Software Requirements Specification

for

TrackIt

Version 1.0

Prepared by

Group #: 9 **Group Name: localghosts:3000**

Sarthak Kalankar	210935	sarthakk281103@gmail.com
Maurya Aryan Swaminath	210595	aryanmaurya383@gmail.com
Saugat Kannojia	210943	saugatkannojia@gmail.com
Harsh Oza	210411	2003ozaharsh@gmail.com
Padulkar Rohan Ravikumar	210689	rohan1.padulkar@gmail.com
Sujal Lalawat	211066	sujallalawat223@gmail.com
Depanshu Sahu	210316	depanshusahu057@gmail.com
Rashi Gupta	190690	grashi253@gmail.com
Shrilakshmi S K	211012	shrilakshmisk13@gmail.com
Chirayush Mohanty	210289	chirayushimo@gmail.com

Course:	CS253
Mentor TA:	Abhinav
Date:	27/01/2023

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1 Introduction

1.1 Product Scope

You would have seen your parents or elders keeping track of how much money they spent on monthly or yearly basis to get an idea of their monthly or yearly savings. Keeping this in mind we have made a web app to ease the hassle of keeping the record or tracking physically or manually typing the entries in excel or similar apps and then using formulas to calculate the total sum spent and all. Using the app any common person whether adult or student will be able to upload their entry on the web app and get the analytics of how much amount they spent and on what like Food, Travel, Groceries, Shopping, etc. monthly.

One of the other benefits of this app would be that the person can access it from any device as it's a web app and will save time in making entries and keeping records. They won't have to keep saving the receipts or tickets for their purchases. They would be able to directly enter it. We hope that **TrackIt** will help most people in keeping a track of their spending.

1.2 Intended Audience and Document Overview

The document is mainly intended for the product end-users (to understand product requirements and give suggestions) and the developers (to build the product as per requirements). The user should start with the introduction, then he/she could read the product overview section (2.1) followed by specific requirements section (3). The developers would be mainly interested in specific requirements section (3) and non-functional requirements section (4).

1.3 Definitions, Acronyms and Abbreviations

1.3.1 Abbreviations

- IITK- Indian Institute of Technology, Kanpur
- OTP- One Time Password
- SQL- Structured Query Language
- UI- User Interface
- DB - Database

1.4 Document Conventions

● General Text:	Font-Arial	Size-12	
● Heading:	Font-Arial	Size-18	
● Subheadings:	Font-Arial	Size-14	Style-Bold
● Margin - 1''			

