Personal Expense Tracker Application

IBM-Project-325-1658287565

PERSONAL EXPENSE TRACKER APPLICATION

NALAIYA THIRAN PROJECT BASED LEARNING ON PROFESSIONAL READLINESS FOR INNOVATION, EMPLOYNMENT AND ENTERPRENEURSHIP

A PROJECT REPORT

WINSON DASS G (710019104303)

ABIRAMI S (710019104002)

RAMANAN N (710019104034)

SADHANA N (710019104038)

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

ANNA UNIVERSITY REGIONAL CAMPUS COIMBATORE
COIMBATORE – 641 046

INDEX

1. **INTRODUCTION**
    a. Project Overview
    b. Purpose
2. **LITERATURE SURVEY**
    a. Existing problem
    b. References
    c. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    a. Empathy Map Canvas
    b. Ideation & Brainstorming
    c. Proposed Solution
    d. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    a. Functional requirement
    b. Non-Functional requirements
5. **PROJECT DESIGN**
    a. Data Flow Diagrams
    b. Solution & Technical Architecture
    c. User Stories
6. **PROJECT PLANNING & SCHEDULING**
    a. Sprint Planning & Estimation
    b. Sprint Delivery Schedule

    c. Reports from JIRA

7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    a. Feature 1
    b. Feature 2
    c. Database Schema (if Applicable)
8. **TESTING**
    a. Test Cases
    b. User Acceptance Testing
9. **RESULTS**
    a. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

Source Code

GitHub & Project Demo Link

1.INTRODUCTION

1.1 Project Overview

Category: Cloud App Development

Team ID : PNT2022TMID42234

💻 **Skills Required:**

IBM Cloud,HTML,Javascript,IBM Cloud Object Storage,Python-Flask,Kubernetes,Docker,IBM DB2,IBM Container Registry

Project Description:

In simple words, personal finance entails all the financial decisions and activities that a Finance app makes your life easier by helping you to manage your finances efficiently. A personal finance app will not only help you with budgeting and accounting but also give you helpful insights about money management.

Personal finance applications will ask users to add their expenses and based on their expenses wallet balance will be updated which will be visible to the user. Also, users can get an analysis of their expenditure in graphical forms. They have an option to set a limit for the amount to be used for that particular month if the limit is exceeded the user will be notified with an email alert.

1.2 Purpose

Personal finance management is an important part of people’s lives. However, everyone does not have the knowledge or time to manage their finances in a proper manner. And, even if a person has time and knowledge, they do not bother with tracking their expenses as they find it tedious and time-consuming. Now, you don’t have to worry about managing your expenses, as you can get access to an expense tracker that will help in the active management of your finances.Also known as expense manager and money manager, an expense tracker is a software or application that helps to keep an accurate record of your money inflow and outflow. Many people in India live on a fixed income, and they find that towards the end of the month they don’t have sufficient money to meet their needs. While this problem can arise due to low salary, invariably it is due to poor money management skills.

People tend to overspend without realizing, and this can prove to be disastrous. Using a daily expense manager can help you keep track of how much you spend every day and on what. At the end of the month, you will have a clear picture where your money is going. This is one of the best ways to get your expenses under control and bring some semblance of order to your finances.Today, there are several expense manager applications in the market. Some are paid managers while others are free. Even banks like ICICI offer their customers expense tracker to help them out. Before you decide to go in for a money manager, it is important to decide the type you want.

2.LITERATURE SURVEY

2.1 Existing problem

In a study conducted by Forrester in 2016 surveying small and medium businesses (SMBs) across the world, 56% companies reported expense management as being the biggest challenge for their finance departments.

In another survey conducted by Levvel Research in 2018 in North America, respondents reported the following pain points in expense management before adopting automation:

1. Manual entry and routing of expense reports (62%)
2. Lack of visibility into spend data (42%)
3. Inability to enforce travel policies (29%)
4. Lost expense reports (24%)
5. Lengthy expense approval system and reimbursement cycles (23%)

References

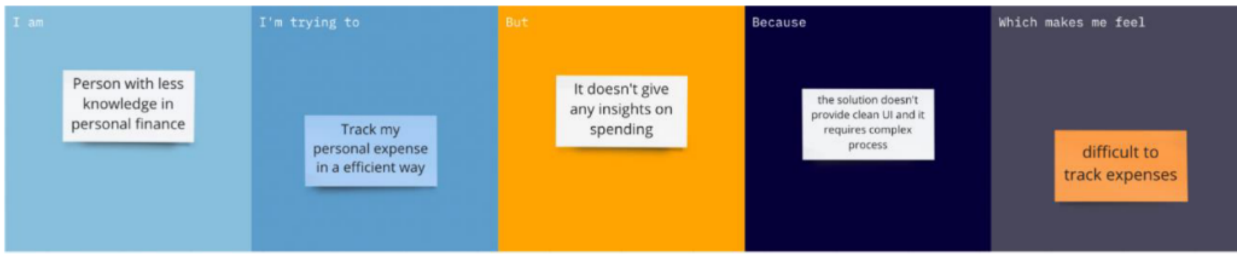
https://www.thebalancemoney.com/is-it-important-to-track-my-expenses-2385679
https://www.icicidirect.com/investonomics/find-your-mojo/articles/surprising-benefits-of-tracking-your-expenses

Problem Statement Defintion

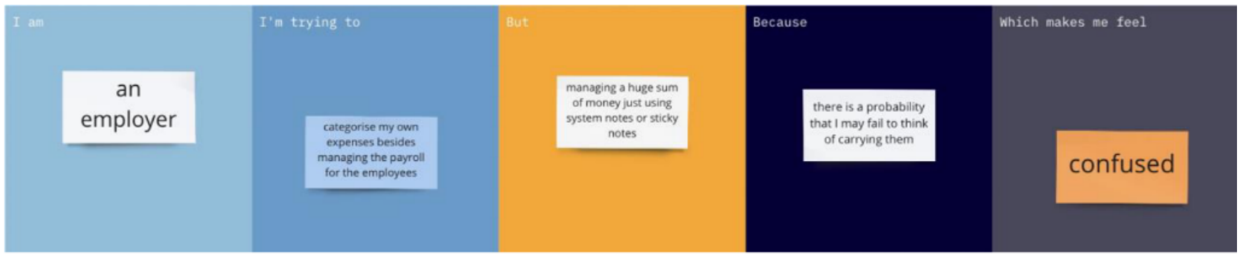
Customer Problem Statement :

A well-articulated customer problem statement allows us to find the ideal solution for the challenges our customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

