

PERSONAL EXPENSE TRACKER

Rithik Sarvesh B, Santhiya K M, Praneetha S, Upendra Singh A
Department of Computer Science and Engineering
Kumaraguru College of Technology (Coimbatore)

1.INTRODUCTION:

Expense Tracker is an everyday expense control application designed to track effortlessly and efficiently each day costs. This helps us to get rid of the need of paper responsibilities that systematically maintains information. This device can be utilized by any individual to govern their income expenditure from each day to annual basis and to hold an eye on their spending, including the person to whom the payments were made and the purpose for the payment. It aids us in remembering and adding information about what money we receive from others and what costs or payments we must make on a given date or month. We have categories in the expense tracker such as add expense, monthly expenses, add new expense, and so on. It gives the daily remainder about the savings we need to do.

1.1 Project overview:

Expense tracking is one of the tedious process to follow up. So, we have implemented an online expense tracker with the help of IBM tools like IBM DB2. Our webpage consists of register page, login page, wallet page and add expenditure page.

1.2 Purpose:

The objective of this app is to act as a remainder to help user track their budget. Through this online application students, small business will be benefited the most in today's fast-paced environment. Many people today are looking for efficient ways to track their expenses because we live in a hurry-up and get-it-done culture. Some research on household budgets has been conducted in recent years which says Budget tracking is challenging in most circumstances since budget management is done verbally rather than on paper.

2.LITERATURE SURVEY

2.1 Existing problem

[1] Many homes budgeting tools (betterbudgeting.com; mybudgetkeeper.com; RLT Family Budget, 2003) already exist but most of them are locally based. These tools are run on the local computer mainly because of security reasons.

Advantages: As it is local, there will be no privacy breach and it can be customizable according to the user's need.

Disadvantages: Existing solutions are static. Dynamic sites can be more beneficiary to the people who need regular updates like updated loan interest, fluctuating stock markets etc. The existing system is not user friendly because data is not maintained efficiently

[2] Expense Tracker project which will keep 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. The expense tracker will generate a report at the end of month to show Income-Expense via multiple graphs.

Advantages: 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.

Disadvantages: This project seems to be more dynamic and erroneous entry will collapse the whole balance of the income-expense and regular entry should be done to track the progress.

[3] A mobile application has been developed that keeps track of all your daily transactions, keeps track of your money lent or borrowed, suggests you with the most effective investment options, offers your discounts in popular categories, view exchange and to read latest authenticated financial news. This Paper's main aim is to eliminate the use of sticky notes, spreadsheets and handling of large chunks of data is successful, the new experience is hassle-free and very handy.

Advantages: With this application the user can manage his/her expenses more effectively. This application can also help digital marketing agencies in rolling out their advertising campaigns more effectively.

Disadvantages: It is a tracker application designed only for people who are doing business so in order to be used by different categories of people, applications can be further developed with those features.

[4] An online application that acts as a remainder to help users track their budget is developed. Students, small business will benefit the most in today's fast-paced environment. Many people today are looking for efficient ways to track their expenses because we live in a hurry-up and get-it-done culture. Some research on household budgets has been conducted in recent years which says Budget tracking is challenging in most circumstances since budget management is done verbally rather than on paper.

Advantages: It will contain a variety of record-keeping choices (for example, food, travel fuel, salary, and so on). It will continue to deliver notifications on its own to cover our daily expenses. The app allows us to clearly understand our expenses to keep track of our earnings and better plan for the next coming month.

Disadvantages: It only keeps recording the expenses and doesn't give suggestions on the amount which is lavishly spent.

[5] Daily Expense Tracker is designed in conformity with managing the utility user's each day price in a greater environment friendly and manageable way. Daily Expense Tracker helps in accordance with maintain the document regarding daily costs yet month-to-month income. The tracking regarding prices is classified daily, hebdomadal yet monthly, such helps in imitation of advice more charges made.

Advantages: The goals that are achieved are user-friendly or bendy interface. efficient estimate yet forecasting. improved productivity and instant access.

Disadvantages: This provision does solely remain chronic via men and women as it consists of solely private expenses. And only admin is allowed in conformity with square the preservation regarding the system.

[6] A web application that tracks minute transactions that we make and monitor expenses. Filters have been implemented such as storing income and expenses based on the transaction, date of the income or expense and the category of items, total expenses per day, scan and store the images of the bill with an optional calculator.

Advantages: The Application can manage users' daily expenses in a more structured and organized manner. Thus, it can minimize the hand-operated computations which would've been involved otherwise.

Disadvantages: The application uses traditional technology and several legacy codes.