1.5 References and Acknowledgments

None

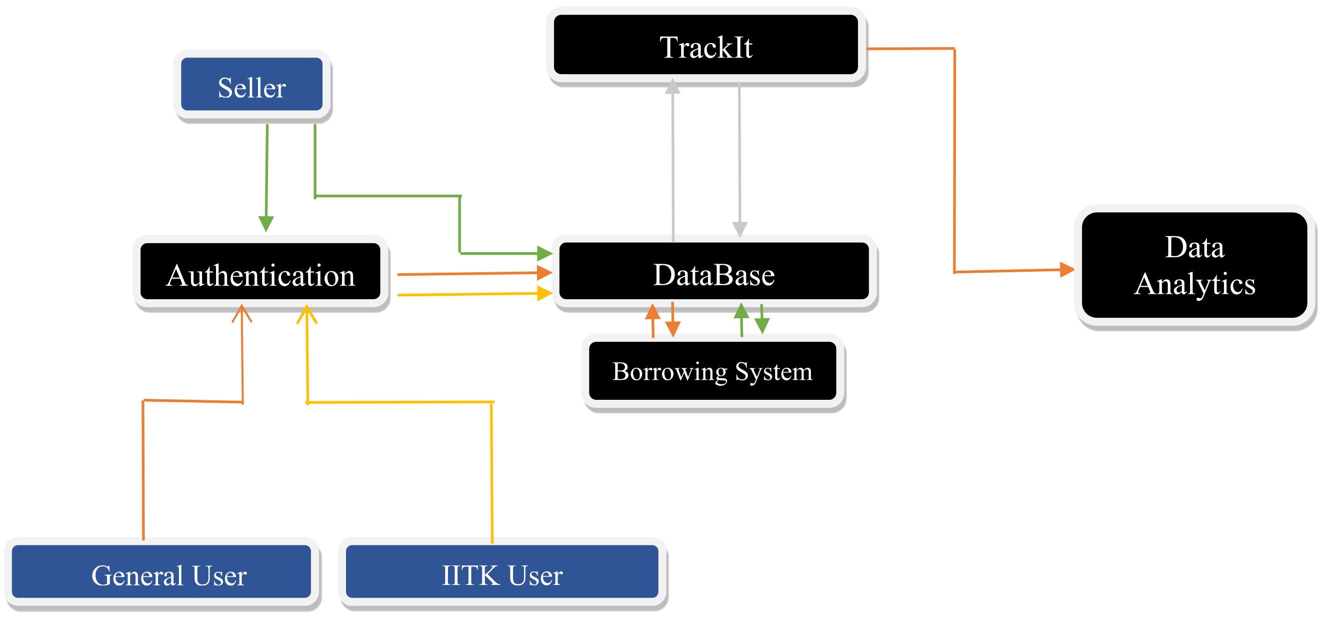
2 Overall Description

2.1 Product Overview

TrackIt is a web-based application where the users can track their timely expenses and manage their expenditure accordingly. Our goal is to develop a user-friendly expense management system that eases the record-keeping troubles that a user might face. It allows the user to categorise their day-to-day expenses after every transaction. After certain intervals of time, the users will be provided with the analytics of their expenditure, helping them keep a track of it. They can also see their analytics on their dashboard throughout the month.

The seller section will keep track of the borrowings and amount lent by them to their customers. The borrowing will be a two-way process. While the seller is adding any borrowing transaction, the customer will also get a prompt to confirm the same. The same will be the case when the money is paid back by the customer.

2.1.1 Context Diagram



2.2 Product Functionality

- Email Authentication by Sign In/ Sign Up page.

- Addition of records by the customers keeping the tracks of amount, date, and the category of expense (food, travel, health etc.)
- Daily/Monthly analysis of their expenditure using statistics and graphs.
- Borrowing system for sellers who are willing to use the feature.
- Feature to set a monthly limit on the expenditure with a Progress Bar
- Getting insights on various categories on the basis of income, if provided

2.3 Design and Implementation Constraints

- We have a specified memory in the server, so the number of users is constrained by an upper bound.
- The number of years as history a user can access is limited due to storage constraints.
- As of now the webapp is in English only.
- Word limit on the length of description a user can write.

2.4 Assumptions and Dependencies

- There is no simultaneous editing of data from different devices. It may lead to data loss.
- The updating is done by the user on the day of transaction for correct analytics.
- The input records entered are of valid datatypes.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

A user just needs familiarity with navigation in browsers to be able to understand all the functions provided by our system.

Right after logging in the software, the first page will be our home page or the current month page. The “Entries” button under the “Current Month” button on the navigation bar will lead to this page. The numerous cards visible on the main screen will help the user track all expenses made on any particular date during that month. Every card will display a description of a payment, the category it falls under, and the amount spent. The Filter card to the right of the page will display the present date, month, amount spent in different categories and total amount spent throughout the month.

The “Borrowing” button on the navigation bar will open a new page. Every card on the page will show the date, the shop name, description, and amount of money the user owes

to any shop. The filter card to the right shows the present date, month, and total money the user owes to all the shops for that month.
The "History" button on the navigation bar will lead to a page where the user can track all payments made during previous months. Every card will display the month, money spent on different categories, total money spent, total money owed to others. The drop-down function of these cards will display a detailed record of that month.
"Dashboard" button will open a page of detailed analysis of the user's expenses. This page will have detailed graphs, doughnut charts, borrowed money records, and summary of the money spent. It serves the purpose of expense analysis.
"Sign Out" button will sign out of the current user's account and return to the login/sign up page.

3.1.2 Hardware Interfaces

Clients must have the required devices and an internet connection to browse the website.

Server Side - We will be requiring a web server to host our website.

3.1.3 Software Interfaces

Client Side - an operating system (Windows, macOS, Linux, etc.), a web browser

Server side - Web server, Database

3.2 Functional Requirements

3.2.1 Potential Users

1. The system expands its service of expense management to the campus community as well as to people outside campus.
2. The system provides additional features of borrowing to the campus community.
3. The system also provides service to the merchants inside campus to take a note of all their payments/dues inside campus.

3.2.2 Maintain user profile

1. The system allows users to create their profiles and set credentials.
2. The system allows users to update their personal information.
3. The system allows users to change passwords.