PS 1

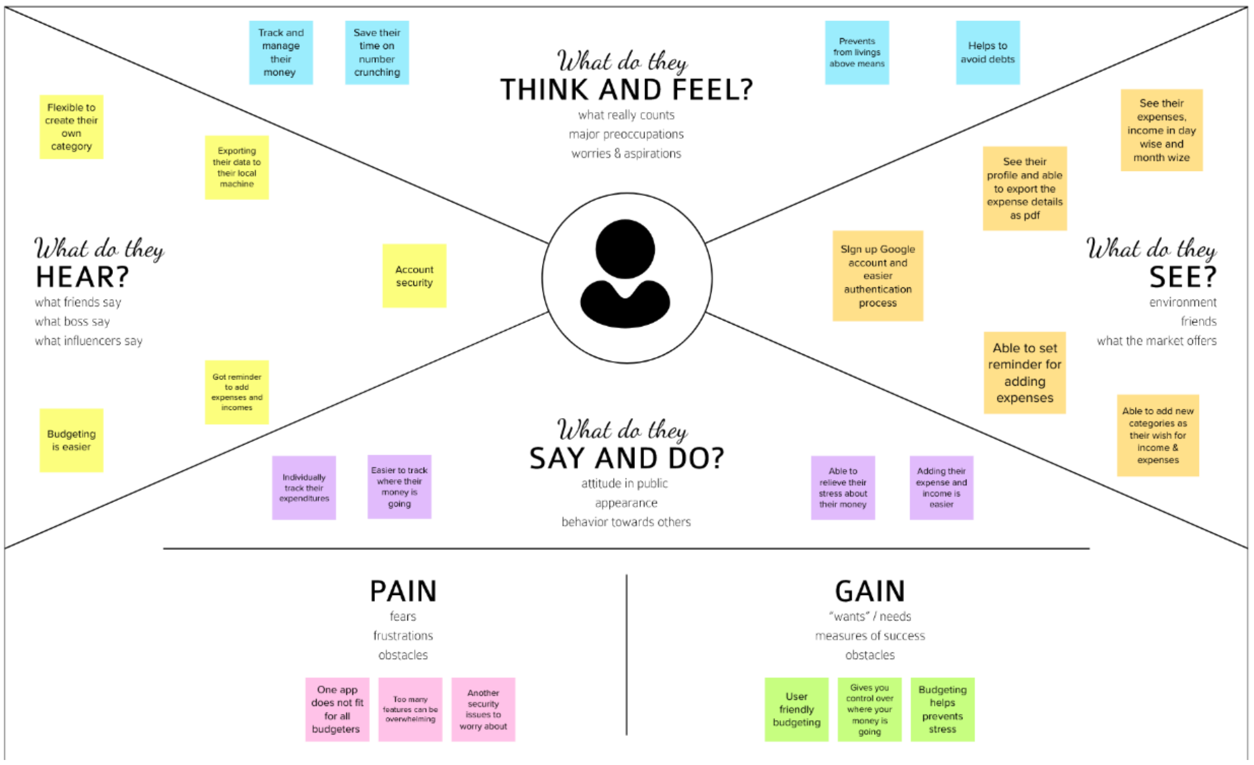

I am
Person with less knowledge in personal finance
I'm trying to
Track my personal expense in a efficient way
But
It doesn't give any insights on spending
Because
the solution doesn't provide clean UI and it requires complex process
Which makes me feel
difficult to track expenses

PS 2


I am
an employer
I'm trying to
categorise my own expenses besides managing the payroll for the employees
But
managing a huge sum of money just using system notes or sticky notes
Because
there is a probability that I may fail to think of carrying them
Which makes me feel
confused

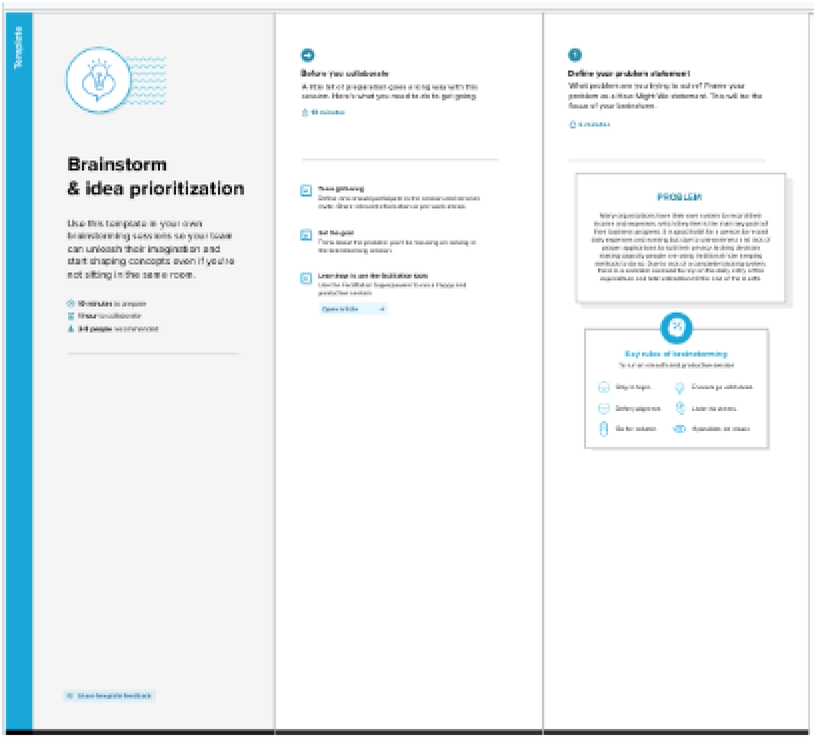
3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas


What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Track and manage their money
Save their time on number crunching
Prevents from livings above means
Helps to avoid debts
What do they HEAR?
what friends say
what boss say
what influencers say
Flexible to create their own category
Exporting their data to their local machine
Account security
Got reminder to add expenses and incomes
Budgeting is easier
What do they SEE?
environment
friends
what the market offers
See their expenses, income in day wise and month wize
See their profile and able to export the expense details as pdf
Sign up Google account and easier authentication process
Able to set reminder for adding expenses
Able to add new categories as their wish for income & expenses
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Individually track their expenditures
Easier to track where their money is going
Able to relieve their stress about their money
Adding their expense and income is easier
PAIN
fears
frustrations
obstacles
One app does not fit for all budgeters
Too many features can be overwhelming
Another security issues to worry about
GAIN
"wants" / needs
measures of success
obstacles
User friendly budgeting
Gives you control over where your money is going
Budgeting helps prevents stress

3.2 Idea on & Brainstorming

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Template

Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Before you collaborate
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

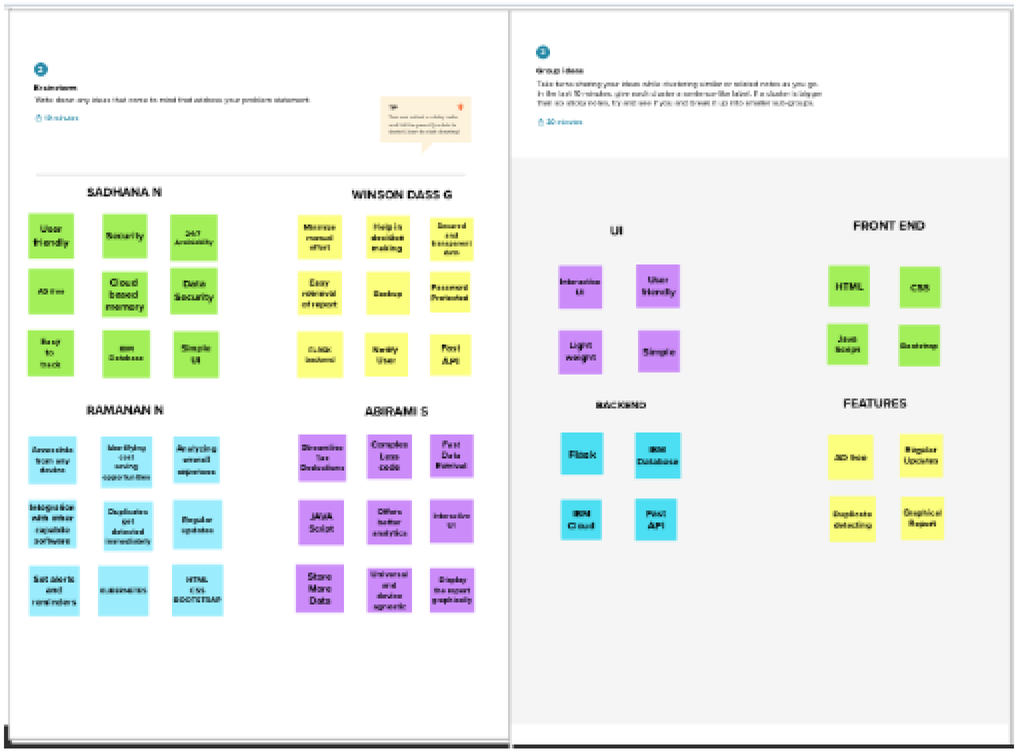
PROBLEM

[illegible]

