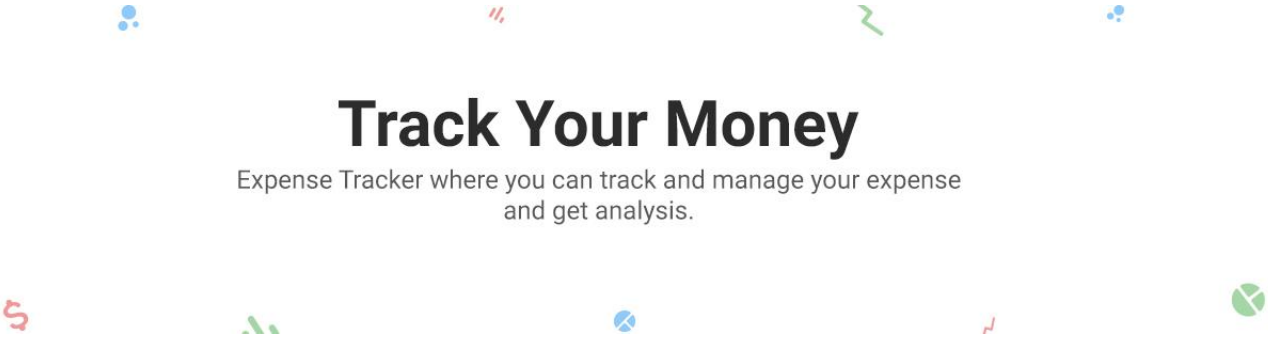

EXPENSE TRACKER

Project report submitted in fulfillment of the requirement for the award of the Diploma in Computer Engineering

By

Yash Gohel - 166020307041

Milap Gol - 166020307042

Vyomesh Jethva - 166020307047

Department of Computer Engineering

A V Parekh Technical Institute

(Affiliated to Gujarat Technological University, Chandkheda)

AVPTI RAJKOT

(Dept. of Technical Education, Govt. Of Gujarat)
Dr. Yagnik Road, Rajkot - 360001
PH - (O)0281-2480175
Email - avpti1948@yahoo.com

Certificate

This is to certify that the project report Expense Tracker being submitted by Yash Gohel, Milap Gol, Vyomesh Jethva in fulfillment for the award of the Diploma in Computer Engineering Sem : 5 to the Gujarat Technological University is record of bonafide work carried out by him under my guidance and supervision.

Mrs. Hemali Rupareliya

Head Of Department

Computer Engineering

Guided By: N.R. Radadiya

ACKNOWLEDGEMENT

It is our great pleasure to present our project report on Expense Tracker which we conceived documentation part during the semester-5thof Diploma of computer Engineering affiliated to Gujarat Technological University, Ahmadabad.

We would like to express a deep sense of gratitude and thank profusely our project guide Prof. N R Radadiya for his proper guidance and valuable suggestions. Without his wise counsel and able guidance, it would not have been possible to complete the report in this manner. It is a moment of great pleasure to acknowledge their help and encouragement. Their interest and constant encouragement helped us in making the report a success.

We are thankful to Head of the Department Mrs. Hemali Rupareliya madam and all the faculty members who have directly or indirectly helped us in completion of the Project.

Table of Contents

Abstract

1 Introduction

1.1 Characteristics of Existing System..... 2

1.2 Overview of Proposed System with Advantage 3

1.3 Scope 4

1.4 Process Model..... 5

2 System Requirements Specification

2.1 User Characteristics..... 11

2.2 Functional Requirements..... 13

2.3 Non-Functional Requirements..... 18

3System Analysis Modeling – User-based

3.1 Feasibility Study of the New System 21

3.2 User-Based Modeling..... 22

3.2.1 Use Case Diagrams 22

4 System Analysis and Design – Data-based

4.1 Data Modeling..... 26

4.1.1 Data Dictionary (List of Database Tables included in the system)..... 26

4.1.2 E-R (Entity-Relationship) Diagram..... 31

4.2 Behavioral Modeling..... 32

4.2.1 Data Flow Diagram 32

4.2.1.1 Context Level Diagram (Level 0) 32

4.2.1.2 DFD – Level 1 33

5 System Design – UML

5.1 Sequence Diagrams 35

5.2 Activity Diagrams 36

6 System Interface Design

6.1 Input- Output Forms Design 38

6.2 Graphical User Interface Design 39

7 Reference 41

8 Bibliography 42

List Of Figures

Sr No.	List	Page No.
1.1	Iterative Waterfall Model	6
3.1	Use case Diagram Administrator	22
3.2	Use case Diagram Authorized user	23
3.3	Use case Diagram Guest User	24
4.1	ER Diagram	31
4.2	DFD Level 0	32
4.3	DFD Level 1	33
5.1	Sequence Diagram	35
5.2	Activity Diagram	36
6.1	Input Output Signup	38
6.2	Input Output Login	38
6.3	GUI Home	39
6.4	GUI Signup	40

List Of Tables

Sr No.	List	Page No.
1	ET_USER	26
2	ET_PROFIL	27
3	ET_PASSWORD	27
4	ET_DATA	27
5	ET_CATEGORY	28
6	ET_FLAG	28
7	ET_DEFAULT	28
8	ET_EXPENSE	29
9	ET_DATE	29
10	ET_BANK	30

ABSTRACT

This project will help user to track and manage their daily Expenses easily on the single tap without requiring any special skill and knowledge. By using this website user don’t need to worry about current currency rate and we provides better synchronization and AI feature that informs you where and how much money you pay at certain places and for which purpose simply using user-friendly graphs. All the features are free for authorized user and all the details are secured and encrypted.

CHAPTER 1

INTRODUCTION

1 INTRODUCTION

1.1 CHARACTERISTICS OF EXISTING SYSTEM

Current working system has less users and doesn't have good graphical user interface. Most of service provider cost more for services with minimal features. It doesn't suited for normal users who are using this system for small business or for home.

Limitations of Current Systems:

- Expensive and not user friendly
- Privacy issues
- Extra charges for additional features
- These are some drawbacks is to learn and needed to implement as new powerful system

1.2 OVERVIEW OF PROPOSED SYSTEM WITH ADVANTAGE

Powerful technology and website implantation in proposed system. Batter customer and user friendly system with free of charges and no extra hidden charges. AI generated report analysis for accurate result. Well maintained data flows and information manipulation techniques applied.

Major Advantages of Proposed System:

- Easy for non-technical users
- Free services and user friendly
- Data privacy maintained
- All advantages &features overcome drawbacks of current systems. Proposed system provides better solution to users.

1.3 SCOPE

Administrator

- An Administrator is a user who manages the operational aspects of a system.
- An Administrator manages entire system.
- An Administrator can add/remove any users any time.
- An Administrator can get analysis
- An Administrator can grant/revoke privileges to user
- An Administrator can provide support to subscribed user

User

- Authorized user can Add/Manage Expenses.
- Authorized user can request to reset password.
- Authorized user can link their bank account.
- Authorized user can get analysis of their Expense.
- Authorized user can Backup/Restore their data.
- Authorized user can manage their account details.

1.4 PROCESS MODEL

Project development approach and justification

The first we have done analysis of the system what are the requirement of the market and what is the market value for this system and how it will be help full to the user.

We have used the “Iterative Waterfall Model” model to implement this system and working and the advantage and the disadvantage of the “Iterative Waterfall Model” model is as the follows.

In the Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and literately enhances the evolving versions until the complete system is implemented and ready to be deployed.

An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.