2.2 References

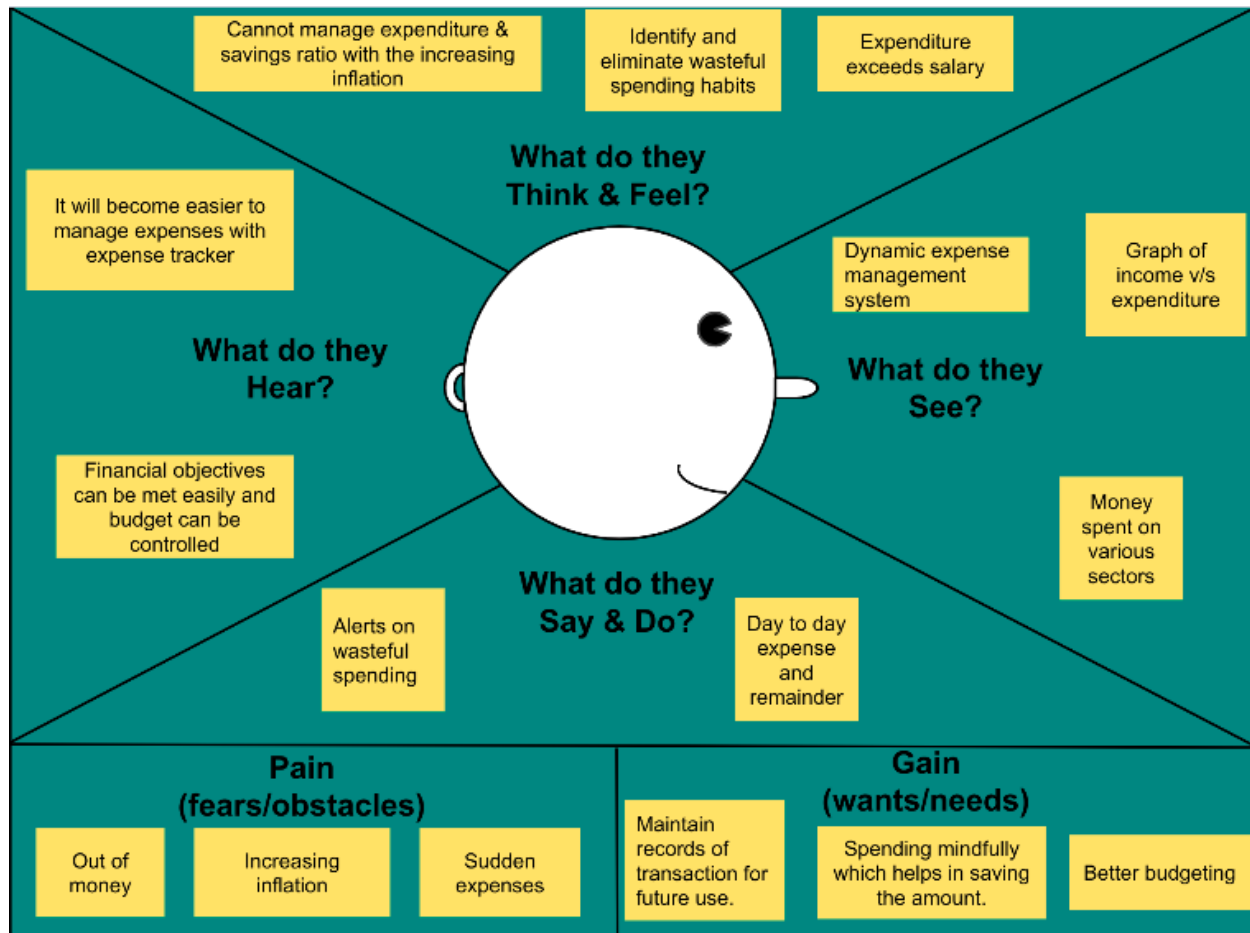
- [1] Bekaroo, Girish & Sunhaloo, Sameer. (2007). Intelligent Online Budget Tracker. Retrieved from https://www.researchgate.net/publication/237448489_Intelligent_Online_Budget_Tracker
- [2] S. Chandini, T. Poojitha, D. Ranjith, V.J. Mohammed Akram, M.S. Vani & V. Rajyalakshmi. (2019), Department of Computer Science & Engineering, Mother Theresa Institute of Engineering & Technology. Retrieved from <https://www.irjet.net/archives/V6/i3/IRJET-V6I31110.pdf>
- [3] Velmurugan, A & Mayan, J. & Niranjana, P & Francis, Richard. (2020). Expense Manager Application. Journal of Physics: Conference Series. 1712. 012039. 10.1088/1742-6596/1712/1/012039. Retrieved from https://www.researchgate.net/publication/347972162_Expense_Manager_Application
- [4] Gomathy, C K. (2022). EXPENDITURE MANAGEMENT SYSTEM. Retrieved from https://www.researchgate.net/publication/360620084_EXPENDITURE_MANAGEMENT_SYSTEM
- [5] Masendu, T. R., & Tripath, A. M. (2022). Daily Expense Tracker. International Journal of Research in Engineering, Science and Management, 5(5), 90–92. Retrieved from <http://www.journals.resaim.com/ijresm/article/view/2039>
- [6] Radhika, R., Praveen, A., Krishna, G.G., Anand, A., Anjali, T. (2022). Stay Home and Stay Safe with the Budgeting Tool a One-Stop Finance Tracker. In: Senjyu, T., Mahalle, P.N., Perumal, T., Joshi, A. (eds) ICT with Intelligent Applications. Smart Innovation, Systems and Technologies, vol 248. Springer, Singapore. https://doi.org/10.1007/978-981-16-4177-0_37

2.3 Problem Statement Definition

Problem Statement	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Working Professional	Limit my expenses	I can't invest enough time to manage money flow	I can't find the appropriate platform that is more user friendly and less time consuming	Frustrated
PS-2	Retired Adult	Manage my savings	I can't figure out the best plan that could keep my savings in control	I don't find any application that provides best plans to spend savings	Unsure & Heavy
PS-3	Student	Control money flow by improving my pocket money	I am not able to find the right way to manage money	I don't no experience in managing money and have less knowledge in finance management	Uncertain

3. IDEATION & PROPOSED SOLUTION


3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

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

Template



Brainstorm & idea prioritization

Personal Expense Tracker

- 10 minutes to prepare
- 1 hour to collaborate
- 20 people recommended

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.

10 minutes

Problem

How might we help the users plan their budget?

How might we help the user's family to track each other spending

How might we help the users to track different kind of expenses?

How might we crosscheck the amount entered is correct or the buget is calculated correct?

how might we manually enter the amount???

Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Managing expenses plays an important role in order to maintain expenses. Expenses sometimes exceed income which lead to shortage of money. With the era of online payments, people tend to overspend and hence the lack of awareness in money management leads to improper dealing of money
2.	Idea / Solution description	An application to manage daily expenses and give insights on spending habits and money flow. This requires one time plans to manage money so that people can focus on their work without wasting time. Alerts and provisions to set limits/budget and personalized expenditure plans that are generated based on user interaction after a series of questionnaires.
3.	Novelty / Uniqueness	Plan recommendation for expenditure will be a unique feature of this application. Customizable plans can be implemented as per the needs.
4.	Social Impact / Customer Satisfaction	Customers will be able to learn how their money flows during a certain period of time and hence can get a rough idea on setting limits/budgets which will eventually help to minimize the expenditure on unnecessary areas thus leading to better financial growth.
5.	Business Model (Revenue Model)	Once this application receives enough site traffic and users, it can be extended to two types of revenue model. The Licensing/one-time purchase in the Transaction-based model can be used for premium plan recommendations & more insights. The Advertisement-based model can be used in case of a higher number of users.

6.	Scalability of the Solution	The solution can be built in the form of a web app that also meets all the terms and conditions. Products dealing with money need to strictly follow nations finance laws and this product won't violate any of the laws as it doesn't deal with customers' bank/money but purely handles with the data input by the user during interaction for plan recommendation.
----	-----------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.4 Problem Solution fit

Define CS, fit into CC	1. CUSTOMER SEGMENTS CS <ul style="list-style-type: none"> * Students * Working men & women * Retired adults * Business Professionals 	6. CUSTOMER CONSTRAINTS CC <ul style="list-style-type: none"> * Fear of not meeting expectations * They don't have any remainder to warn them about their expenses and help to overcome the high expenses. * Lack of personal customizations in existing solutions 	5. AVAILABLE SOLUTIONS AS <ul style="list-style-type: none"> * Currently, they have solutions as platforms to keep an eye on their expenses & savings * They try to use these platforms to manage expenses but could end up losing interest because of bad UX or failing to manage finances 	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS J&P <ul style="list-style-type: none"> * Lack of awareness * Unexpected expenditures * Setting boundaries of expenses * Setting limits to specific expenses 	9. PROBLEM ROOT CAUSE RC <ul style="list-style-type: none"> * Carelessness * Not having limits for expenses * Overspending habits 	7. BEHAVIOUR BE <ul style="list-style-type: none"> * Roughly think about all of expenses to spend money on and prioritize which is important and plan accordingly. 	
Focus on J&P, tap into BE, understand RC	3. TRIGGERS TR <ul style="list-style-type: none"> * Fear of not being able to afford in case expense exceeds * Financial goals 	10. YOUR SOLUTION SL <ul style="list-style-type: none"> * An expense tracking application that helps users manage and control their expenses and get a better view graphically to analyze the money flow. * Alerts and limit setting options to have full control on a regular time basis (weekly/monthly/yearly) * Set goals for savings 	8. CHANNELS OF BEHAVIOUR CH <ul style="list-style-type: none"> 1. ONLINE <ul style="list-style-type: none"> * Social Media * Advertisements * Online Communities 2. OFFLINE <ul style="list-style-type: none"> * Friends guidance * Parents control * Seeking Professionals * Offline Awareness Events 	Identify strong TR & EM
	4. EMOTIONS: BEFORE / AFTER EM <ul style="list-style-type: none"> * BEFORE: Uncertain, Fear, Anxiety, Frustration, Pessimistic * AFTER: Secure, confident 			

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

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 Form Social logins such as, Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login through Form
FR-4	Reset Password	Sending OTP to Email Sending OTP to Phone Number
FR-5	Dashboard	Add Expense Add Income Set Limit/Budget
FR-6	Insights	Preview Expenditure Plan Graphical chart creation
FR-7	User Logout	No sub requirement