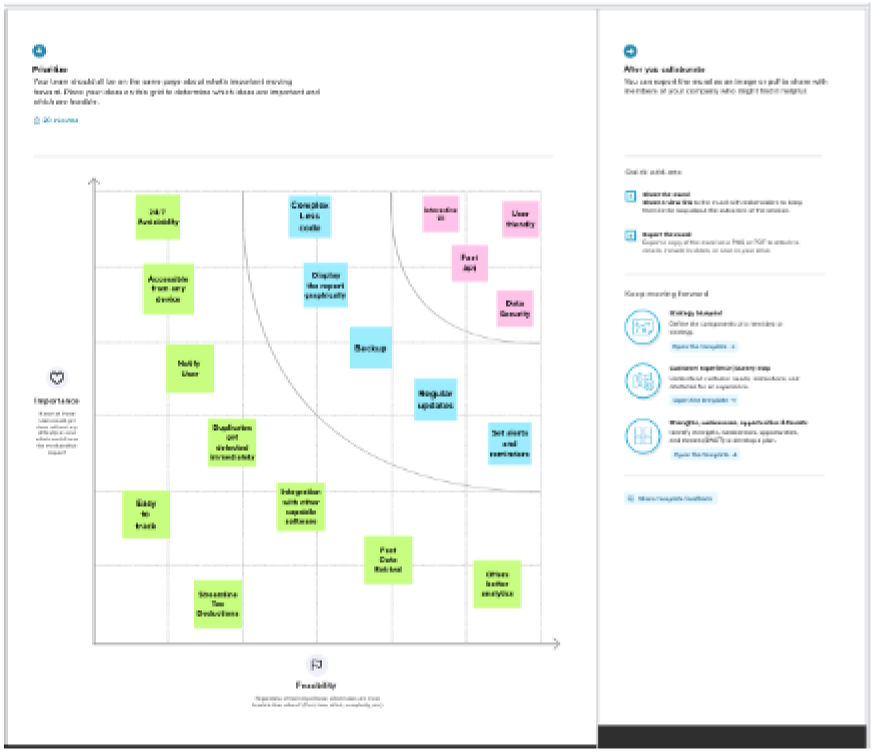
Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

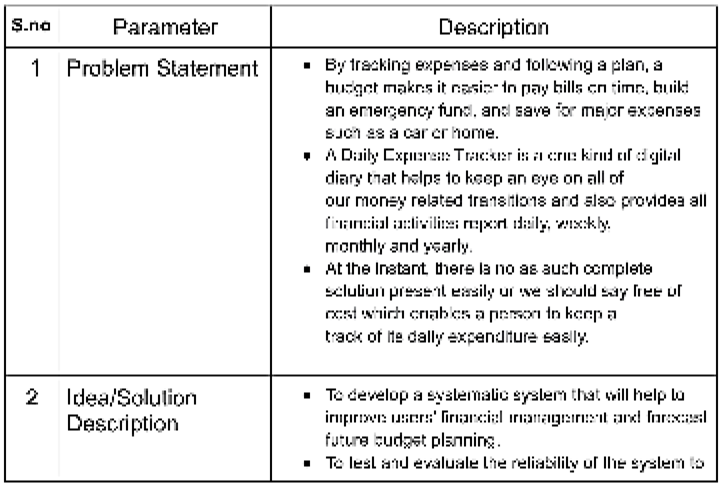
Step-2 : Brainstorm,Idea Listing and Grouping

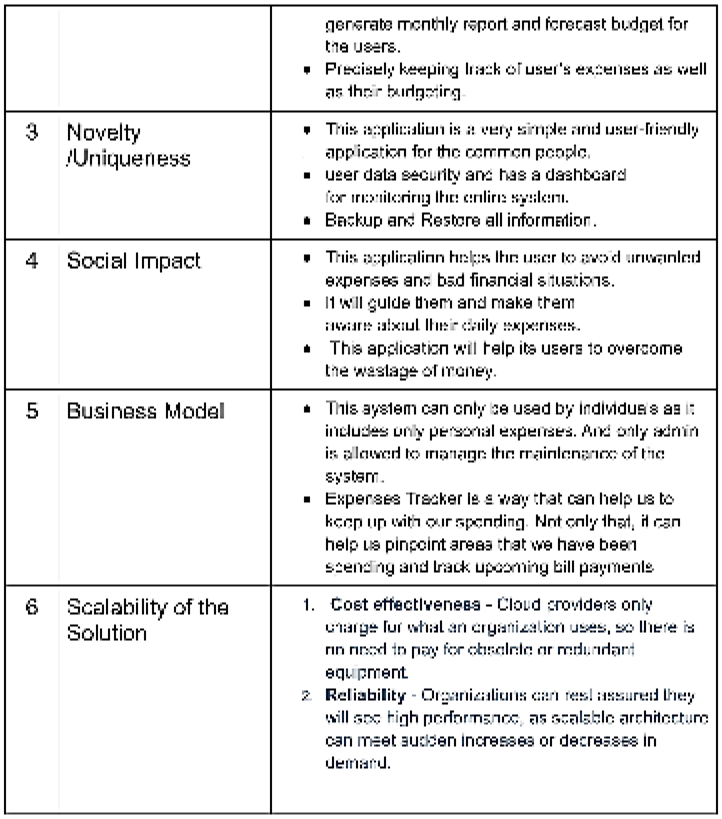


Step-3: Idea Prioritization


Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
24/7 Availability
Complete Less code
Interactive UI
User friendly
Fast API
Accessible from any device
Display the report graphically
Data Security
Backup
Notify User
Importance
Regular updates
Duplicates get detected immediately
Set alerts and reminders
Easy to track
Integration with other capable software
Fast Data Retrieval
Offers better analytics
Streamline Tax Deductions
Feasibility
After you collaborate
You can export the mural as an image or pdf to share with members of your company who might find it helpful.
Quick add-ons
Export the mural
Keep moving forward
Strategy blueprint
Customer experience journey map
Strengths, weaknesses, opportunities & threats
Open the template
Share template feedback