Iterative Waterfall Model - Design

- Iterative process starts with a simple implementation of a subset of the software requirements and literately enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).
- Iterative and Incremental development is a combination of both iterative design or iterative method and incremental build model for development. "During software development, more than one iteration of the software development cycle may be in progress at the same time." This process may be described as an "evolutionary acquisition" or "incremental build" approach.

The following illustration is a representation of the Iterative and Incremental model

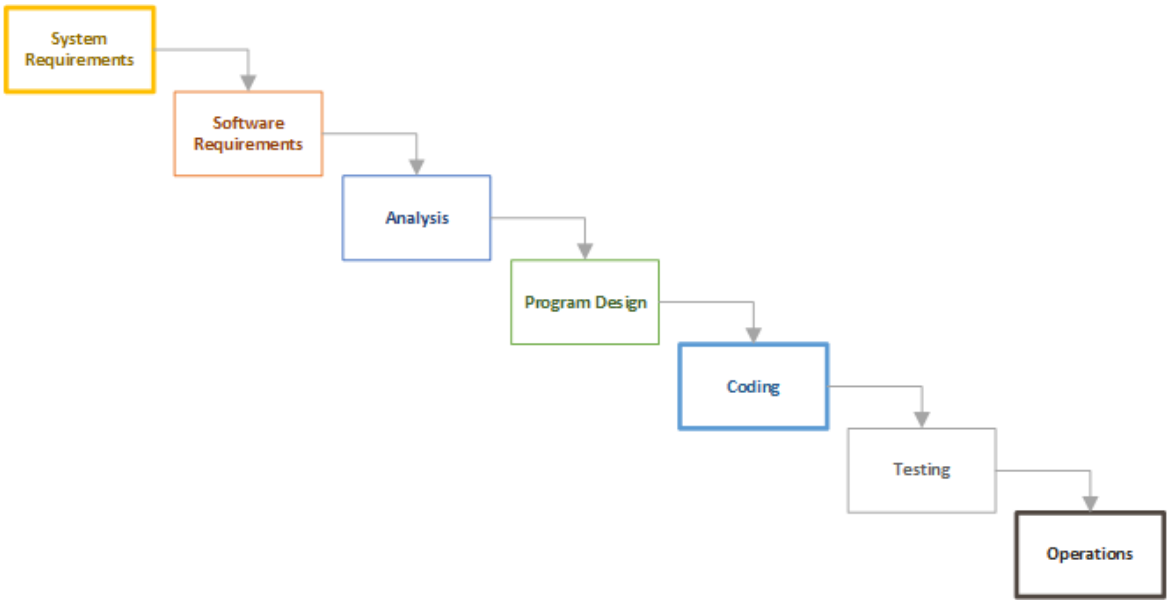

Fig 1.1 : Iterative Waterfall Model

- In this incremental model, the whole requirement is divided into various builds. During each iteration, the development module goes through the requirements, design, implementation and testing phases. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is ready as per the requirement.

- The key to a successful use of an iterative software development life cycle is rigorous validation of requirements, and verification & testing of each version of the software against those requirements within each cycle of the model. As the software evolves through successive cycles, tests must be repeated and extended to verify each version of the software.

Iterative Waterfall Model – Application

Like other SDLC models, Iterative and incremental development has some specific applications in the software industry. This model is most often used in the following scenarios

- Requirements of the complete system are clearly defined and understood.
- Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time.
- There is a time to the market constraint
- A new technology is being used and is being learn by the development team while working on the project.
- Resources with needed skill sets are not available and are planned to be used on contract basis for specific iterations.
- There are some high-risk features and goals which may change in the future.

Iterative Waterfall Model – Pros and Cons

The advantage of this model is that there is a working model of the system at a very early stage of development, which makes it easier to find functional or design flaws. Finding issues at an early stage of development enables to take corrective measures in a limited budget.

The disadvantage with this SDLC model is that it is applicable only to large and bulky software development projects. This is because it is hard to break a small software system into further small serviceable increments/modules.

The advantages of the Iterative and Incremental SDLC Model are as follows

- Some working functionality can be developed quickly and early in the life cycle.
- Results are obtained early and periodically.
- Parallel development can be planned.
- Progress can be measured.
- Less costly to change the scope/requirements.
- Testing and debugging during smaller iteration is easy.
- Risks are identified and resolved during iteration; and each iteration is an easily managed milestone.
- Easier to manage risk - High risk part is done first.
- With every increment, operational product is delivered.
- Issues, challenges and risks identified from each increment can be utilized/applied to the next increment.
- Risk analysis is better.
- It supports changing requirements.

The disadvantages of the Iterative and Incremental SDLC Model are as follows

- More resources may be required.
- Although cost of change is lesser, but it is not very suitable for changing requirements.
- More management attention is required.
- System architecture or design issues may arise because not all requirements are gathered in the beginning of the entire life cycle.
- Defining increments may require definition of the complete system.
- Not suitable for smaller projects.
- Management complexity is more.
- End of project may not be known which is a risk.
- Highly skilled resources are required for risk analysis.

CHAPTER 2

SYSTEM REQUIREMENTS SPECIFICATION

2.0 SYSTEM REQUIREMENTS SPECIFICATION

2.1 USER CHARACTERISTICS

There is one main user in this web portal

- Administrator
- Authorized User
- Guest User

2.1.1 Administrator

An Administrator is user having overall access of website. Administrator has full control over the web portal and manages its front end and back end.

Administrator user can,

- Create or add other user
- Delete or remove other user
- Manage user profile
- Give privileges to other user
- Temporary block user to assess website
- Administrator sends feedback and important updates notification to authorized user

2.1.2 Authorized User

Authorized user is the user who actually uses this website. They can access this website by logging into the website using their username and password.

Authorized user uses the all the functionalists provide in website. Authorized user can't add another user. Authorized user has limited privileges provided by an administrator.

User can add their expanses they can update expanses any time. User can get analysis of how much money they pay for what purpose and where they spent it. User can export their expanses reports based on time, location, category etc.

Authorized user can,

- Create their account
- Delete their account
- Add expense
- Modify expense
- Set its budget limit
- Add daily common expense
- Connect their bank account
- Get analysis of their expense
- Export expense reports
- Backup or Restore their data
- Reset it's password if they forgot

2.1.3 Guest User

Guest user has similar functionalities same as authorized user. Guest user doesn't need to create their account to use functionalities of website.

Guest user has limited functionalities and has minimal privileges. Guest user can only access full features of website after finishing authorization process.

Guest User can,

- Add expense
- Set its budget limit
- Get analysis of their expense

2.2 FUNCTIONAL REQUIREMENTS

A functional requirement (FR) defines statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.

Functional requirements for the proposed system as follow,

1 Register User

This operation can be done by any website visitor. Visitor need to enter their public details like name, email and personal details like password, mobile no. to create their account.

1.1 Check If User Already Exist

Input: User details

Output: Acknowledgement to user

Process: Check database

1.2 Register User

Input: User Details

Output: Acknowledgement to user

Process: Store details in database

2 Login

To use this website authorized user need to enter details like his/her username and password to logging into website.

Input: User details

Output: Authorized user can use functionalities of the website

Process: Check username, password in database

3 Guest Login

Guest user can use some functionalists of the website without providing its personal details and for limited time period.

Input: Enter name

Output: User can use some functionalists of website

Process: Store details in database

4 Set Defaults

Authorized user can set their weekly or monthly budget limit for getting accurate expense records. They can chose category to add them latter.

Input: Enter budget, time period and select category

Output: Acknowledgement to user

Process: Store details in database

5 Daily Common Expense

Authorized user can set expenses which they pay in every day life. So they don't need to add them regularly.

Input: Enter expense title, money, location, category, comments

Output: Acknowledgement to user

Process: Add records in database daily

6 Expense

User can add expense how much they pay at where and for what purpose by choosing category and adding comments.