3.2.3 Email Confirmation

1. The system stores the email-ids of all its users.
2. Campus community can only use their IITK email-ids to make a profile.

3.2.4 Addition of daily expense by consumers

1. The system allows the IITK and non-IITK consumers to add their daily expenses into the system which gets stored in the system database and used to give the users various insights.
2. The users can select the category for their expenses like food, health, travel and shopping.
3. The system allows the users to enter the amount spent with a small description(optional).
4. The users can also upload a pic of the receipt/bill from their device or camera.

3.2.5 Borrowing feature for IITK users inside campus

1. The system allows an additional feature of borrowing/later payment for the campus community from those shops that allow this feature from their side.
2. The user can add the amount along with the merchant's name and request for borrowing/later payment through the system, if the merchant accepts, this information gets stored in the database and reflected in the “BORROWING” section.
3. The system provides timely notification to the consumers regarding the borrowings/pending payments.
4. The system also provides the list of all borrowings to the user with option to remove if paid.

3.2.5 Provides analytics

1. The system provides weekly and monthly analytics of total expenditures in the form of pie charts and graphs.
2. The system provides a feature to set the daily/weekly/monthly limits on total expenses.
3. The system provides analytics of expenses spent in different categories.
4. The system compares the user's monthly expense with the average expense of all users.