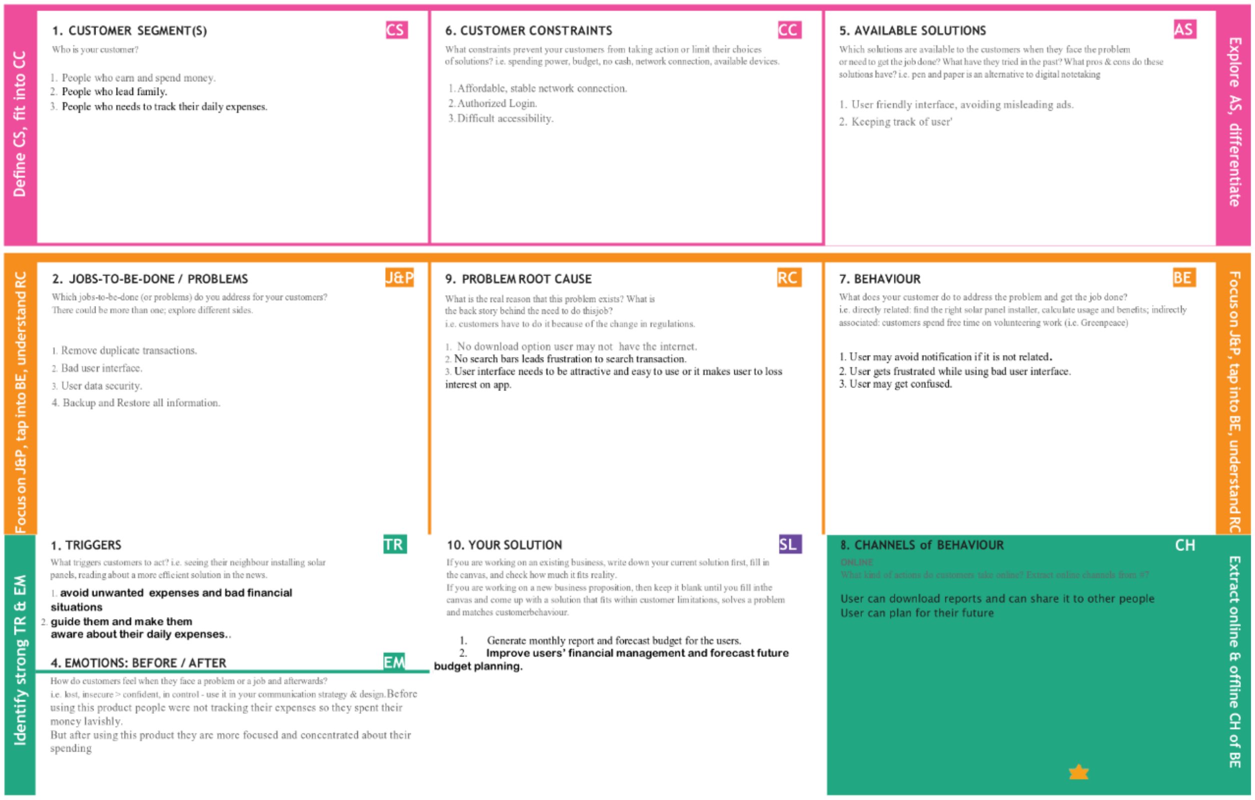
3.3 Proposed Solution





S.no	Parameter	Description
1	Problem Statement	▪ By tracking expenses and following a plan, a budget makes it easier to pay bills on time, build an emergency fund, and save for major expenses such as a car or home. ▪ A Daily Expense Tracker is a one kind of digital diary that helps to keep an eye on all of our money related transitions and also provides all financial activities report daily, weekly, monthly and yearly. ▪ At the Instant, there is no as such complete solution present easily or we should say free of cost which enables a person to keep a track of its daily expenditure easily.
2	Idea/Solution Description	▪ To develop a systematic system that will help to improve users' financial management and forecast future budget planning. ▪ To test and evaluate the reliability of the system to generate monthly report and forecast budget for the users. ▪ Precisely keeping track of user's expenses as well as their budgeting.
3	Novelty /Uniqueness	▪ This application is a very simple and user-friendly application for the common people. ▪ user data security and has a dashboard for monitoring the online system. ▪ Backup and Restore all information.
4	Social Impact	▪ This application helps the user to avoid unwanted expenses and bad financial situations. ▪ It will guide them and make them aware about their daily expenses. ▪ This application will help its users to overcome the wastage of money.
5	Business Model	▪ This system can only be used by individuals as it includes only personal expenses. And only admin is allowed to manage the maintenance of the system. ▪ Expenses Tracker is a way that can help us to keep up with our spending. Not only that, it can help us pinpoint areas that we have been spending and track upcoming bill payments
6	Scalability of the Solution	1. **Cost effectiveness** - Cloud providers only charge for what an organization uses, so there is no need to pay for obsolete or redundant equipment. 2. **Reliability** - Organizations can rest assured they will see high performance, as scalable architecture can meet sudden increases or decreases in demand.

3.4 Problem Solution fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

1. People who earn and spend money.
2. People who lead family.
3. People who needs to track their daily expenses.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

1. Affordable, stable network connection.
2. Authorized Login.
3. Difficult accessibility.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

1. User friendly interface, avoiding misleading ads.
2. Keeping track of user'

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

1. Remove duplicate transactions.
2. Bad user interface.
3. User data security.
4. Backup and Restore all information.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do thisjob?
i.e. customers have to do it because of the change in regulations.

1. No download option user may not have the internet.
2. No search bars leads frustration to search transaction.
3. User interface needs to be attractive and easy to use or it makes user to loss interest on app.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

1. User may avoid notification if it is not related.
2. User gets frustrated while using bad user interface.
3. User may get confused.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

1. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

1. **avoid unwanted expenses and bad financial situations**
2. **guide them and make them aware about their daily expenses..**

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.Before using this product people were not tracking their expenses so they spent their money lavishly.
But after using this product they are more focused and concentrated about their spending

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill inthe canvas and come up with a solution that fits within customer limitations, solves a problem and matches customerbehaviour.

1. Generate monthly report and forecast budget for the users.
2. **Improve users' financial management and forecast future budget planning.**

8. CHANNELS of BEHAVIOUR CH

ONLINE
What kind of actions do customers take online? Extract online channels from #7

User can download reports and can share it to other people
User can plan for their future

Extract online & offline CH of BE

4.REQUIREMENT ANALYSIS

4.1 Functional requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through Application Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User monthly expense tentative data	Data to be registered in the app
FR-4	User monthly income data	Data to be registered in the app
FR-5	Alert/ Notification	Alert through E-mail Alert through SMS
FR-6	User Budget Plan	Planning and Tracking of user expense vs budget limit

4.2 Non-Functional requirements

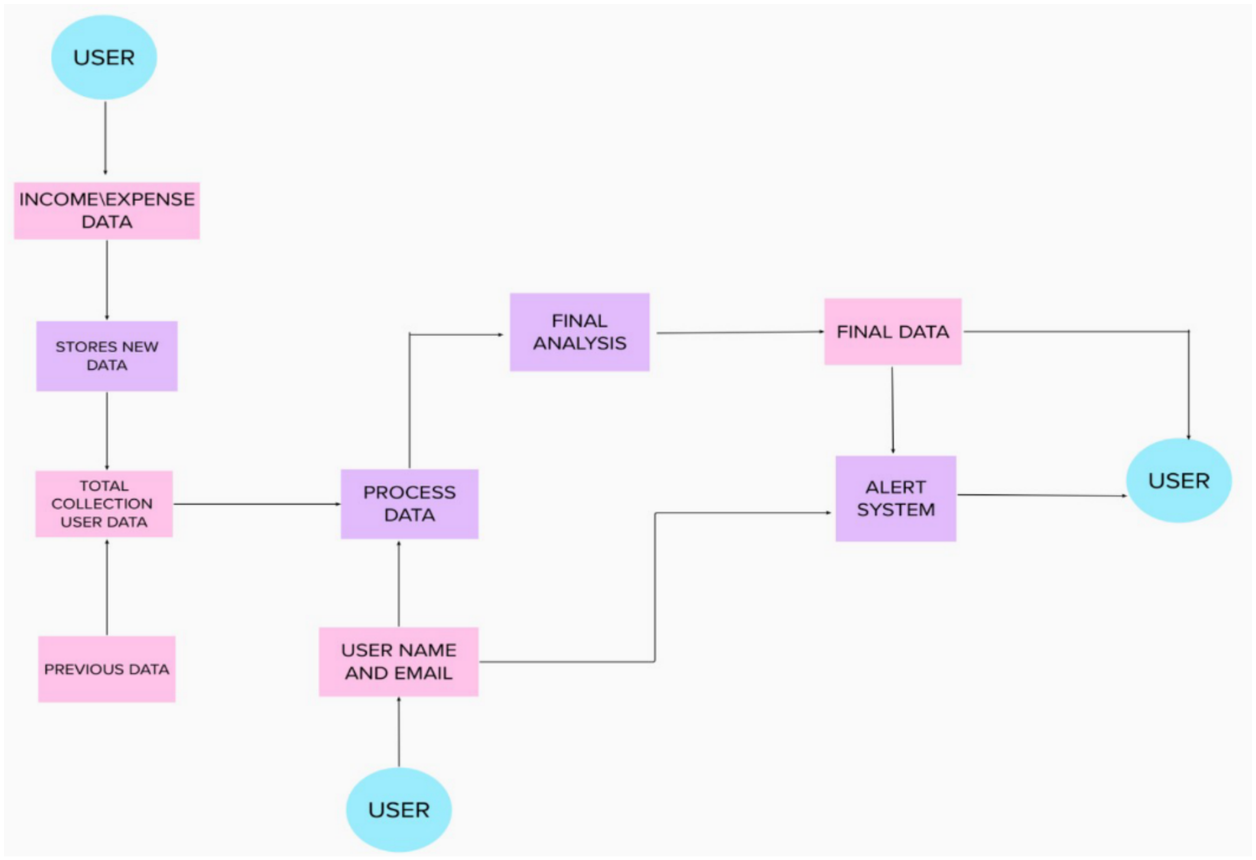
Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	Effectiveness, efficiency and overall satisfaction of the user while interacting with our application.
NFR-2	**Security**	Authentication, authorization, encryption of the application.
NFR-3	**Reliability**	Probability of failure-free operations in a specified environment for a specified time.
NFR-4	**Performance**	How the application is functioning and how responsive the application is to the end-users.
NFR-5	**Availability**	Without near 100% availability, application reliability and the user satisfaction will affect the solution.
NFR-6	**Scalability**	Capacity of the application to handle growth, especially in handling more users.