User can modify expense record by selecting them. User can also delete or hide their expense record for privacy concern.

6.1 Add Expense

Input: Enter Expense title, money, location, category, comments

Output: Acknowledgement to user

Process: Store details in database

6.2 Update Expense

Input: Enter new details

Output: Acknowledgement to user

Process: Update records in database

6.3 Delete Expense

Input: Select Expense

Output: Acknowledgement to user

Process: Delete expense record from database

6.4 Hide Expense

Input: Select Expense

Output: Acknowledgement to user

Process: Change expense status to hide mode and update database

7 Link Bank Account

This functionality provides authorized user to automatically fetch transaction records from bank in server's database made by online payments or by Credit/Debit card. By doing this authorized user does not required to add them manually.

Input: Enter bank Details

Output: Acknowledgement to user

Process: Securely link users bank account to website

8 Analysis

This function provides analysis of users expenses by displaying appropriate graphs. This feature can predict how much more money they spend in future by calculating transaction made in past.

Input: Chose display type like location, time, category etc.

Output: Display Graph and statistic reports

Process: Fetch details from database

9 Export Reports

Authorized user can export their generated expense reports in standard format(s).

Input: Select report extension like PDF, CSV, XLX

Output: Download reports

Process: Generate report and convert it into specific format

10 Backup And Restore

Authorized user can backup and restore their data to the website server.

10.1 Backup Data

Input: Trigger backup button

Output: Acknowledgement to user

Process: Backup whole data to the server

10.2 Restore Data

Input: Select back up data

Output: Acknowledgement to user

Process: Update user database and replace all data with selected back up data

11 Reset Password

Authorized user can reset their account password for security reason or if they forgot it.

11.1 Send OTP

Input: Select email / mobile number to receive OTP

Output: Send OTP

Process: Generate OTP and sends to the user's selected method like email or mobile

11.2 Reset/Update Password

Input: Enter new password

Output: change user password if they enter right OTP

Process: If OTP is right than change user password and update password in database

2.3 NON-FUNCTIONAL REQUIREMENTS

A non-functional requirement is constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, security etc.

Non-Functional requirements for the web portal as follow,

1. Performance Requirement

- The Data should be store in database within 2 seconds
- Query result should be return within 2 second approx.
- Web portal load time should not take more than 10 seconds
- Login validation should be done in 4 seconds
- Online transaction should be updated in database as soon as it done
- Graphs should not be generate wrong reports

2. Security Requirement

- All the communication between client and server should be fully encrypted
- All the transaction records from the bank should be securely transmitted
- Password should be encrypted in database
- OTP is not valid more than 15 minutes

3. Safety Requirement

- In case of emergency all the database should be backup in frequent time
- Proper SSL should be installed in the server
- Frequently scan for the virus into the server

4. System Requirement

- **Portable**
  - ✧ Web portal has responsive design along desktop, tablet and mobile
  - ✧ Web portal should be opened in all latest version of the browsers
- **Usability**
  - ✧ Web portal should be easy to learn by new user
  - ✧ It should having user friendly interface
- **Maintainability**
  - ✧ For development it should easy to integrate new updates to system
  - ✧ Maintenance should be cost effective and easy
- **Reliability**
  - ✧ Product reliability is measured in terms of working of project under different working environment and different conditions
  - ✧ For this Web portal it should not be down in normal circumstances rather than heavy traffic
- **Integrity**
  - ✧ Product integrity can be measured in security term that hacker can't break the server.
  - ✧ Only authorized user can access their data
  - ✧ Only right user can make certain changes - eg only administrator can remove user

CHAPTER 3

SYSTEM ANALYSIS MODELING – USER-BASED

3.0 SYSTEM ANALYSIS MODELING – USER-BASED

3.1 FEASIBILITY STUDY OF THE NEW SYSTEM

Feasibility study is carried out whenever there is a complex problem or opportunity, it is a fact a preliminary investigation which emphases the "Lock under taken to determine the Possibility or probability of either improving the existing system or developing a completely new system.

It helps to obtain an overview of the problem and to get rough assessment of whether feasible solutions exist. There is essential to avoid committing large resources to a project and the repent on it later.

We started the project by listing the possible queries, which need to be addressed when it comes to customer satisfaction and on this lines we have surged ahead on the project.

The three main points, kept in mind at the time of project are :

- Possible (to build it with the given technology and resources)
- Affordable (followed by given time and cost constraints of the organization
- Acceptable (for use by the eventual users of the system)

Technical Feasibility

The System will be implemented using the current technology and within the given cost and schedule constraint.

We have used the PHP language, MySQL database and CSS framework to implement the system.

Economic Feasibility

Yes, the system will contribute the overall objecting of the organization we can satisfy all the requirement of the user.

It will also save the time and cost of the organization.