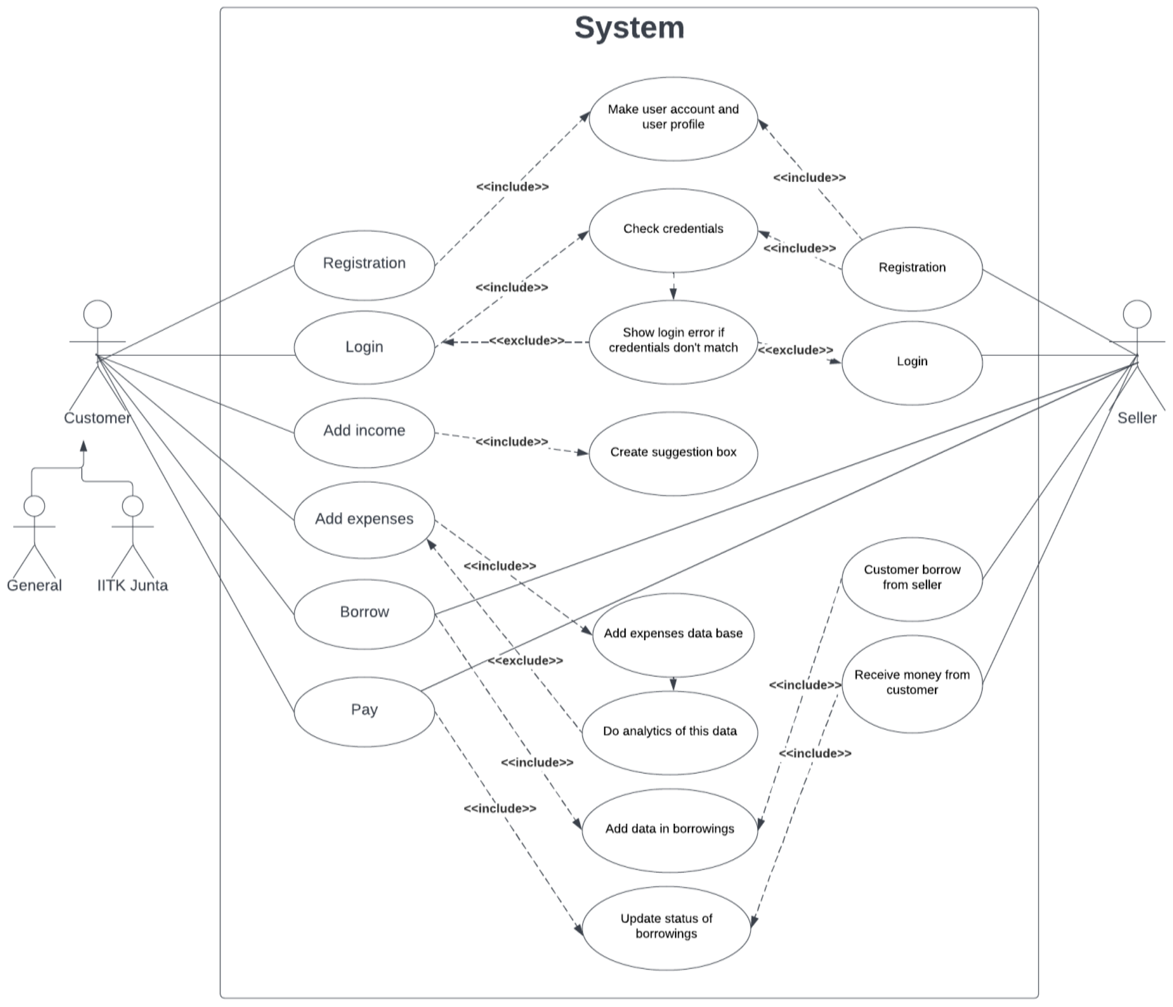
3.2.6 Provides past history

1. The system provides past history of all the expenses done by the user.
2. The system provides a filter where the user can select a particular year, month, date, and category to see those specific entry/entries.
3. The user can also see the bills/receipts they attached while adding it to the system along with description.

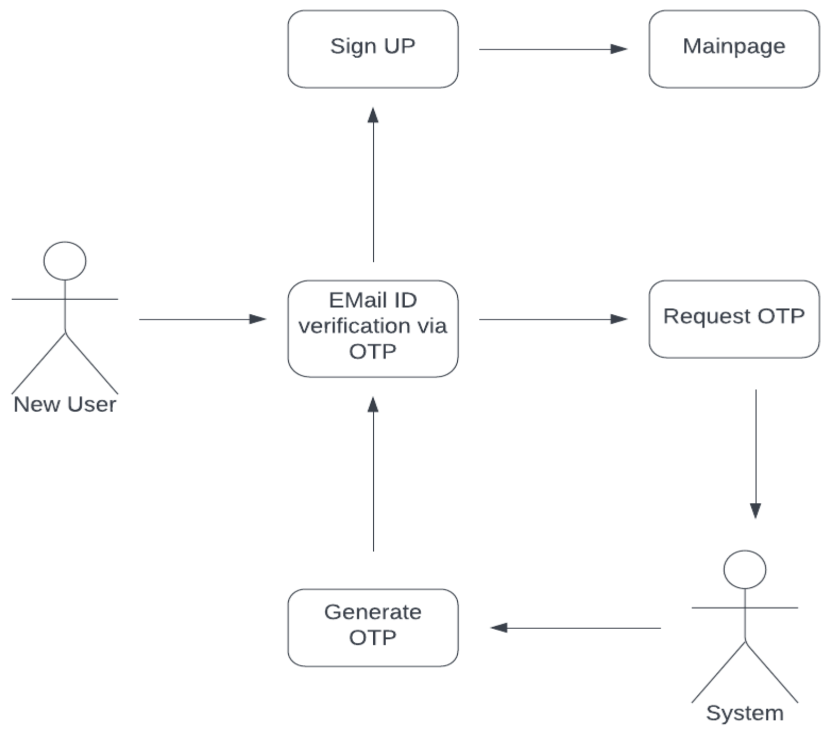
3.2.7 Features for the merchants

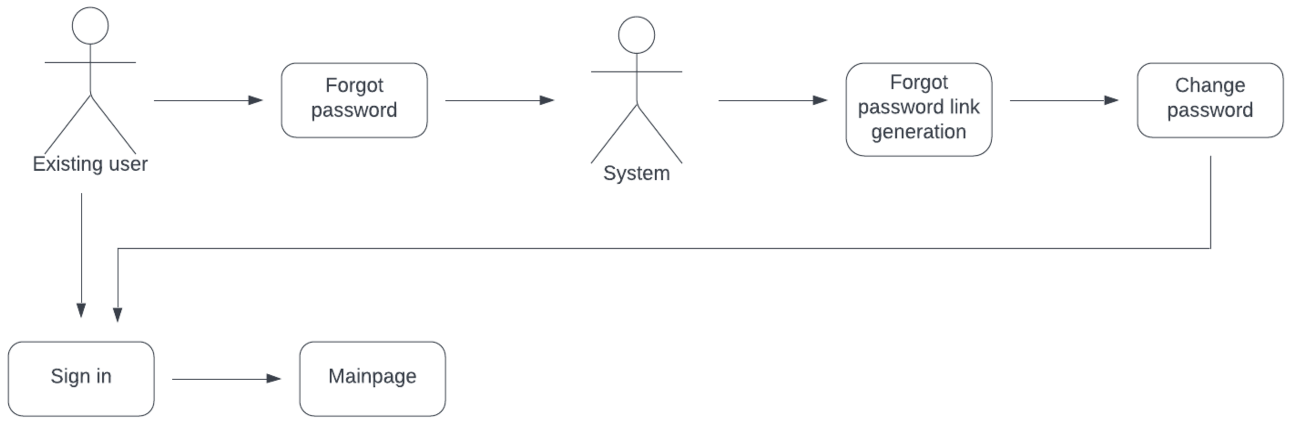
1. The system provides the information of all the people who have borrowed money from the merchants inside campus.
2. The system also provides the date on which particular transaction took place so that the merchants can keep a record of all borrowings.
3. The system provides a filter with which the merchants can select the day, week, month, etc. to see the pending borrowings.
4. The system allows the merchants to mark paid/unpaid to borrowing and shows its status accordingly.
5. The system also allows merchants to add descriptions on the borrowings.
6. The system provides a feature by which merchants can set deadlines for the payments and the respective user gets timely informed.
7. The merchant can also see the total amount of borrowings of any month.