4.2 Non-Functional requirements

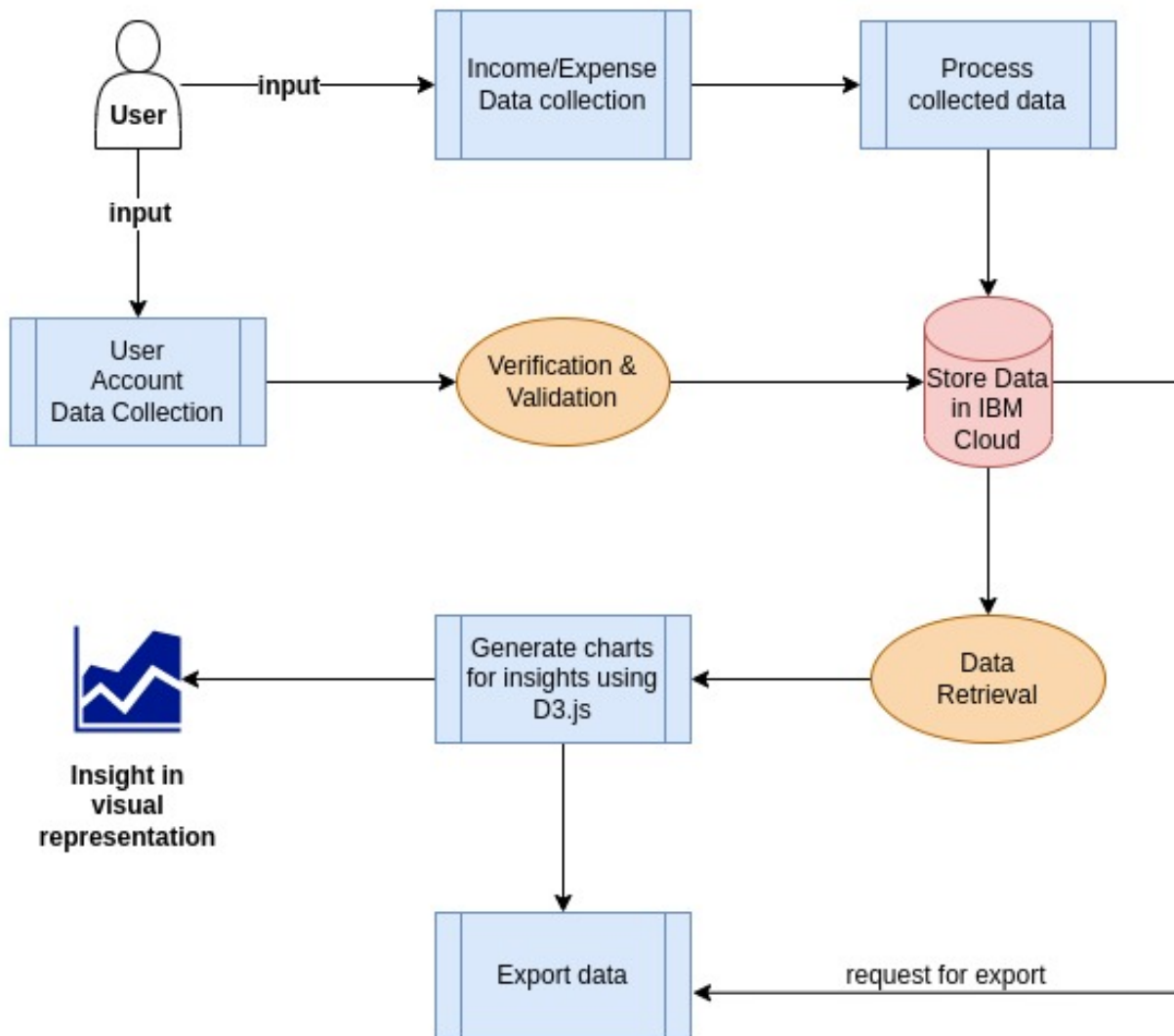
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The error rate of users submitting their details at the dashboard page mustn't exceed 10 percent.
NFR-2	Security	In case of buying the personalized plans, the payment processing gateway must be PCI DSS compliant.
NFR-3	Reliability	The web application must perform in 90% of the use cases
NFR-4	Performance	The landing page supporting 5,000 users per hour must provide 6 second or less response time in a Chrome desktop browser, including the rendering of text and images over an LTE connection.

NFR-5	Availability	The expense tracker dashboard must be available to its users (in India) 99.99% of the time every month during business hours
NFR-6	Scalability	Horizontal scaling is provided by adding more machines to the pool of servers. Vertical scaling is achieved by adding more CPU and RAM to the existing machines. The system must be scalable enough to support 1,000,000 visits at the same time while maintaining optimal performance.