3.2 USER-BASED MODELING

3.2.1 Use Case Diagrams

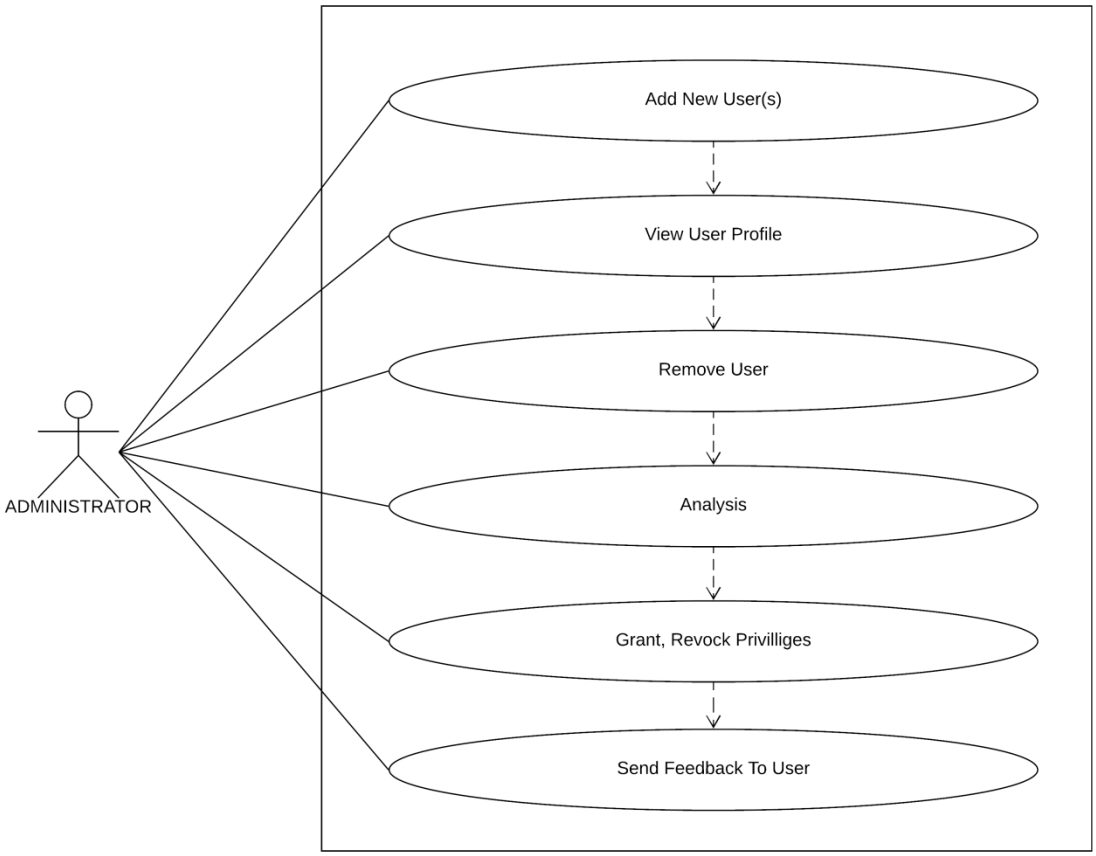

Fig 3.1 : Use case Diagram - Administrator

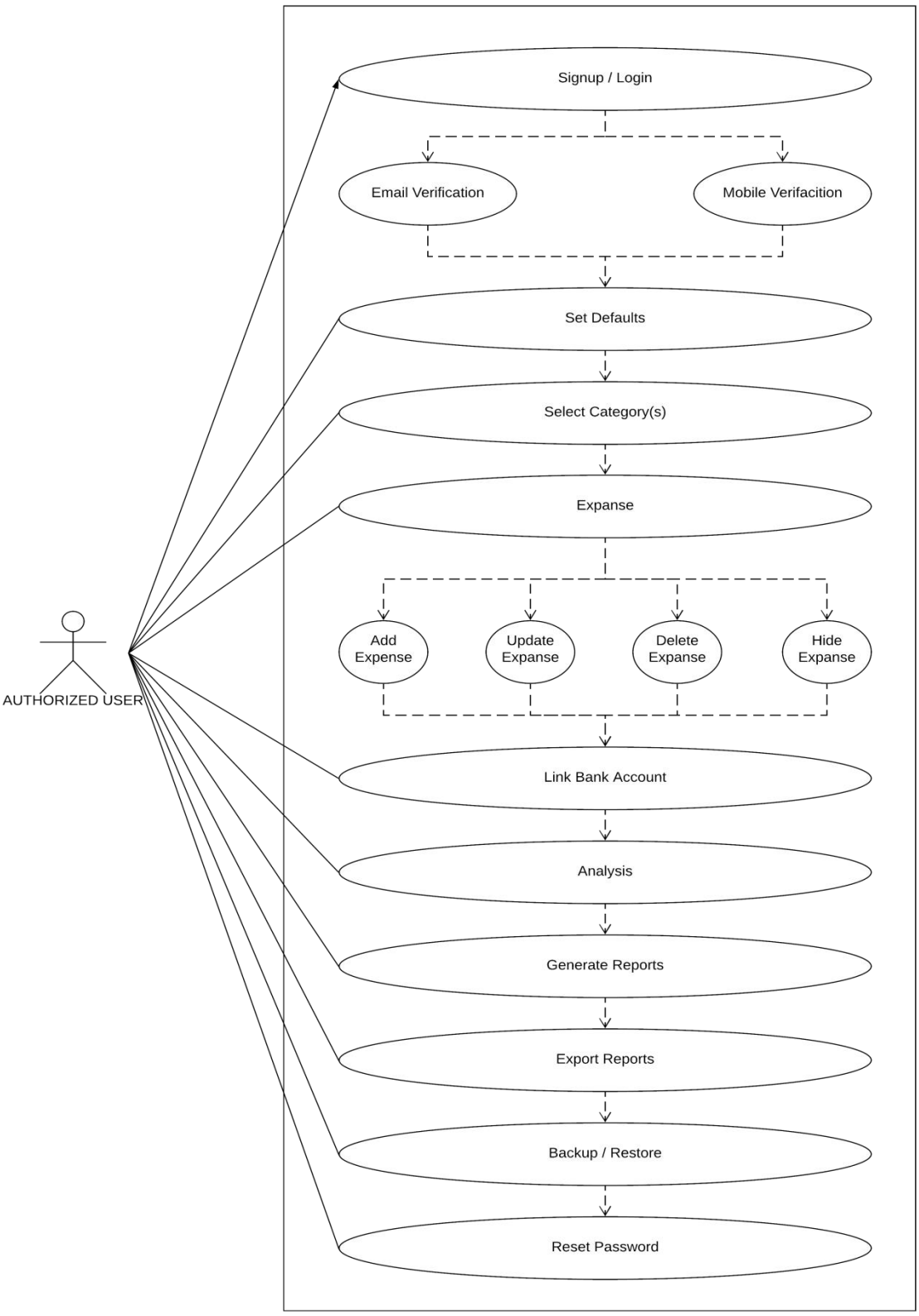

Fig 3.2 : Use case Diagram - Authorized user

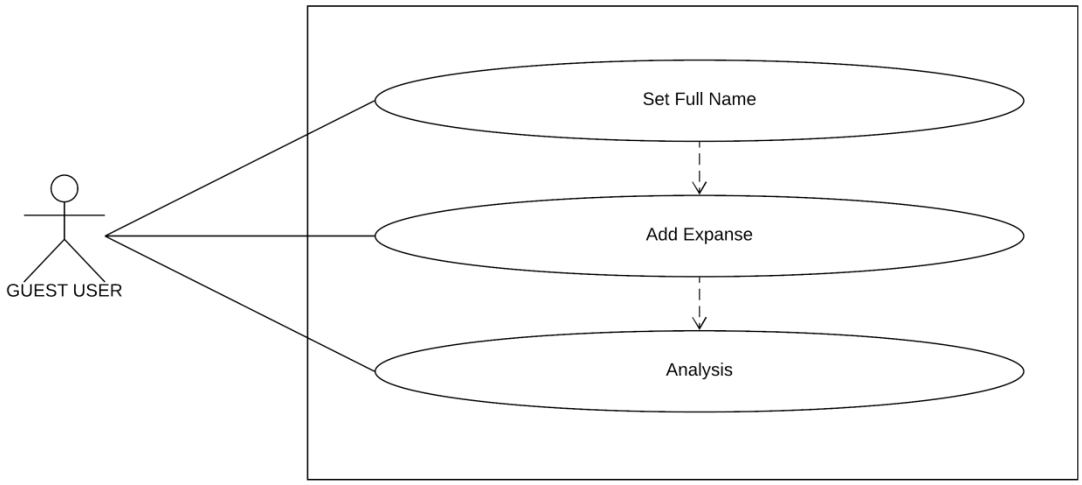

Fig 3.3 : Use case Diagram - Guest user

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN – DATA-BASED

4.0 SYSTEM ANALYSIS AND DESIGN – DATA-BASED

4.1 DATA MODELING

4.1.1 Data Dictionary

Data dictionaries store and communicate meta data about data in a database, a system, or data used by website

Data dictionaries contains table information like column name, data type, size, constraint etc.

There are two types of data dictionaries

- Active Data Dictionaries
- Passive Data Dictionaries

The data dictionary is automatically updated by the database management system when any changes are made in the database. This is known as an active data dictionary as it is self updating.

The passive data dictionary has to be manually updated to match the database. This needs careful handling or else the database and data dictionary are out of sync.