3.3 Use Case Model



3.3.1 Use Case #1 (U1- Authentication)





Author – Chirayush Mohanty

Purpose - To authenticate valid users

Requirements Traceability – Sign up interface, Login interface, Reset password interface, DB

Priority - High

Preconditions - The user must have a valid Email ID.

Post conditions - The user can login and interact with the system.

Actors – Human, System

3.3.2 Use Case #2 (U2- Logging the amount spent)

Author – Rohan Padulkar

Purpose - To provide an interface where users can enter the details of their expense into the system and the system stores it in the database.

Requirements Traceability – Main Page, user profile, interface to add expense, database

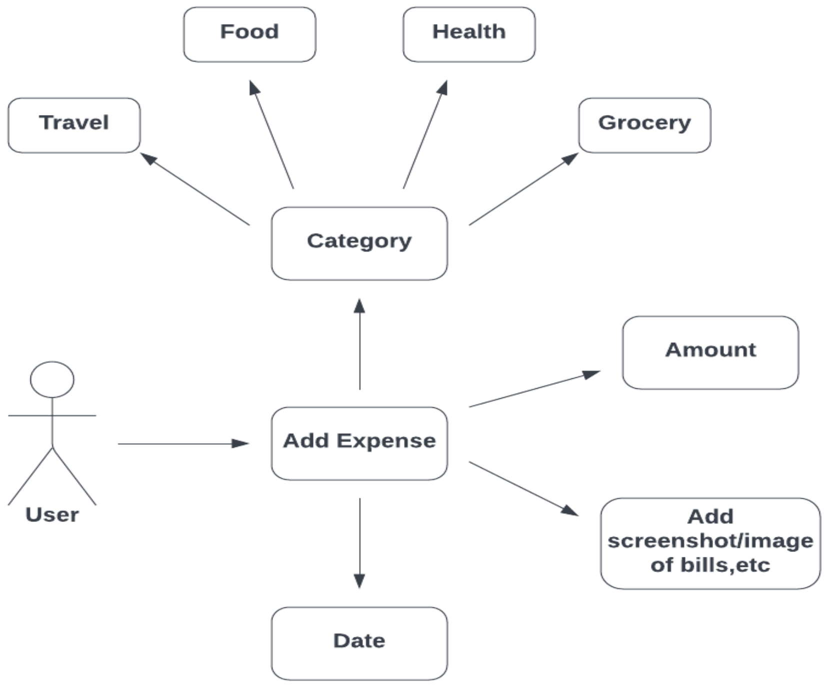
Priority - High

Preconditions - Users must be authenticated.

Post conditions -Users can access the details of their expenses made on any day/week/month.

Actors – User, system

Includes U1



3.3.3 Use Case #3 (U3 - View analytics)

Author – Sujal Lalawat

Purpose - This provides an interface where the users get reports on their spendings and the analytics of their daily/weekly/monthly expenditures in the form of graphs and pie charts. Users also get analytics of their expenditures in different categories.