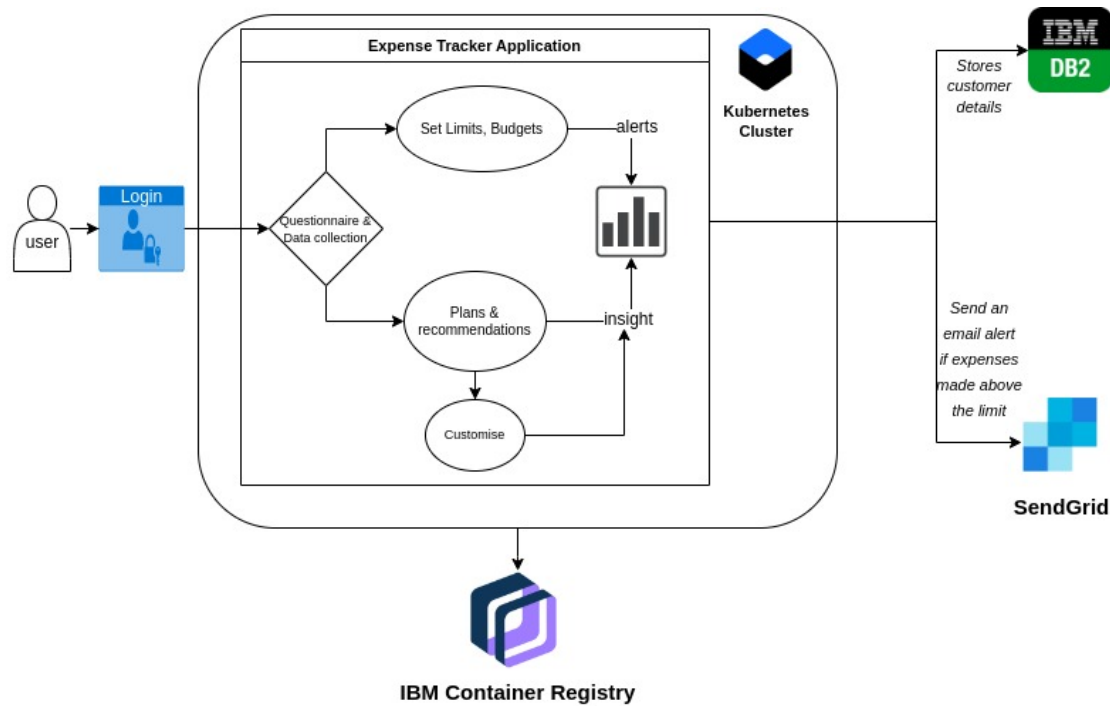
5. PROJECT DESIGN

5.1 Data Flow Diagrams

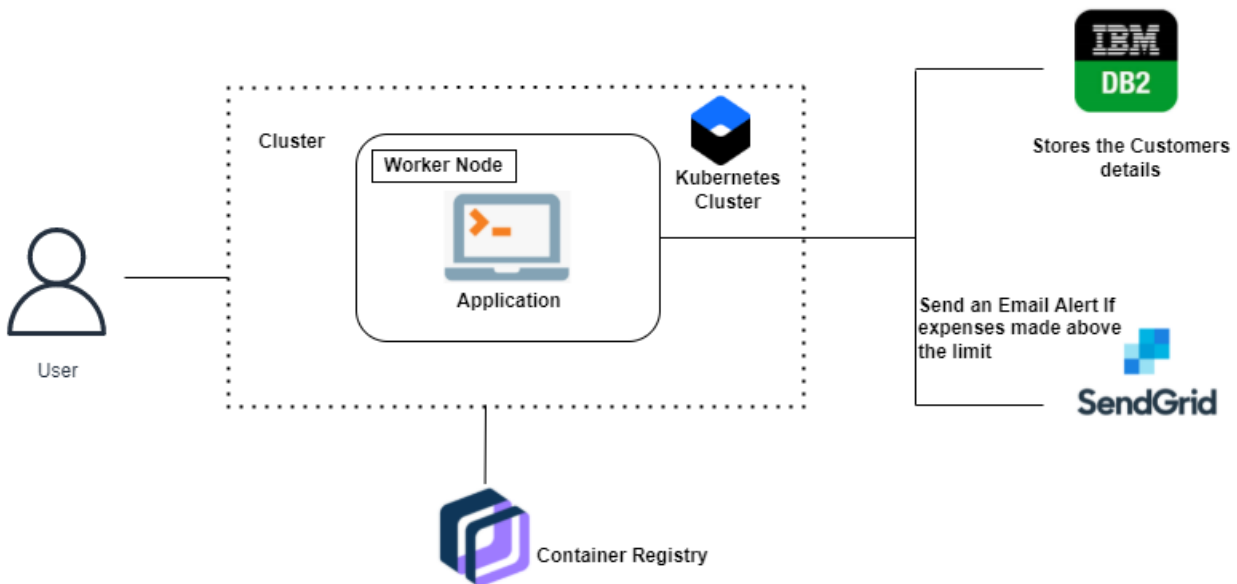


5.2 Solution & Technical Architecture

Solution architecture:



Technical Architecture:



Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The user can interact with the application with the use of Chatbot	HTML, CSS, JavaScript / Angular Js / React Js etc.

2.	Application Logic-1	Register/Login page where user can create account and login into the expense wallet	Java / Python
3.	Application Logic-2	Wallet & Dashboard with provisions to set budget, limits, add expense, income, savings, etc.	IBM Watson STT service
4.	Application Logic-3	Insights in the form of graphs and charts in a standalone page that are visually informative to the user.	IBM Watson Assistant
5.	Database	Collected data from the Chatbot can be stored in Database	MySQL, NoSQL, etc.
6.	Cloud Database	The IBM db2 is the database that will be stored in cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API	An API for sending mails, An API for bot	Sendgrid, IBM Watson Assistant
9.	Machine Learning Model	No ML Model will be used	Nil
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Python's Flask Framework is used for development of web applications specifically in the server side (backend)	Python-Flask
2.	Security Implementations	User data will be Encrypted for security	SHA-256, Encryptions

3.	Scalable Architecture	3 – tier, Micro-services with highly reliable vertical and horizontal scaling provisions	Docker, Kubernetes Cluster
4.	Availability	The application will be available 99.99% of the time (e.g. use of load balancers, distributed servers etc.)	IBM Cloud, IBM Cloud Object Storage
5.	Performance	4000 users per hour with max of 100000 requests per second	Kubernetes,Docker,IBM DB2,IBM Container Registry

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	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	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	Medium	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2

		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Google Login	Low	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can successfully enter the dashboard after Login	High	Sprint-1
	Dashboard	USN-6	As a user, I can Add my expenses & savings	I can receive confirmation on updating expenses per month	High	Sprint-2
		USN-7	As a user, I can set budget/limits to my expenditure	My expense control flow can be blocked when limit hits	High	Sprint-3
	Wallet	USN-8	As a user, I can create wallet	The expense/savings money flow is stored	Medium	Sprint-3
	Insight	USN-9	As a user, I can view my monthly money usage	I can use charts to visualize flow of money	High	Sprint-4

		USN-10	As a user, I can modify my budget/limit from the insight	I can see the change in budget being reflected	Medium	Sprint-4
--	--	--------	----------------------------------------------------------	------------------------------------------------	--------	----------