ET_USER					
Column Name	**Data type**	**Size**	**Constraints**	**Description**	**Example**
Uid	INT	8	PRIMARY KEY	Unique id of each user	00001234
Username	VARCHAR	16	PRIMARY KEY	Username of users	Someown
Email	VARCHAR	30	NOT NULL	Email of user	someown@example.com
No	INT	10	NOT NULL	Mobile no. Of user	1111222278

ET_PROFILE					
Column Name	**Data type**	**Size**	**Constraints**	**Description**	**Example**
username	VARCHAR	16	PRIMARY KEY	Use as Foreign key	Someown
Fname	VARCHAR	20	NOT NULL	First name of user	John
Lname	VARCHAR	20	NOT NULL	Last name of user	Doe
gender	BINARY	1	NOT NULL	0 - male 1- female	0
City	VARCHAR	30	NOT NULL	City of user	Rajkot
country	VARCHAR	30	NOT NULL	Country of user	India

ET_PASSWORD					
Column Name	**Data type**	**Size**	**Constraints**	**Description**	**Example**
username	VARCHAR	16	PRIMARY KEY	Use as foreign key	Someown
P1	VARCHAR	256	-	Old password	-
P2	VARCHAR	256	-	Old password	-
P3	VARCHAR	256	NOT NULL	Current Password	-

ET_DATA					
Column Name	**Data type**	**Size**	**Constraints**	**Description**	**Example**
did	VARCHAR	4	PRIMARY KEY	Unique id of data	0000
username	VARCHAR	16	PRIMARY KEY	Use as foreign key	Someown
time	DATETIME	-	NOT NULL	Time Stamp	-

ET_CATEGORY					
Column Name	**Data type**	**Size**	**Constraints**	**Description**	**Example**
cid	INT	2	PRIMARY KEY	Unique id for category	00
username	VARCHAR	16	NOT NULL	Use as foreign key	Someown
Title	VARCHAR	30	NOT NULL	Name of category	Transport
parent	INT	2	-	Id of parent category	18

ET_FLAG					
Column Name	**Data type**	**Size**	**Constraints**	**Description**	**Example**
username	VARCHAR	16	PRIMARY KEY	Use as foreign key	Someown
Guest	BINARY	1	NOT NULL	1 if user is Guest User	0
status	BINARY	1	NOT NULL	0 - offline 1 - online	1
Link	BINARY	1	NOT NULL	Bank account	1

ET_DEFAULT					
Column Name	**Data type**	**Size**	**Constraints**	**Description**	**Example**
did	VARCHAR	4	PRIMARY KEY	Unique id	0000
username	VARCHAR	16	NOT NULL	Use as foreign key	Someown
budget	INT	10	NOT NULL	Budget limit	10000
day	INT	3	NOT NULL	Total day for budget limit	30

ET_EXPENSE					
Column Name	**Data type**	**Size**	**Constraints**	**Description**	**Example**
eid	VARCHAR	4	PRIMARY KEY	Unique id of expanse	0000
username	VARCHAR	16	FORIGN KEY	Use as foreign key	Someown
title	VARCHAR	30	NOT NULL	Title for expense	Pizza
money	INT	10	NOT NULL	Total money paid	700
location	VARCHAR	30	NOT NULL	Location where it paid	somewhere
time	DATETIME	-	NOT NULL	Time when paid	-
category	INT	4	NOT NULL	Category id	18
comment	VARCHAR	128	-	Comments	Pizza party
daily	BINARY	1	NOT NULL	0 - Normal 1- Pay daily	0

ET_DATE					
Column Name	**Data type**	**Size**	**Constraints**	**Description**	**Example**
username	VARCHAR	16	PRIMARY KEY	Use as foreign key	Someown
signup	DATETIME	-	NOT NULL	User Signup date	-
login	DATETIME	-	NOT NULL	Last login time	-
backup	DATETIME	-	-	Last backup time	-
restore	DATETIME	-	-	Last restore time	-
trans	DATETIME	-	-	Last transaction	-

ET_BANK					
Column Name	**Data type**	**Size**	**Constraints**	**Description**	**Example**
username	VARCHAR	16	PRIMARY KEY	Use as foreign key	Someown
card	INT	16	NOT NULL	Card number	-
date	DATETIME	-	NOT NULL	Expiry date	08 22
cvv	INT	3	NOT NULL	CVV	232