Requirements Traceability – Main page, Analytics page, Database

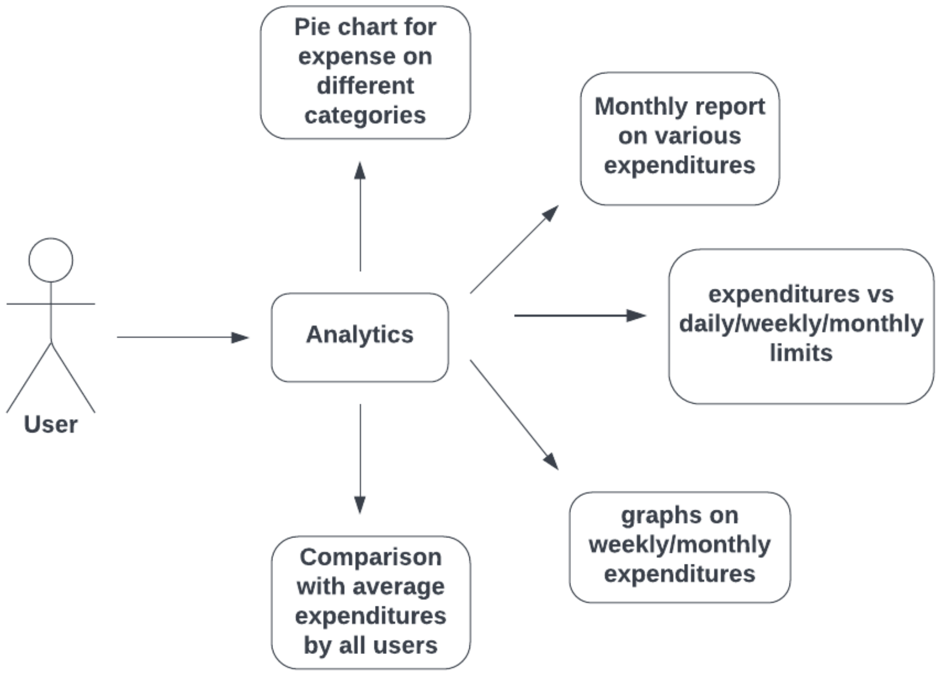
Priority - High

Preconditions - Users must be authenticated.

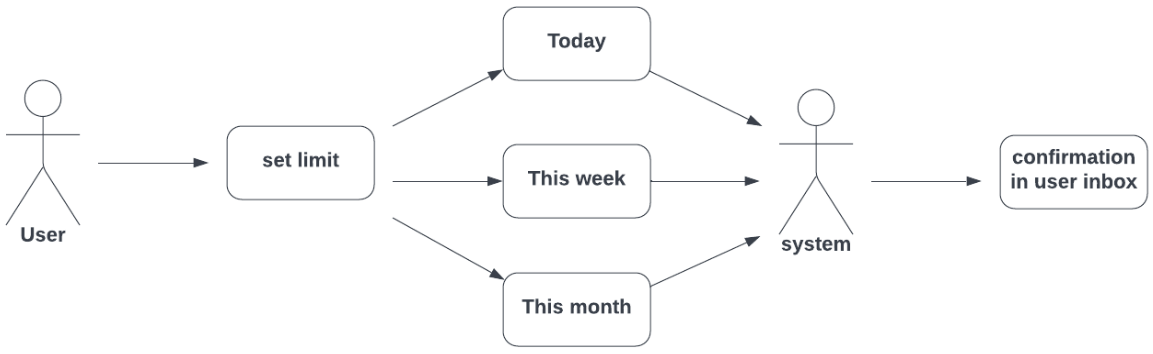
Post conditions -Users can get various insights on their expenditures across different categories in various timelines.

Actors – User, system, database.

Includes U1 and U2


Pie chart for expense on different categories
Monthly report on various expenditures
User
Analytics
expenditures vs daily/weekly/monthly limits
graphs on weekly/monthly expenditures
Comparison with average expenditures by all users

3.3.4 Use Case #4 (U4 - Setting limits)



Author – Harsh Oza

Purpose - to provide an interface to the users where they can set limits to their daily/weekly/monthly expenditures,

Requirements Traceability – main page, limit interface, user inbox.