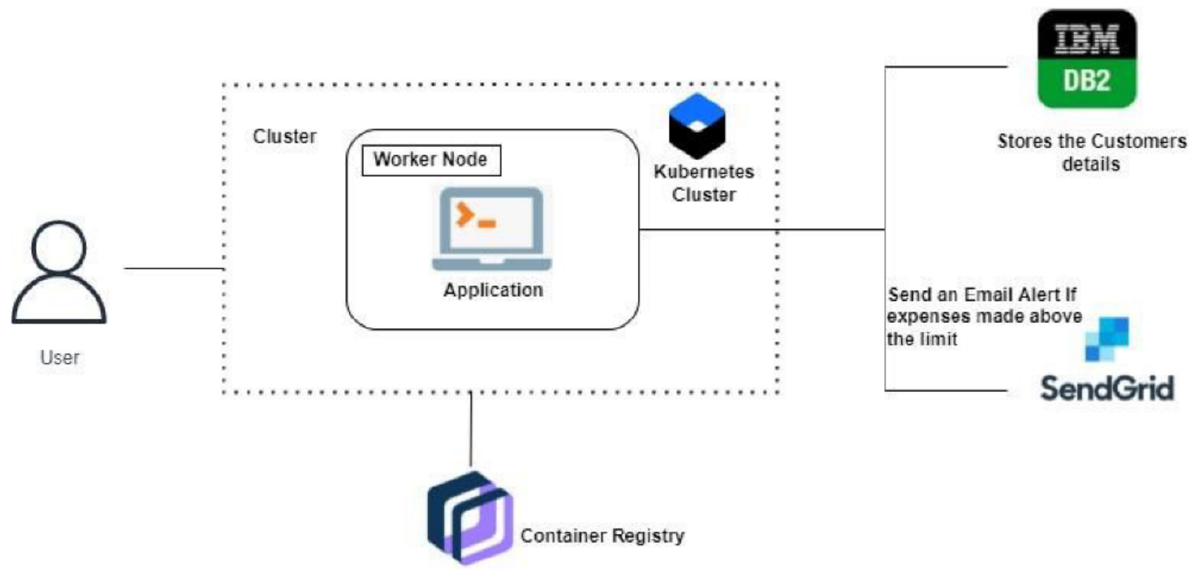
5.PROJECT DESIGN

5.1 Data Flow Diagrams

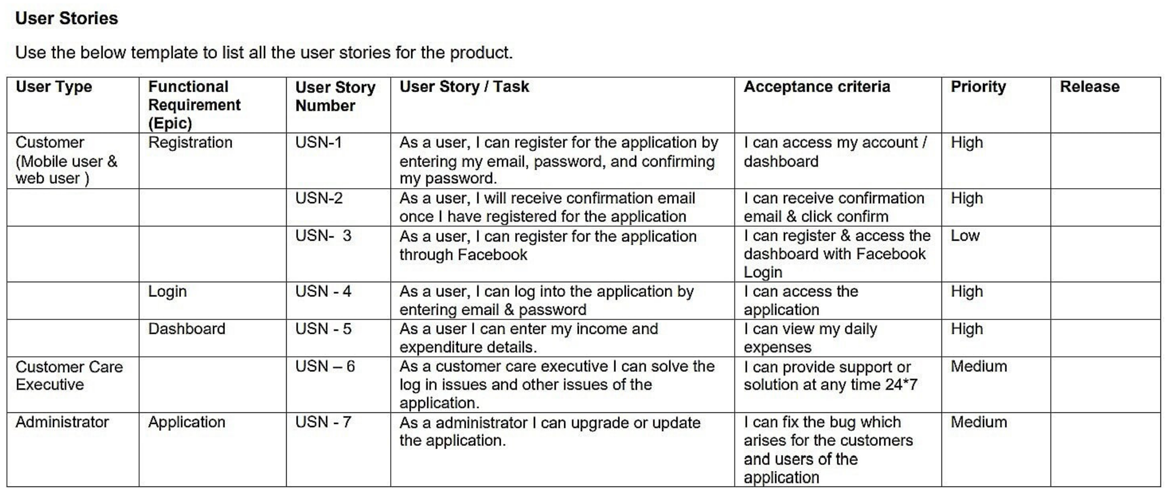


5.2 Solution & Technical Architecture

Technical Architecture:



5.3 User Stories



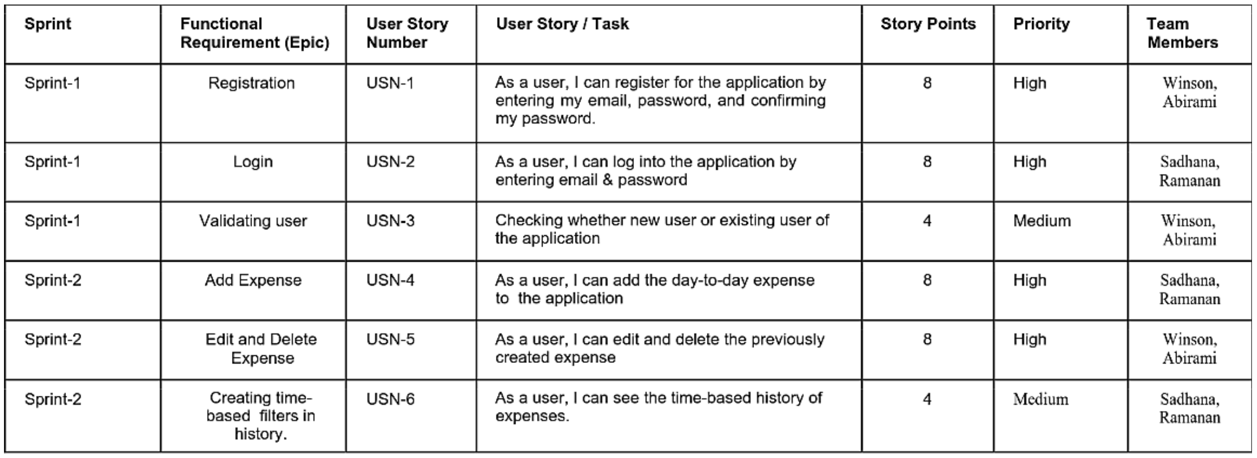
User Stories

Use the below template to list all the user stories for the product.

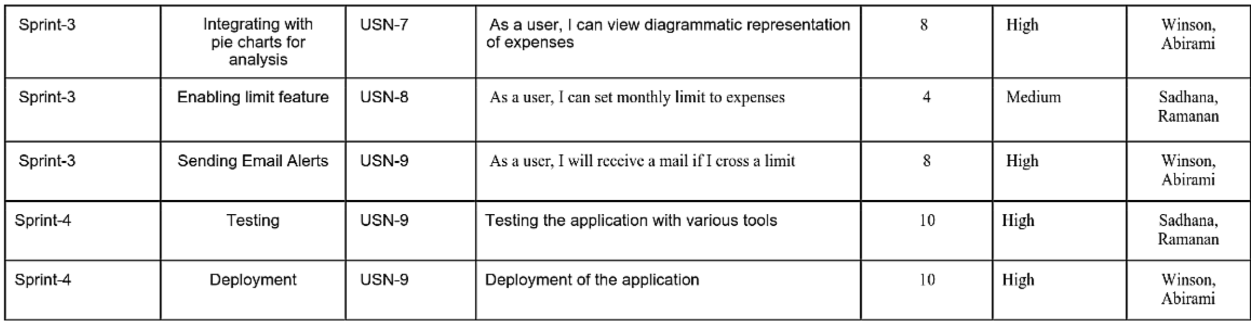
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user & web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	
		USN- 3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	
	Login	USN - 4	As a user, I can log into the application by entering email & password	I can access the application	High	
	Dashboard	USN - 5	As a user I can enter my income and expenditure details.	I can view my daily expenses	High	
Customer Care Executive		USN – 6	As a customer care executive I can solve the log in issues and other issues of the application.	I can provide support or solution at any time 24*7	Medium	
Administrator	Application	USN - 7	As a administrator I can upgrade or update the application.	I can fix the bug which arises for the customers and users of the application	Medium	