4.1.2 E-R (Entity-Relationship) Diagram

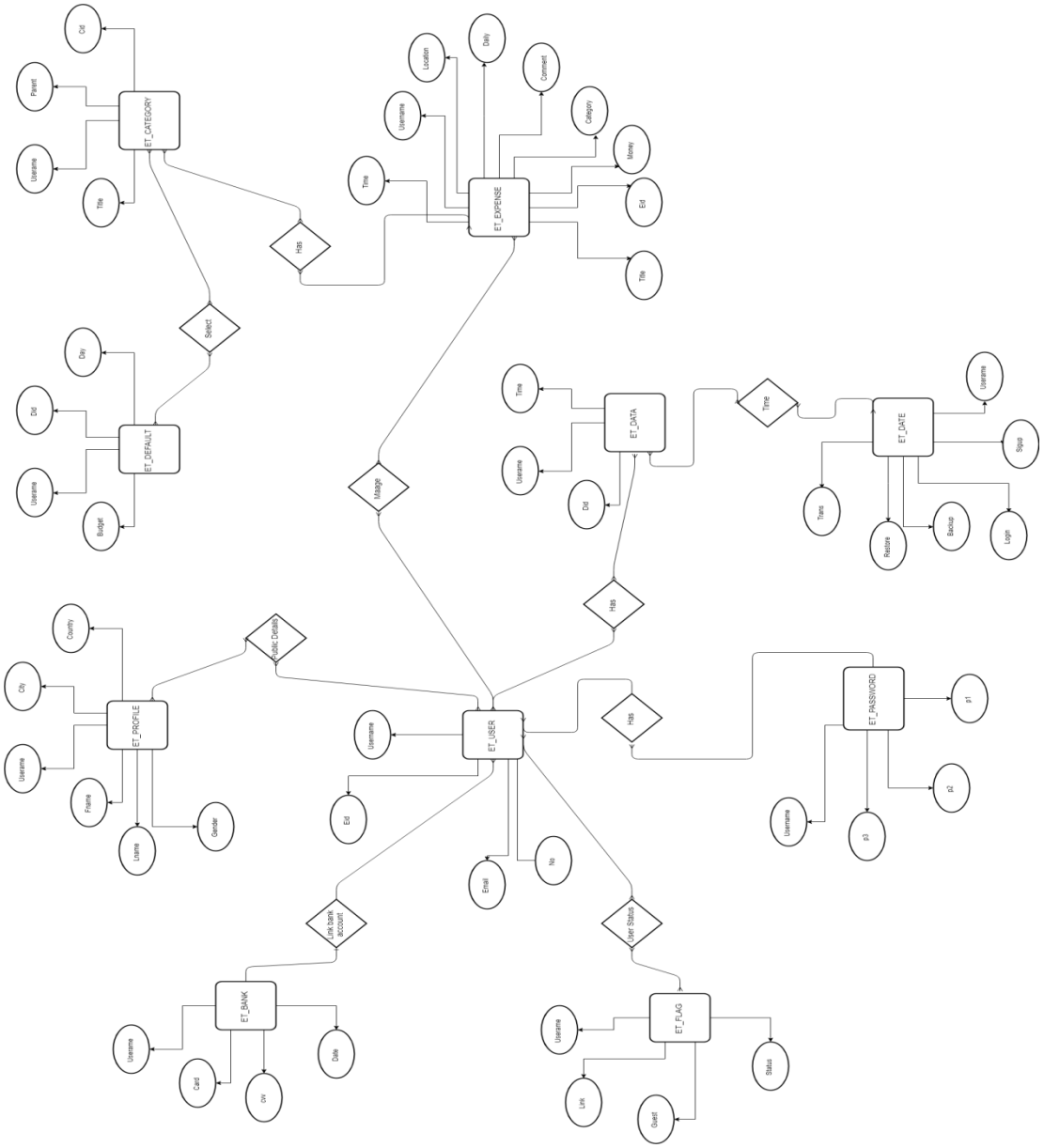

Fig 4.1 : ER Diagram

4.2 BEHAVIORAL MODELING

4.2.1 Data Flow Diagram

4.2.1.1 Context Level Diagram (Level 0)

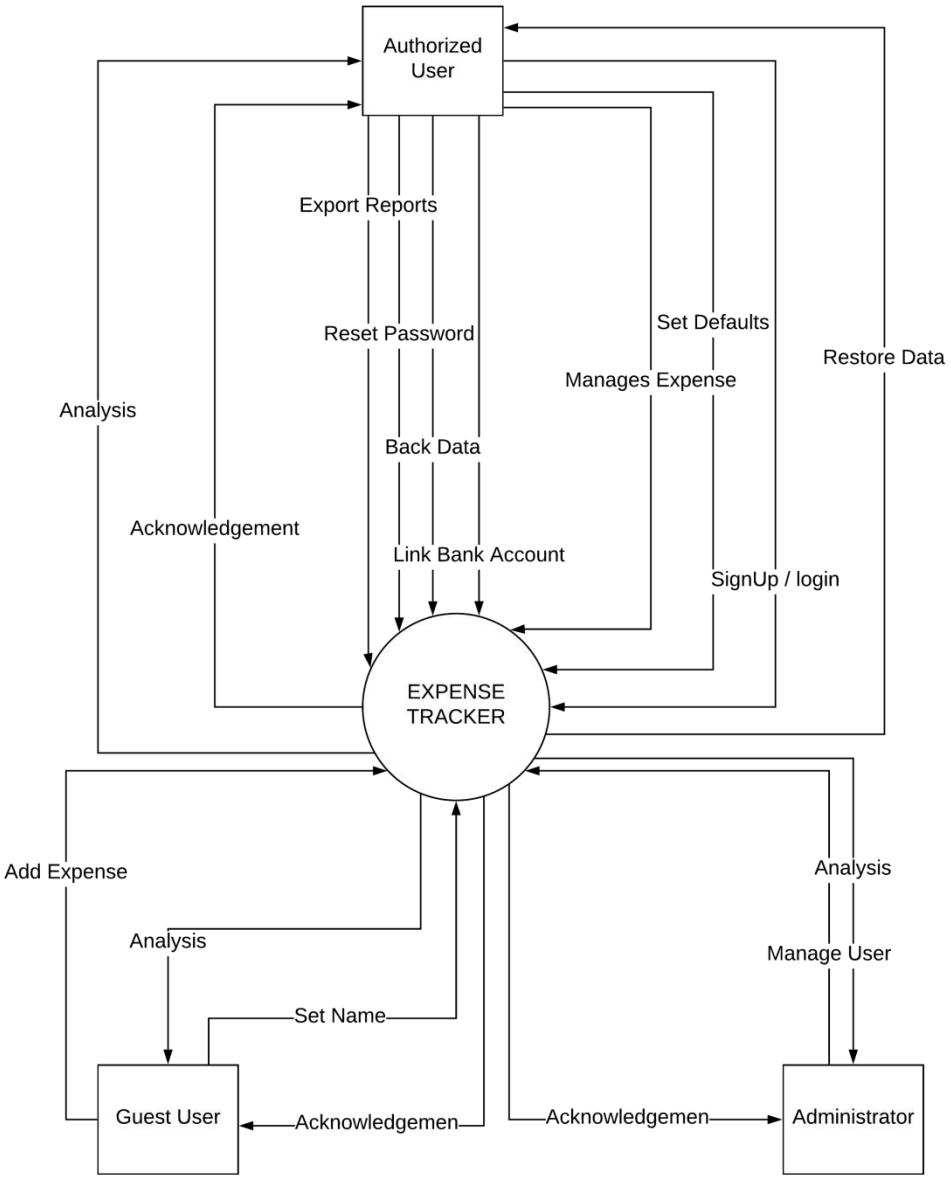

Fig 4.2 :DFD Level

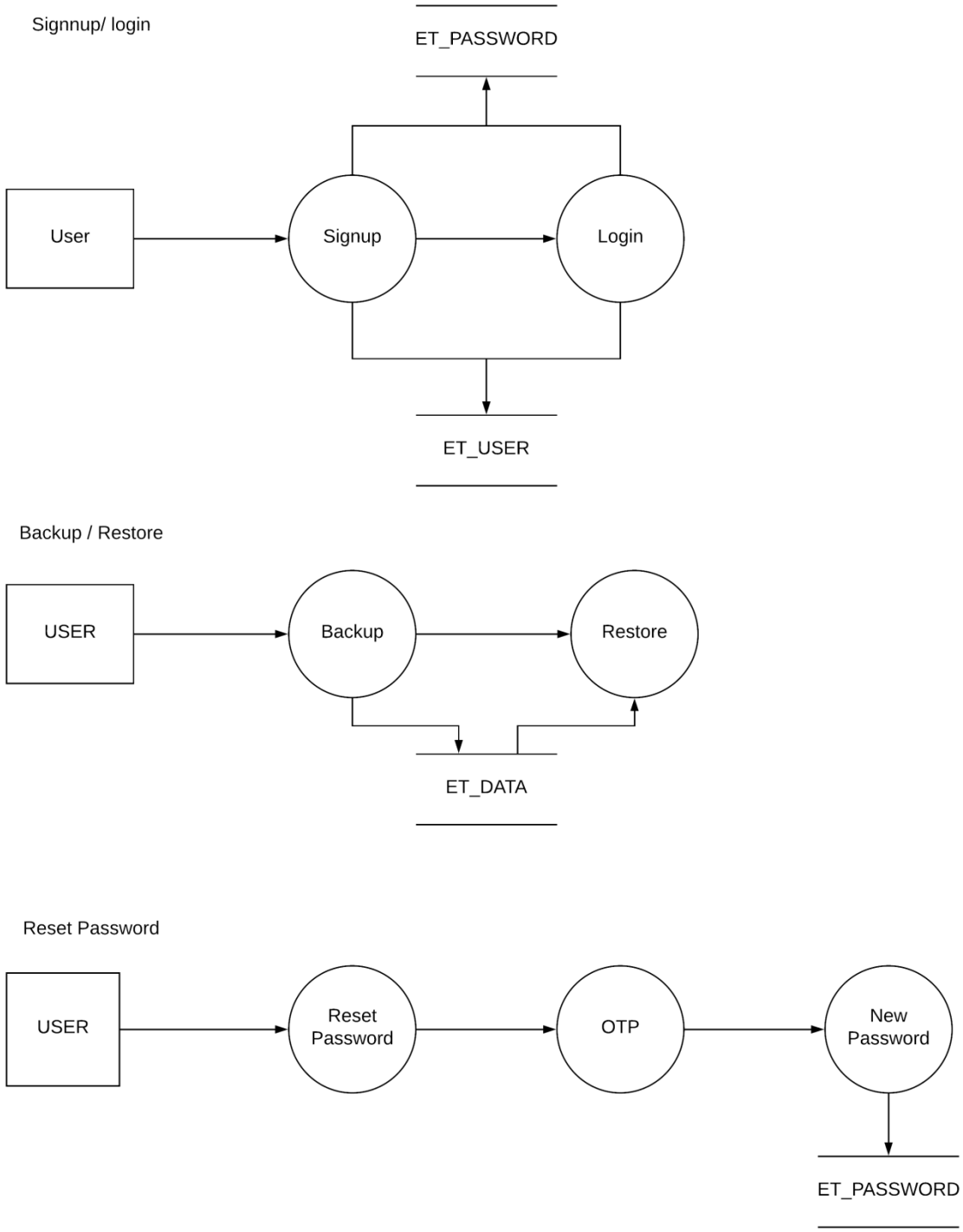

Fig 4.3 :DFD Level 1

CHAPTER 5

SYSTEM DESIGN – UML

5.0 SYSTEM DESIGN – UML

5.1 SEQUENCE DIAGRAMS

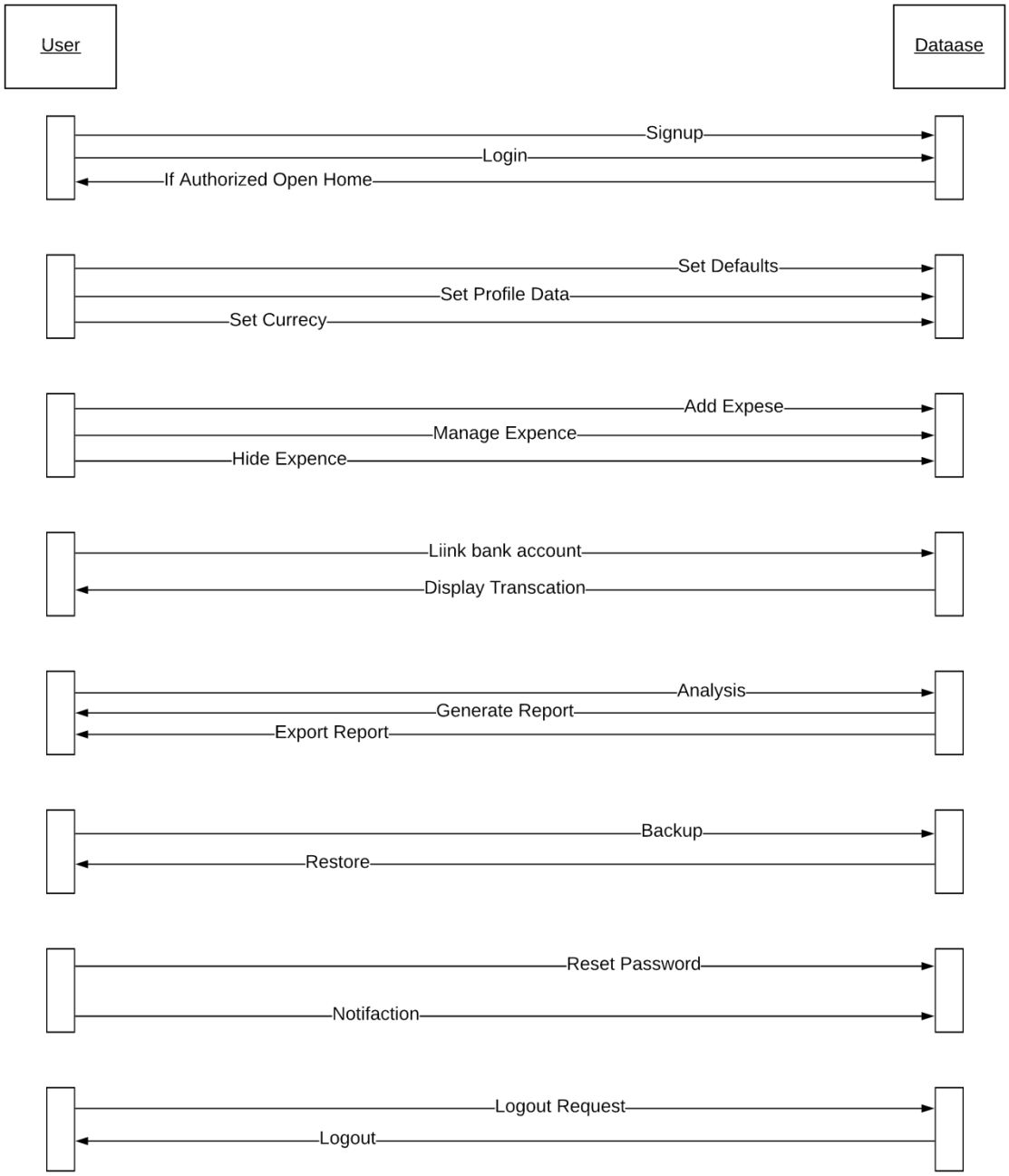