Priority - Medium

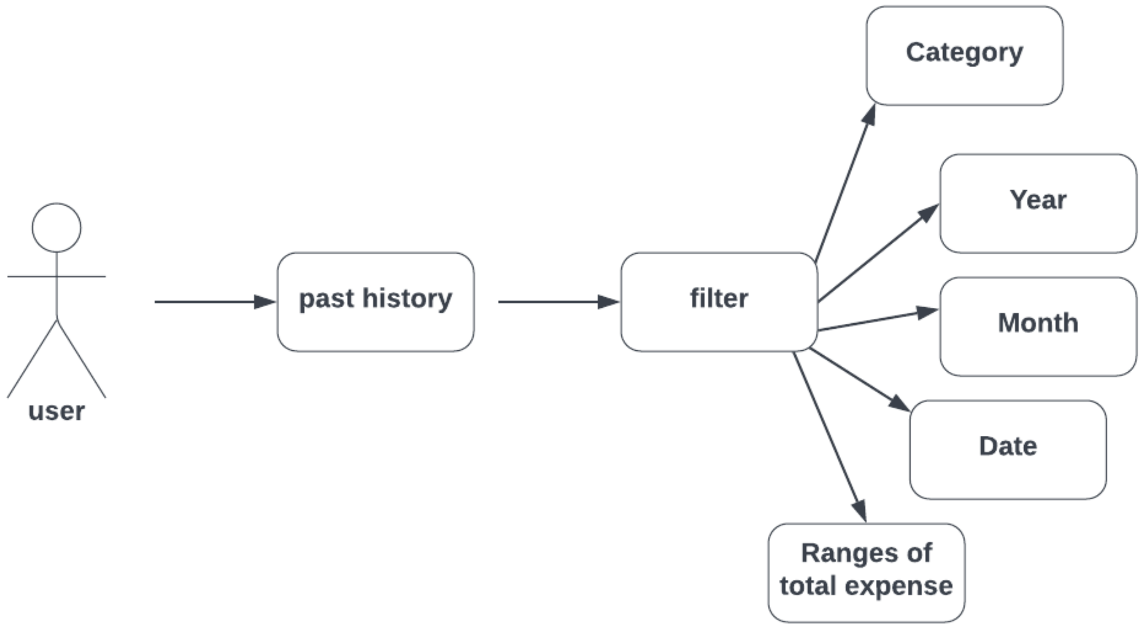
Preconditions - users must be authenticated.

Post conditions - on reaching closer to the limit set by the user, the system notifies the user through the user inbox.

Actors – User, system, database

Includes U1

3.3.5 Use Case #5 (U5- Applying filters)



Author – Chirayush Mohanty

Purpose - To provide an interface to the users where they can apply filters like date, month, year, category etc. to see their current/past expenditures along with their descriptions,

Requirements Traceability – past history page, filter interface, DB

Priority - High

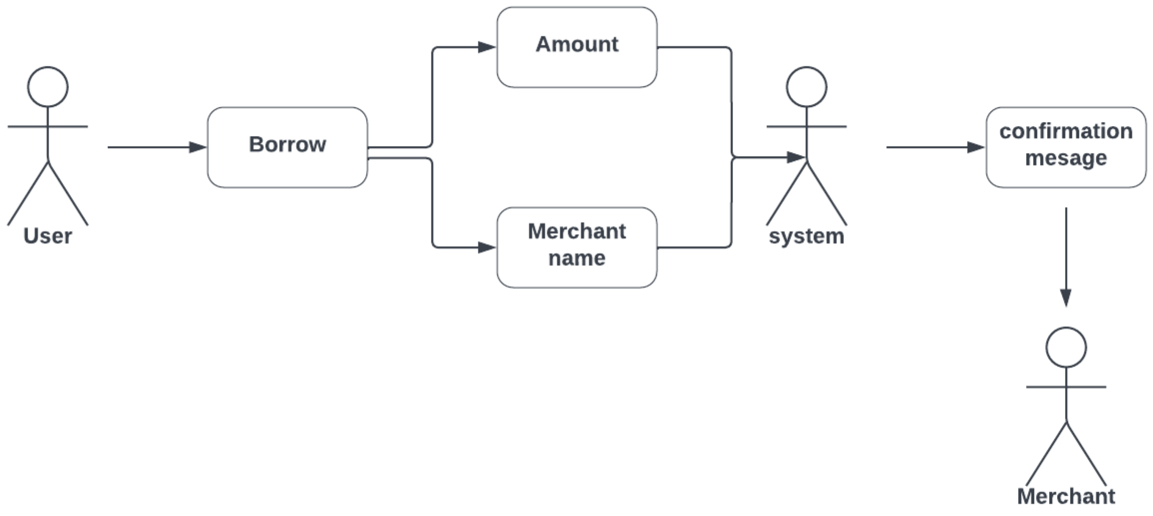
Preconditions - Users must be authenticated.

Post conditions - users get reports of their past expenditures in different categories to modify/compare with current spendings and limits.