6. PROJECT PLANNING & SCHEDULING

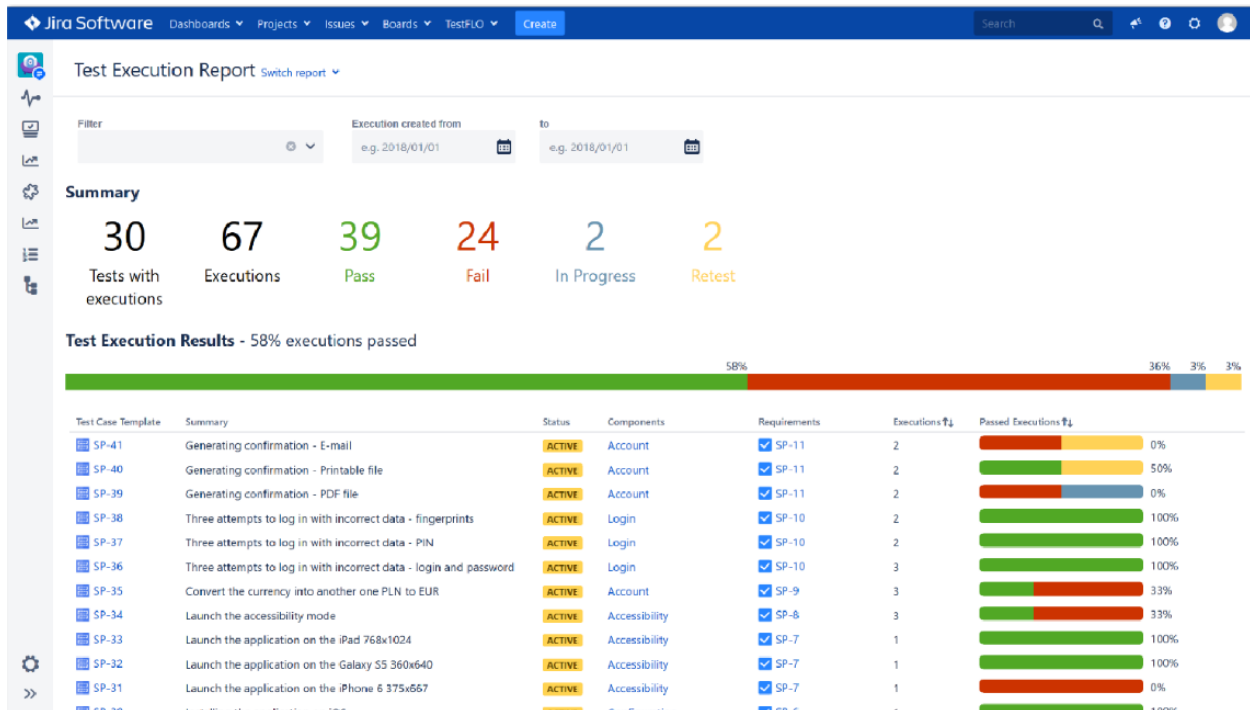
6.1 Sprint Planning & Estimation

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	Upendra, Rithik
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password	8	High	Praneetha, Santhiya
Sprint-1	Validating user	USN-3	Checking whether new user or existing user of the application	4	Medium	Upendra
Sprint-2	Add Expense	USN-4	As a user, I can add the day-to-day expense to the application	8	High	Rithik
Sprint-2	Edit and Delete Expense	USN-5	As a user, I can edit and delete the previously created expense	8	High	Upendra
Sprint-2	Creating time-based filters in history.	USN-6	As a user, I can see the time-based history of expenses.	4	Medium	Santhiya, Praneetha
Sprint-3	Integrating with pie charts for analysis	USN-7	As a user, I can view diagrammatic representation of expenses	8	High	Santhiya, Praneetha
Sprint-3	Enabling limit feature	USN-8	As a user, I can set monthly limit to expenses	4	Medium	Rithik
Sprint-3	Sending Email Alerts	USN-9	As a user, I will receive a mail if I cross a limit	8	High	Upendra
Sprint-4	Testing	USN-9	Testing the application with various tools	10	High	Santhiya, Praneetha
Sprint-4	Deployment	USN-9	Deployment of the application	10	High	Rithik, Upendra

6.2 Sprint Delivery Schedule

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

6.3 Reports from JIRA



7. CODING & SOLUTIONING

7.1 Feature 1: Add Expense

for key in CATS.keys():

 for cat in CATS[key]:

 if request.form['submit'] == "Set {} amount".format(cat):

 _expenditure_userid=User.query.filter_by(username=

username).first().id

 _spent = request.form['amount']

 _where_spent = request.form['location']

 _category_id=Category.query.filter_by(category=

cat).first().id

 _date_of_expenditure = datetime.today()

 _description = request.form['comment']

 expenditure_object = Expenditure(expenditure_userid =

_expenditure_userid, spent = _spent, where_spent= _where_spent, category_id =

_category_id, date_of_expenditure = _date_of_expenditure, description =

_description)

 db.session.add(expenditure_object)

 db.session.commit()

 db.session.close()

 gc.collect()

 flash("Expenditure recorded of {}".format(cat),"success")

 pie_data = [pie_chart([cat for cat in CATS['Daily'] +

CATS['Monthly']],

```

convert_toPercent([calculate_expenditure(category_object.id,
userid=User.query.filter_by(username=username).first().id, today= False) for
category_object in Category.query.all()] , "My Expenditure Distribution this
Month."),pie_chart([cat for cat in CATS['Daily']],
convert_toPercent([calculate_expenditure(category_object.id,
userid=User.query.filter_by(username=username).first().id, today= True) for
category_object in Category.query.all()] ) , "My Expenditure Distribution
today!")]
l
=
[calculate_expenditureBudget_month(userid=User.query.filter_by(username=usern
ame).first().id, month = month) for month in range(1,13)]
exp, budg = zip(*l)
gauge_data = gauge_chart(['{}{}'.format(a,b) for a, b in
zip(months,[' Expenses']*12)], exp, budg)
if Category.query.filter_by(category
=
cat).first().category_daily == True:
return render_template('dashboard.html',CATS = CATS,
html_code = html_code, active_tab = 'expense', isDaily=True, pie_data =
pie_data, gauge_data = gauge_data, user_email = user_email)
else:
return render_template('dashboard.html',CATS = CATS,
html_code = html_code, active_tab = 'expense', isDaily=False, pie_data =
pie_data, gauge_data = gauge_data, user_email = user_email)