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

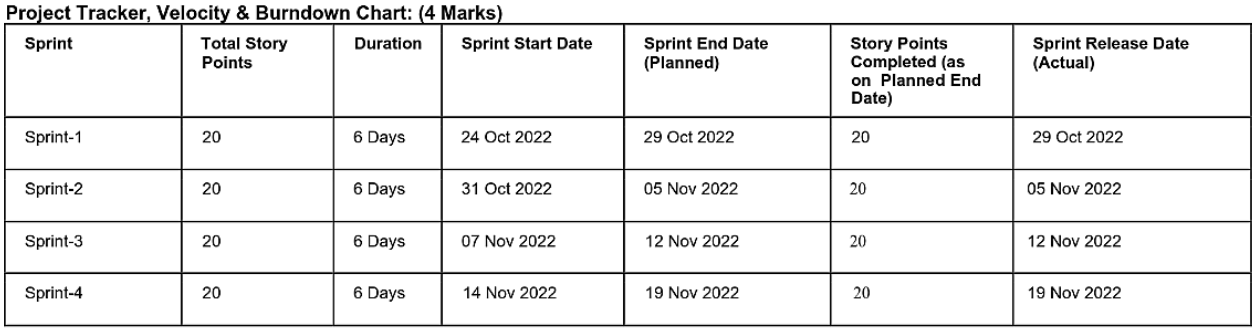


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	8	High	Winson, Abirami
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password	8	High	Sadhana, Ramanan
Sprint-1	Validating user	USN-3	Checking whether new user or existing user of the application	4	Medium	Winson, Abirami
Sprint-2	Add Expense	USN-4	As a user, I can add the day-to-day expense to the application	8	High	Sadhana, Ramanan
Sprint-2	Edit and Delete Expense	USN-5	As a user, I can edit and delete the previously created expense	8	High	Winson, Abirami
Sprint-2	Creating time-based filters in history.	USN-6	As a user, I can see the time-based history of expenses.	4	Medium	Sadhana, Ramanan



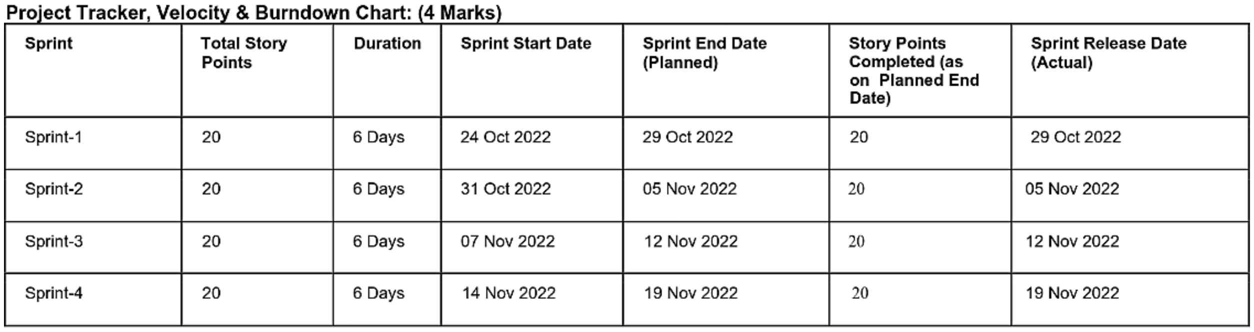
Sprint-3	Integrating with pie charts for analysis	USN-7	As a user, I can view diagrammatic representation of expenses	8	High	Winson, Abirami
Sprint-3	Enabling limit feature	USN-8	As a user, I can set monthly limit to expenses	4	Medium	Sadhana, Ramanan
Sprint-3	Sending Email Alerts	USN-9	As a user, I will receive a mail if I cross a limit	8	High	Winson, Abirami
Sprint-4	Testing	USN-9	Testing the application with various tools	10	High	Sadhana, Ramanan
Sprint-4	Deployment	USN-9	Deployment of the application	10	High	Winson, Abirami

Project Tracker, Velocity & Burndown Chart: (4 Marks)



Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Project Tracker, Velocity & Burndown Chart: (4 Marks)



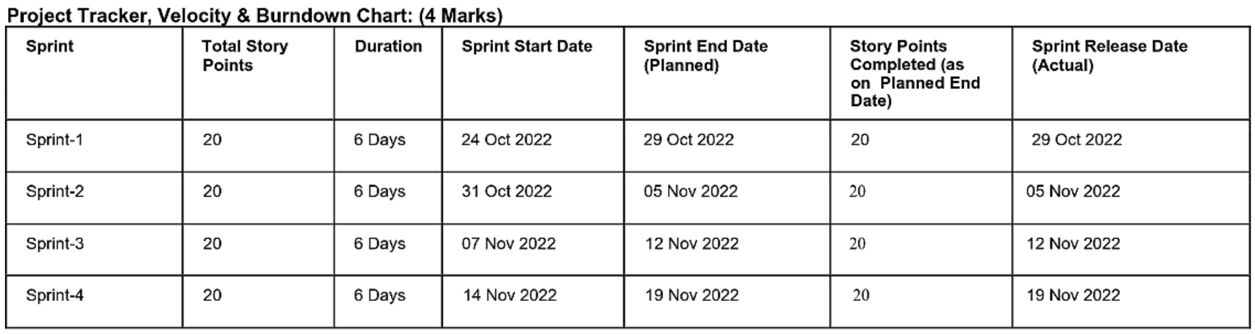
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

AV = 20/6 = 3.33

6.2 Sprint Delivery Schedule



Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

7.CODING & SOLUTIONING

7.1 Feature 1

We have added the data visualiza on methods for expenditure. The pie chart have been used to represent the monthly expenses. The pie chart is a pictorialrepresentation of data that makes it possible to visualize the relationshipsbetween the parts and the whole of a variable. For example, it is possible tounderstand the industry count or percentage of a variable level from the division by areas or sectors. The recommended use for pie charts is two-dimensional, as three-dimensional use can be confusing.

The dimensions form sectors of the measurement values; they can have one or two sizes and up to two measures. The first dimension is used to define the angle of each sector that makes up the chart and the second dimension optionally determines the radius of each sector. Additionally, these plots are useful for comparing data over a fixed period since they do not show changes over time. Therefore, their use should be considered if:

- You are looking to categorize and compare a set of data.

1. You only have positive values.
2. You have less than seven categories since a larger number can make itdifficult to perceive each segment.