Fig 5.1 : Sequence Diagram

5.2 ACTIVITY DIAGRAMS

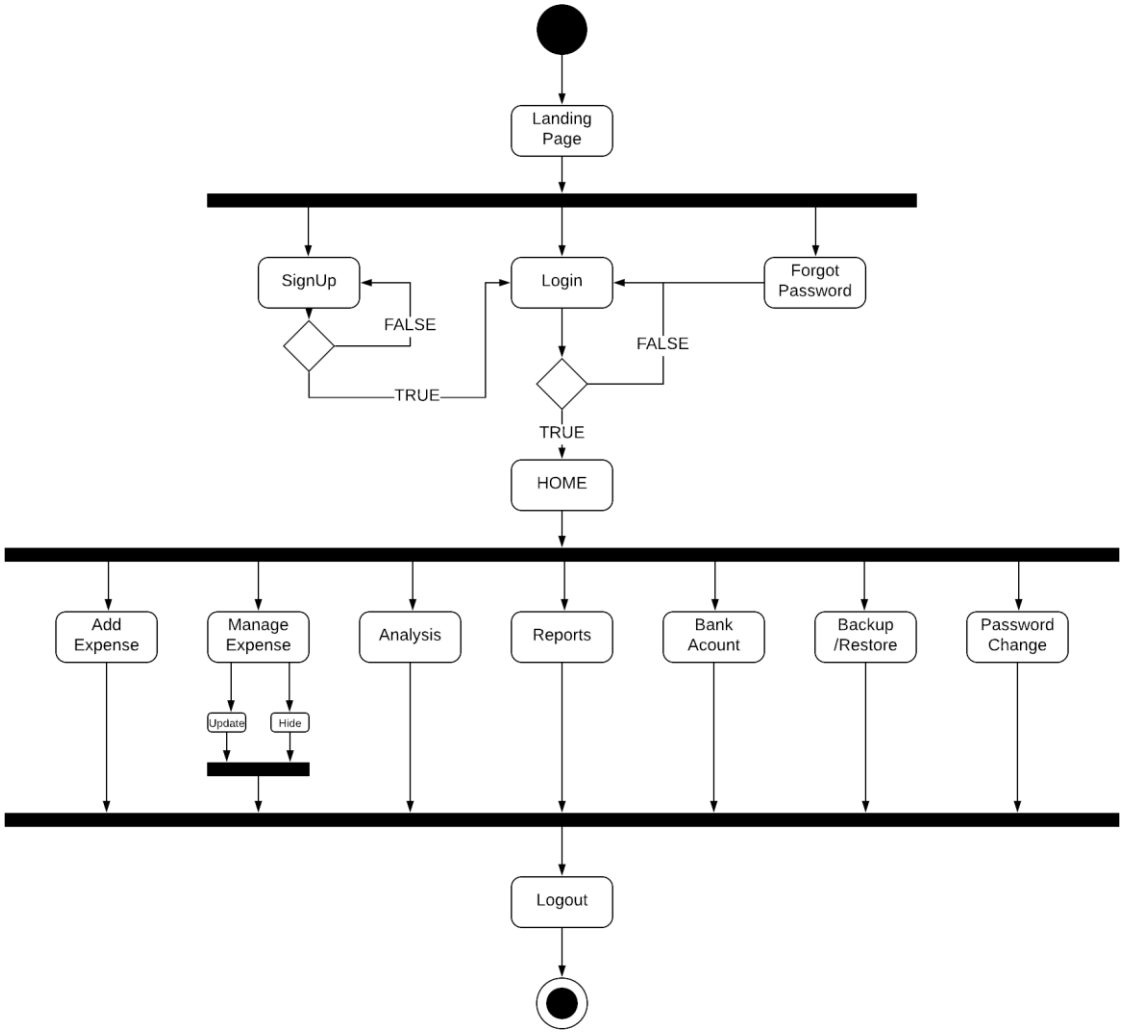

Fig 5.2 : Activity Diagrams

CHAPTER 6

SYSTEM INTERFACE DESIGN

6.0 SYSTEM INTERFACE DESIGN

6.1 INPUT- OUTPUT FORMS DESIGN

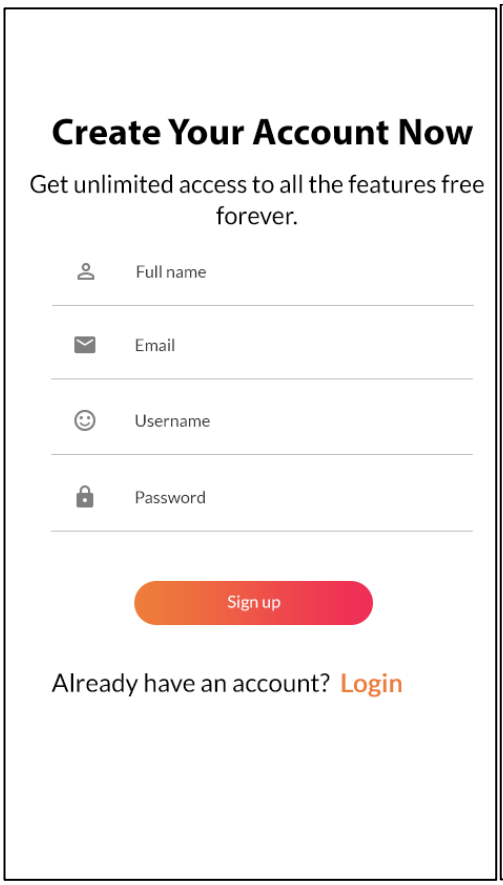

fig 6.1 : Input output – sigup

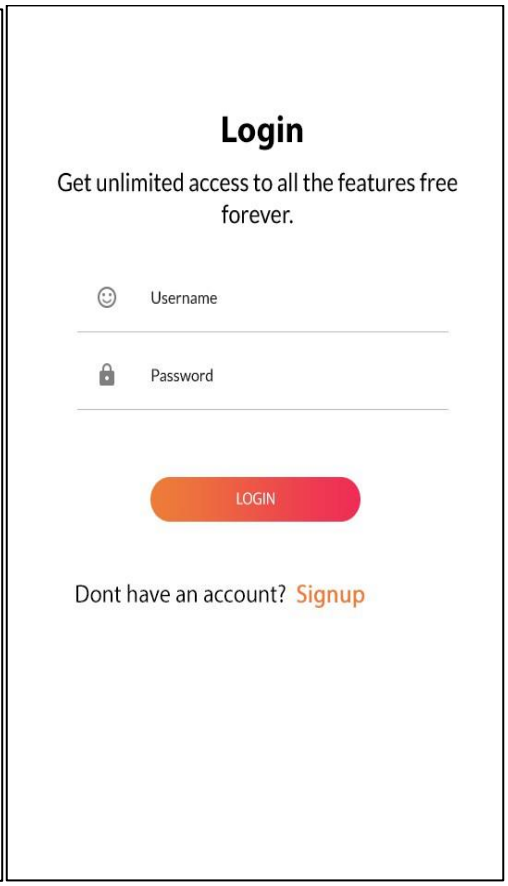

fig 6.2 : Input output - login

6.2 GRAPHICAL USER INTERFACE DESIGN

