created is work more proficient than the other income and expense tracker. The project effectively keeps away from the manual figuring for trying not to ascertain the pay and cost each month. It's a user-friendly application.

In further days, there will be mails and pay mode embedded with the app. Also, backup details will be there in the local storage.

REFERENCES

1. <https://sciresol.s3.us-east-2.amazonaws.com/IJST/Articles/2015/Issue-Supplementary-2/Article16.pdf>
2. https://d1wqtxts1xzle7.cloudfront.net/60555255/IRJET-V6I3111020190910-38299-1elg6et-with-cover-page-v2.pdf?Expires=1639995853&Signature=T7-sSHDrERETKF8O621j5sxW6n0dqatrMrWr3JVq3bc7s4TCP2YCWBUdQIy8ERF6hjCiVEgKkTba0Fr3ltcOVx-OGAYsSWMQ9cmgmIr6PWPcowFfFrPxu4cwUWUfKYRLXpyq6C0dKeUo1eUzqbQ0tBCjk8rA6dRz2DsJl1zxCvTnnornz8EBbiKSYoJHE2tbOBbyC7btJk5YeidpFoAC91B5jFd2NXf-hZH1oWwStbZksTswWB~WqE17V6~WpLPToNHf6Xr2yousA6RW3s-vgpESJ8VqhOYnRNDVtzoT2raRq2AZ27-U3KOoKOXn1z9woHu~Jv1i~UMYB3CTAhXZJw__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA
3. <http://dspace.daffodilvarsity.edu.bd:8080/handle/123456789/4026> 2005.
4. <https://www.irejournals.com/formatedpaper/1702687.pdf>
5. <https://code.visualstudio.com/>
6. <https://www.speechly.com/>
7. Good Budget: : <https://goodbudget.com/> Mint: <https://mint.intuit.com/>
8. <https://www.irjmets.com/404.php>
9. Cho, Lee, Hahn, Kim, Lee and Lee. International Placement and Engineering Educational Accreditation. Proceedings of 99 International Conference on Engineering Education. p. 197–205.
10. ABET Engineering Accreditation Commission. 2005-2006 Criteria for Accrediting Engineering Programs. ABET, Inc; 2004.
11. <https://turcomat.org/index.php/turkbilmat/article/view/8759>
12. IRE Journal: Expense Tracker by Atiya Kazi , Praphulla S. Kherade , Raj S. Vilankar , Parag M. Sawant
13. shorturl.at/knADK
14. <https://www.educative.io/edpresso/how-to-use-chartjs-to-create-charts-in-react>.