CODE :

```
{% extends 'base.html' %}

{% block content%}
{% if success%}
<div class="container-fluid position-fixed z">
    <div class="mt-4 float-end">
        <div class="alert alert-success alert-
dismissible fade show" role="alert">
            <i class="fas fa-check-circle"></i>
&nbsp;{{ success }}
            <button type="button" class="btn-close"
data-bs-dismiss="alert" aria-
```

```
label="Close"></button>
        </div>
    </div>
</div>
{% endif%}

{% if danger%}
<div class="container-fluid position-fixed z">
  <div class="mt-4 float-end">
    <div class="alert alert-danger alert-dismissible fade show" role="alert">
     <i class="fas fa-exclamation-triangle"></i> &nbsp; {{ danger }}
    <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
  </div>
 </div>
</div>
{% endif %}

<div class="container-fluid h-custom pb-5" >
  <div class="row d-flex justify-content-center align-items-center h-100">
           <div class="col-md-9 col-lg-6 col-xl-5 mt-5">
          <img src="https://wdbucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/bg.png" alt="Index Logo"
            class="img-fluid login_logo">
        </div><div class="col-md-8 col-lg-6 col-xl-4 offset-xl-1 mt-5 mb-5">
          <div class="line me-5" style="margin-top: 8rem; margin-left: 5px;">
            <h2 class="h2"  font-weight:bolder">
```

```
Personal Expense Tracker</h2>
            <h4 class="h4" style="color: grey;">Beware of little expenses. A small leak will sink a great ship.</h4>
          </div>

          {% if not current_user.is_authenticated %}
          <button onclick="location.href='{{url_for('login')}}'" type="button" class="btn btn-success">Login</button>
          <button onclick="location.href='{{url_for('register')}}'"type="button" class="btn btn-warning">Register</button>

          {% endif %}


          {% if current_user.is_authenticated %}
          <button onclick="location.href='{{url_for('logout')}}'" type="button" class="btn btn-danger">Logout</button>

          {% endif %}

        </div>
      </div>
  </div>

{% endblock%}
```

```python
from flask import Flask, render_template, url_for, request, redirect, flash, abort
from flask_login import LoginManager
from flask_login import login_required, current_user, login_user, logout_user, UserMixin
from werkzeug.security import generate_password_hash, check_password_hash
import ibm_db
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
api_key = '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'
sg = SendGridAPIClient(api_key)

# Ibm Db2


def connection():
    try:
        conn = ibm_db.connect(
            "DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;\
            PORT=30119;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fpq67161;PWD=3IOG7aiAt5sF2eBq", '', '')
        print("Connected to Database")
        return conn
    except:
        print("Not Connected to Database")


# App
app = Flask(__name__)
app.config['SECRET_KEY'] = '71001910'

conn = connection()

login_manager = LoginManager()
login_manager.login_view = 'login'
login_manager.init_app(app)


class User(UserMixin):
    def __init__(self, user_json):
        self.user_json = user_json

    def get_id(self):
        object_id = self.user_json.get('PERSONID')
        return str(object_id)


@login_manager.user_loader
def load_user(user_id):
    sql = "SELECT * FROM login WHERE personid=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, user_id)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    return User(account)


@app.route('/')
def index():
    return render_template('index.html', index='active', success=request.args.get('success'), danger=request.args.get('danger'))


@app.route('/login')
def login():
    return render_template('login.html', login='active', danger=request.args.get('danger'), success=request.args.get('success'))


@app.route('/login', methods=['POST', 'GET'])
def login_rec():
    if request.method == 'POST':

        email = request.form['email']
        password = request.form['password']
        remember = True if request.form.get('remember') else False

        sql = "SELECT * FROM login WHERE email=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if not account:
            return redirect(url_for('login', danger="You do not have an registered account so, please register and login"))
        else:
            if not check_password_hash(account['PASSWORD'], password):
                return redirect(url_for('login', danger="You've entered a wrong password"))
            else:
                userdetails = User(account)
                login_user(userdetails, remember=remember)
                return redirect(url_for('dashboard', success='Login Successfull'))


@app.route('/logout')
@login_required
def logout():
    logout_user()
    return redirect(url_for('index', success="Logout successfull"))


@app.route('/register')
def register():
    return render_template('register.html', register='active',
danger=request.args.get('danger'))


  @app.route('/register', methods=['POST', 'GET'])
  def addrec():
      if request.method == 'POST':

          firstname = request.form['firstname']
          lastname = request.form['lastname']
          email = request.form['email']
          password = request.form['password']
          re_password = request.form['re-password']
          print(firstname)
          sql = "SELECT * FROM login WHERE email=?"
          prep_stmt = ibm_db.prepare(conn, sql)
          ibm_db.bind_param(prep_stmt, 1, email)
          ibm_db.execute(prep_stmt)
          account = ibm_db.fetch_assoc(prep_stmt)

          if account:
              return redirect(url_for('login', danger="You already have an account so, please login with your credentials"))

          elif (password != re_password):
              return redirect(url_for('register', danger="Your password doesn't match"))

          else:
              insert_sql = "INSERT INTO login(firstname,lastname,email,password) VALUES
```

```python
(?,?,?,?)"
            prep = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep, 1, firstname)
            ibm_db.bind_param(prep, 2, lastname)
            ibm_db.bind_param(prep, 3, email)
            ibm_db.bind_param(prep, 4, generate_password_hash(
                password, method='sha256'))
            ibm_db.execute(prep)

            message = Mail(
                from_email='admin@pta.com',
                to_emails=email,
                subject='Registration Successfull',
                html_content='<strong>and easy to do anywhere,</strong>')
    return redirect(url_for('login', success="Registration Successfull"))


@app.route('/dashboard')
@login_required
def dashboard():

    # Expense Details SQL
    expensedetails = []
    sql = "SELECT AMOUNT,DETAILS,CHAR(DATE(DANDT),USA) AS DATEADDED, CHAR(TIME(DANDT),USA) AS TIMEADDED FROM USERDATA
```

```
WHERE USERID = ?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, current_user.user_json['PERSONID'])
    ibm_db.execute(stmt)
    details = ibm_db.fetch_assoc(stmt)
    while details != False:
        expensedetails.append(details)
        details = ibm_db.fetch_assoc(stmt)

    label = [row['DATEADDED'] for row in expensedetails]
    amountlabel = [row['AMOUNT'] for row in expensedetails]

    # Totalexpense SQL
    sql2 = "SELECT SUM(AMOUNT) AS TOTALVAL FROM USERDATA WHERE USERID = ?"
    stmt2 = ibm_db.prepare(conn, sql2)
    ibm_db.bind_param(stmt2, 1, current_user.user_json['PERSONID'])
    ibm_db.execute(stmt2)
    totalexpense = ibm_db.fetch_assoc(stmt2)
    if totalexpense['TOTALVAL'] is None:
        totalexpense['TOTALVAL'] = 0

    # walletbalance SQL
    sql3 = "SELECT SUM(WALLETAMOUNT) AS TOTALVAL FROM WALLET WHERE WALLETID = ?"
    stmt3 = ibm_db.prepare(conn, sql3)
    ibm_db.bind_param(stmt3, 1,
current_user.user_json['PERSONID'])
    ibm_db.execute(stmt3)
    walletbalance = ibm_db.fetch_assoc(stmt3)
    if walletbalance['TOTALVAL'] is None:
        walletbalance['TOTALVAL'] = 0
        availablebalance = 0
    else:
        availablebalance = int(
            walletbalance['TOTALVAL']) - int(totalexpense['TOTALVAL'])
    if (availablebalance <= 50):
        flash("Your balance is too low!!!")
    elif (availablebalance > 50 and availablebalance <= 200):
        flash("Your balance is getting low so take care of your expenses...!!!")

    return render_template('dashboard.html', dashboard='active', name=current_user.user_json['FIRSTNAME'], success=request.args.get('success'), danger=request.args.get('danger'), expensedetails=expensedetails, totalexpense=totalexpense['TOTALVAL'], walletbalance=availablebalance, label=label, amountlabel=amountlabel)


@app.route('/addexpense/<balance>', methods=['POST'])
@login_required
```

```python
def addexpense(balance):
    amount = request.form['amount']
    detail = request.form['details']

    if (int(amount) == 0):
        return redirect(url_for('dashboard', danger="Please enter some amount"))

    else:
        sql = "INSERT INTO USERDATA(USERID,AMOUNT,DETAILS) VALUES(?,?,?)"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, current_user.user_json['PERSONID'])
        ibm_db.bind_param(stmt, 2, amount)
        ibm_db.bind_param(stmt, 3, detail)
        ibm_db.execute(stmt)
        print('sendMail')
        if (int(balance) <= 100):
            message = Mail(
                from_email='admin@pta.com',
                to_emails=current_user.user_json['EMAIL'],
                subject='Low Balance !!',
                html_content='<strong>and easy to do anywhere,</strong>')
        return redirect(url_for('dashboard', success="Expense added successfully"))


@app.route('/addmoney', methods=['POST'])
```

```
@login_required
def addmoney():
    amount = request.form['walletamount']

    if (int(amount) == 0):
        return redirect(url_for('dashboard', danger="Please enter some amount"))

    else:
        sql = "INSERT INTO WALLET(WALLETID,WALLETAMOUNT) VALUES(?,?)"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, current_user.user_json['PERSONID'])
        ibm_db.bind_param(stmt, 2, amount)
        ibm_db.execute(stmt)

        return redirect(url_for('dashboard', success="Money added successfully"))


# Delete
@app.route('/deleteexpense/<val>/<amount>')
@login_required
def deleteexpense(val, amount):

    sql = "DELETE USERDATA WHERE USERID=? AND CHAR(TIME(DANDT),USA)= ? AND AMOUNT=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, current_user.user_json['PERSONID'])
    ibm_db.bind_param(stmt, 2, val)
    ibm_db.bind_param(stmt, 3, amount)
    ibm_db.execute(stmt)

    return redirect(url_for('dashboard', success="Deleted Successfully"))


@app.errorhandler(500)
def page_not_found():
    return redirect(url_for('index', danger="oops!!! error occured, try again")), 500


@app.errorhandler(404)
def not_found():
    return redirect(url_for('index', danger="oops!!! error occured, try again")), 404


if __name__ == "__main__":
```

7.2 Feature 2

Email notifications will be sent to the users once they cross the expenditure limit through send grid mail system. Most notifications are transactional, meaning a recipient’s action or account activity triggers them. But some notifications are marketing related, encouraging the recipient to take a specific action. Ecommerce product notifications inform recipients about new products or discounts. Plus, unlike general marketing emails, these are highly personalized and focus on a single product. For example, if a customer views an item on your website and that item goes on sale, you can send the customer a notification to let them know this is the best time to buy. Users can also opt into receiving notifications when an out-of-stock item is back in stock.

Notification emails tend to perform well because the content is highly relevant to the recipient. But the only way for the recipient to know this is if you state the content clearly in the subject line.

For example, the subject line “New Sign-in to Your Account” gets straight to the point, letting the user know why you sent this notification.