```

7.2 Feature 2: Add Budget

```

if request.form['submit'] == "Set Budget":
    _budget_userid = User.query.filter_by(username = username).first().id
    flag = 0
    for obj in Budget.query.filter_by(budget_userid= _budget_userid).all():
        if obj.budget_year == datetime.today().year and obj.budget_month
== datetime.today().month:
            flash("Budget successfully changed for this month! from {}
to {}".format(obj.budget_amount , request.form['amount'], ), "success")
            obj.budget_amount = request.form['amount']
            db.session.commit()
            db.session.close()
            gc.collect()
            flag = 1
    if flag == 0:
        _budget_amount = request.form['amount']
        _budget_month = datetime.today().month
        _budget_year = datetime.today().year
        budget_object = Budget(budget_userid = _budget_userid,
budget_year = _budget_year, budget_month = _budget_month, budget_amount =
_budget_amount)

```

```

        db.session.add(budget_object)
        db.session.commit()
        session['current_budget_id'] = budget_object.id
        db.session.close()
        gc.collect()
        flash("Budget Set!", "success")
        l=calculate_expenditureBudget_month(userid=User.query.filter_by(username=username).first().id, month = month) for month in range(1,13)]exp, budg = zip(*l)
        gauge_data = gauge_chart(['{}{}'.format(a,b) for a, b in zip(months,['Expenses']*12)], exp, budg)
        return render_template('dashboard.html',CATS = CATS, html_code = html_code, active_tab = 'Home', isDaily=True, pie_data = pie_data, gauge_data = gauge_data, user_email = user_email)

```

7.3 Feature 3: View Stat

Generating the chart to show insights on expenses on daily and monthly basis

```

def pie_chart(_categories, _values, _title='Expenditure'):
    pie_chart = pygal.Pie(width=800, height=400)
    pie_chart.title = _title
    for cat, val in zip(_categories, _values):
        pie_chart.add(cat, val)
    return pie_chart.render_data_uri()

def gauge_chart(title_list, val_list, max_valList):
    gauge = pygal.SolidGauge(
        half_pie=True, inner_radius=0.70,
        style=pygal.style.styles['default'](value_font_size=10))
    percent_formatter = lambda x: '{:.10g}%'.format(x)
    rupees_formatter = lambda x: '{:.10g} Rs'.format(x)
    gauge.value_formatter = rupees_formatter
    for title, val, max_val in zip(title_list, val_list, max_valList):
        if max_val == 0:
            max_val = 1
        gauge.add(title, [{'value': int(val), 'max_value': int(max_val)}])
    return gauge.render_data_uri()

```

7.4 Database Schema

Tables

New table



<input type="checkbox"/> Name ▼	Schema	Properties
<input type="checkbox"/> BUDGET	BWZ22240	...
<input type="checkbox"/> CATEGORIES	BWZ22240	...
<input type="checkbox"/> EXPENDITURES	BWZ22240	...
<input type="checkbox"/> USERS	BWZ22240	...

USERS

Approximate 2 rows (32.0 KB)
Updated on 2022-11-18 13:33:07

Name	Data type	Nullable	Length	Scale	
ID	INTEGER	N		0	
USERNAME	VARCHAR	N	80	0	
EMAIL	VARCHAR	N	120	0	
password	VARCHAR	N	120	0	

CATEGORIES

Approximate 9 rows (32.0 KB)
Updated on 2022-11-16 11:32:58

Name	Data type	Nullable	Length	Scale	
ID	INTEGER	N		0	
CATEGORY	VARCHAR	Y	64	0	
CATEGORY_DAILY	SMALLINT	Y		0	
CATEGORY_PRIMARY	SMALLINT	Y		0	

BUDGET

Approximate 1 rows (32.0 KB)
Updated on 2022-11-16 11:33:10

Name	Data type	Nullable	Length	Scale	
ID	INTEGER	N		0	
BUDGET_AMOUNT	DECIMAL	Y	15	2	
BUDGET_USERID	INTEGER	Y		0	
BUDGET_MONTH	INTEGER	Y		0	
BUDGET_YEAR	INTEGER	Y		0	

EXPENDITURES

Approximate 0 rows (0 KB)
Updated on 2022-11-16 11:12:39

Name	Data type	Nullable	Length	Scale	
ID	INTEGER	N		0	👁
CATEGORY_ID	INTEGER	Y		0	👁
SPENT	DECIMAL	Y	15	2	👁
DATE_OF_EXPENDITURE	TIMESTAMP	Y	10	6	👁
EXPENDITURE_USERID	INTEGER	Y		0	👁
WHERE_SPENT	VARCHAR	Y	100	0	👁
DESCRIPTION	LONG VARGRAPHIC	Y	16350	0	👁