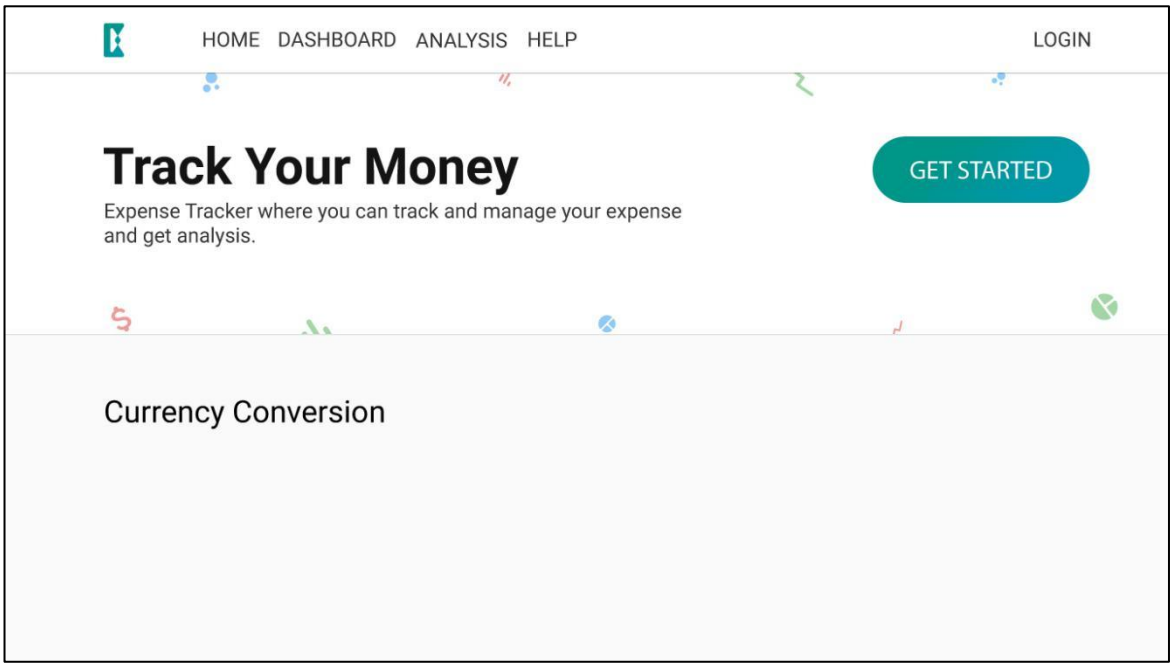

Figure 6.3 GUI Home

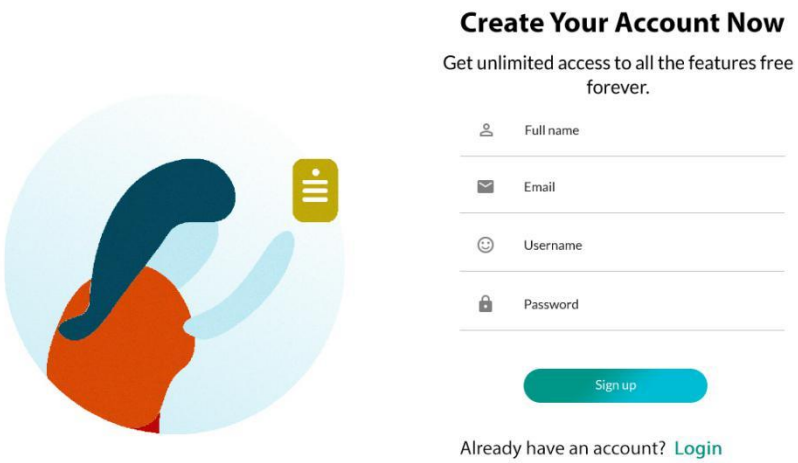

Figure 6.4 GUI Sign up

7 REFERENCE

1. https://google.com
2. https://fonts.google.com
3. https://fontawesome.com
4. https://github.com
5. https://uplabs.com

8 BIBLIOGRAPHY

1. Fundamental of software engineering by Rajib Mall
2. Web Design with HTML, CSS, Java-script and Jquery set by John Duckett
3. SQL, PL/SQL by Ivan Bayross