7.3 Database Schema

Tables : 1)
REGISTER
id INT NOT NULL GENERATED ALWAYS AS
IDENTITY, username VARCHAR(255) NOT NULL,
email VARCHAR(255) NOT NULL,
password VARCHAR(255) NOT
NULL

1. EXPENSES

id INT NOT NULL GENERATED ALWAYS AS
IDENTITY, userid INT NOT NULL,
date TIMESTAMP NOT NULL, expensename
VARCHAR(255) NOT NULL,

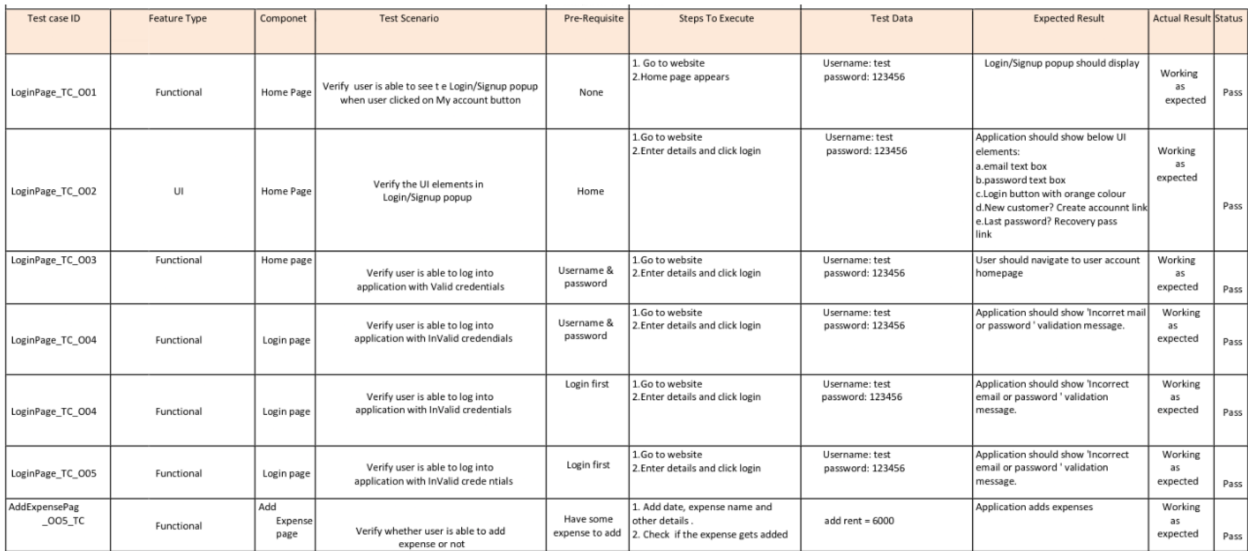
amount INT NOT NULL,
paymode VARCHAR(255) NOT NULL,
category VARCHAR(255) NOT NULL

2. LIMITS

id INT NOT NULL GENERATED ALWAYS AS IDENTITY, userid VARCHAR(255) NOT NULL,
limitss VARCHAR(255) NOT NULL

8.TESTING

8.1 Test Cases



Test case ID	Feature Type	Componet	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
LoginPage_TC_O01	Functional	Home Page	Verify user is able to see t e Login/Signup popup when user clicked on My account button	None	1. Go to website 2.Home page appears	Username: test password: 123456	Login/Signup popup should display	Working as expected	Pass
LoginPage_TC_O02	UI	Home Page	Verify the UI elements in Login/Signup popup	Home	1.Go to website 2.Enter details and click login	Username: test password: 123456	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create accounnt link e.Last password? Recovery pass link	Working as expected	Pass
LoginPage_TC_O03	Functional	Home page	Verify user is able to log into application with Valid credentials	Username & password	1.Go to website 2.Enter details and click login	Username: test password: 123456	User should navigate to user account homepage	Working as expected	Pass
LoginPage_TC_O04	Functional	Login page	Verify user is able to log into application with InValid credendials	Username & password	1.Go to website 2.Enter details and click login	Username: test password: 123456	Application should show 'Incorret mail or password ' validation message.	Working as expected	Pass
LoginPage_TC_O04	Functional	Login page	Verify user is able to log into application with InValid credentials	Login first	1.Go to website 2.Enter details and click login	Username: test password: 123456	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass
LoginPage_TC_O05	Functional	Login page	Verify user is able to log into application with InValid crede ntials	Login first	1.Go to website 2.Enter details and click login	Username: test password: 123456	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass
AddExpensePag _OO5_TC	Functional	Add Expense page	Verify whether user is able to add expense or not	Have some expense to add	1. Add date, expense name and other details . 2. Check if the expense gets added	add rent = 6000	Application adds expenses	Working as expected	Pass

8.2 User Acceptance Testing

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	5	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	75

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	29	0	0	29
Security	4	0	0	4
Outsource Shipping	6	0	0	6
Exception Reporting	7	0	0	2
Final Report Output	5	0	0	5
Version Control	1	0	0	1

9.RESULTS

9.1 Performance Metrics

1. Tracking income and expenses: Monitoring the income and tracking all expenditures (through bank accounts, mobile wallets, and credit & debit cards).
2. Transaction Receipts: Capture and organize your payment receipts to keep track of your expenditure.
3. Organizing Taxes: Import your documents to the expense tracking app, and it will streamline your income and expenses under the appropriate tax categories.
4. Payments & Invoices: Accept and pay from credit cards, debit cards, net banking, mobile wallets, and bank transfers, and track the status of your invoices and bills in the mobile app itself. Also, the tracking app sends reminders for payments and automatically matches the payments with invoices.
5. Reports: The expense tracking app generates and sends reports to give a detailed insight about profits, losses, budgets, income, balance sheets, etc.,

6. E-commerce integration: Integrate your expense tracking app with your eCommerce store and track your sales through payments received via multiple payment methods.
7. Vendors and Contractors: Manage and track all the payments to the vendors and contractors added to the mobile app.
8. Access control: Increase your team productivity by providing access control to particular users through custom permissions.
9. Track Projects: Determine project profitability by tracking labor costs, payroll, expenses, etc., of your ongoing project.
10. Inventory tracking: An expense tracking app can do it all. Right from tracking products or the cost of goods, sending alert notifications when the product is running out of stock or the product is not selling, to purchase orders.
11. In-depth insights and analytics: Provides in-built tools to generate reports with easy-to-understand visuals and graphics to gain insights about the performance of your business.
12. Recurrent Expenses: Rely on your budgeting app to track, streamline, and automate all the recurrent expenses and remind you on a timely basis.

10.ADVANTAGES & DISADVANTAGES

1. **Achieve your business goals** with a tailored mobile app that perfectly fits your business.
2. **Scale-up** at the pace your business is growing.
3. Deliver an **outstanding** customer experience through additional control over the app.
4. Control the **security** of your business and customer data.
5. Open **direct marketing channels** with no extra costs with methods such as push notifications.
6. **Boost the productivity** of all the processes within the organization.
7. Increase **efficiency** and **customer satisfaction** with an app aligned to their needs.
8. **Seamlessly integrate** with existing infrastructure.

9. Ability to provide **valuable insights**.
10. Optimize sales processes to generate **more revenue** through enhanced data collection.

11.CONCLUSION

From this project, we are able to manage and keep tracking the dailyexpenses as well as income. While making this project, we gained a lot ofexperience of working as a team. We discovered various predicted andunpredicted problems and we enjoyed a lot solving them as a team. We adopted things like video tutorials, text tutorials, internet and learningmaterials to make our project complete.

12.FUTURE

The project assists well to record the income and expenses in general. However, this project has some limita ons:

1. The application is unable to maintain the backup of data once it isuninstalled.

2. This application does not provide higher decision capability.

To further enhance the capability of this application, we recommend thefollowing features to be incorporated into the system:

3. Multiple language interface.
4. Provide backup and recovery of data.
5. Provide better user interface for user.

6. Mobile apps advantage.

13.APPENDIX

Source Code Github Link : https://github.com/IBM-EPBL/IBM-Project-325-1658287565 **IBM-Project-325-1658287565**

Project Demo Video Link :

https://drive.google.com/file/d/19gNcaIW91kQFQI8ytmLBq1pELsY-TH3s/view?usp=share_link

Project Live Link :

http://169.51.204.92:32434/