Here is the database model implemented in python:

```
class User(db.Model):
    id=db.Column(db.Integer, autoincrement=True, primary_key=True, nullable=False)
    username = db.Column(db.String(80), nullable=False)
    email = db.Column(db.String(120), nullable=False)
    password = db.Column(db.String(120), nullable=False)
    def __repr__(self):
        return '<User {}>'.format(self.username)

class Category(db.Model):
    id = db.Column(db.Integer, autoincrement=True, primary_key=True)
    category = db.Column(db.String(64))
    category_daily=db.Column(db.Boolean, default=False) # is it daily expense related,
False implies, it can be both daily and monthly!?
    category_primary=db.Column(db.Boolean, default=False) # if not true, it means ,
this category is added explicitly by user!

class Budget(db.Model):
    id = db.Column(db.Integer, autoincrement=True, primary_key=True)
    # the data type of the budget should match the data type of the price
    budget_amount = db.Column(db.Numeric(15, 2))
    budget_userid = db.Column(db.Integer, db.ForeignKey('users.id'))
    budget_month = db.Column(db.Integer)
```

```

budget_year = db.Column(db.Integer)
user = db.relationship("User", backref=db.backref('budget'))
def __repr__(self):
    return "<Budget id=%s budget=%s budget_userid=%s budget_month=%s
budget_year=%s>"%(self.id, self.budget_amount, self.budget_userid, self.budget_month,
self.budget_year)

class Expenditure(db.Model):
    id = db.Column(db.Integer, autoincrement=True, primary_key=True)
    category_id = db.Column(db.Integer, db.ForeignKey('categories.id'))
    spent = db.Column(db.Numeric(15, 2), default=0)
    date_of_expenditure = db.Column(db.DateTime)
    expenditure_userid = db.Column(db.Integer, db.ForeignKey('users.id'))
    where_spent = db.Column(db.String(100))
    description = db.Column(db.UnicodeText)
    user = db.relationship("User", backref=db.backref('expenditures'))
    category = db.relationship("Category", backref=db.backref('expenditures'))

```

8. TESTING

8.1 Test Cases

1. Login Page (Functional)
2. Login Page (UI)
3. Add Expense Page (Functional)
4. Add Budget Page (Functional)
5. Expense Stats (UI)
6. Security
7. Exception Reporting
8. Version Control

8.2 User Acceptance Testing

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

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	4	2	1	0	7
Duplicate	1	0	1	0	2
External	3	0	0	1	4
Fixed	4	0	0	3	7

Not Reproduced	0	1	0	0	1
Skipped	1	1	1	1	4
Won't Fix	0	2	2	4	8
Totals	13	6	5	9	33

Test Case Analysis: This report shows the number of test cases that have passed, failed and untested

Section	TotalCases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	39	0	0	39
Security	6	0	0	6
Outsource Shipping	3	0	0	3
Exception Reporting	11	0	0	11
Final ReportOutput	8	0	0	8
Version Control	4	0	0	4

9. RESULTS

9.1 Performance Metrics

Performance Metrics

First Contentful Paint How quickly content like text or images are painted onto your page. A good user experience is 0.9s or less.	Good - Nothing to do here 524ms	Time to Interactive How long it takes for your page to become fully interactive. A good user experience is 2.5s or less.	Good - Nothing to do here 717ms
Speed Index How quickly the contents of your page are visibly populated. A good user experience is 1.3s or less.	OK, but consider improvement 1.4s	Total Blocking Time How much time is blocked by scripts during your page loading process. A good user experience is 150ms or less.	Good - Nothing to do here 0ms
Largest Contentful Paint How long it takes for the largest element of content (e.g. a hero image) to be painted on your page. A good user experience is 1.2s or less.	Longer than recommended 1.9s	Cumulative Layout Shift How much your page's layout shifts as it loads. A good user experience is a score of 0.1 or less.	Good - Nothing to do here 0.01

Browser Timings

Redirect	0ms	Connect	154ms	Backend	158ms
TTFB	312ms	First Paint	525ms	DOM Int.	717ms
DOM Loaded	718ms	Onload	1.9s	Fully Loaded	2.2s

10. ADVANTAGES & DISADVANTAGES

Advantages:

- 1) **Prioritize Your Spending:** Remember you are on a fixed income and have a limit to spend which you cannot cross. If you start examining what you are spending your hard-earned money on, you will be able to prioritize the spending. This way, you will spend just on things you need, like paying your EMIs, utility bills, rent, and grocery shopping, rather than spending frivolously.
- 2) **Become Aware of Poor Spending Habits:** If you tend to spend money on a whim, using an expense tracker will help you identify those habits. When you will spend more than what you have thought to save, then there will be a color change of the card which will let you know that you should spend carefully. Basically, you will think twice before doing useless expenses.
- 3) **Expenses Reminder:** You may forget to add expenses on some day as it is human nature to forget about some things. So, an expense tracker will help you to remind about adding expenses daily. It can send you reminder either through email or SMS.
- 4) **Take Control of Your Finances:** When you track your expenses, you are taking control of your finances. It lets you to regulate spending impulses and eliminate worthless spending, thereby avoiding debt. At every point, you will be aware about how much money you are left with.
- 5) **Saving and Investment:** When you track your expenses, you are aware about the area where you can save better and invest in your future. Spending carelessly does not give you leeway to

save and invest for your future.

Disadvantage:

The only con is that we have to manually add the data so if we forget to add some expense then it is difficult for us to track the amount spent on various things.

11. CONCLUSION

Daily spending tracking can not only help you save money, but it can also help you set financial objectives for the future. If you know exactly where your money goes each month, you can quickly see where you can make some savings and compromises. We have designed a project that is more efficient than other income and expense trackers. The project succeeds in avoiding manual calculations for estimating monthly revenue and expenses. The modules have been designed to be both efficient and appealing.

12. FUTURE SCOPE

The system can be integrated with credit cards, debit cards, bank accounts. It can also be integrated with application such as bookmyshow, IRCTC, SCTC, etc. So that transaction can be directly entered into the system without the need for manual input. Since corporate cards are linked to the software, it makes it easier to reconcile credit card statements with expense reports.

13. APPENDIX

Source Code: [Final Project Source Code](#)

GitHub: [IBM-Project-11742-1659343364](#)

Project Demo Link: [Video Demonstration](#)