Actors – User, system, database

Includes U1, U2

3.3.6 Use Case #6 (U6- Borrowing from merchant)



Author – Rohan Padulkar, Shrilakshmi S K

Purpose – To provide an interface for user (IITK customer) to borrow money from IITK merchants over the purchased items.

Requirements Traceability – Borrowings page, Merchant home page, database

Priority - High

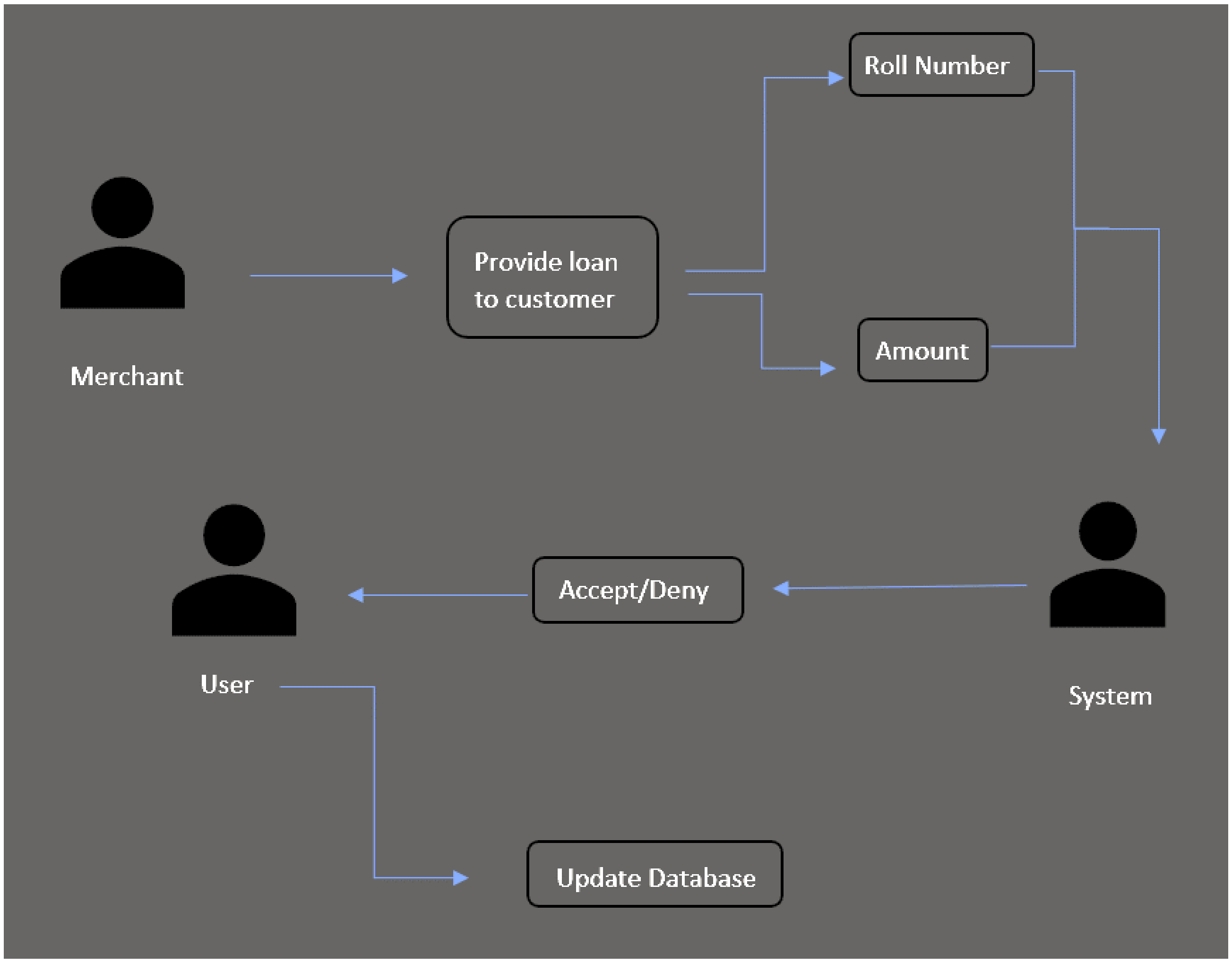
Preconditions – User must be authenticated and must hold an IITK Roll No.

Post conditions – The borrowed amount is updated in database. User will be able to see the amount they owe to different merchants.

Actors – customer, merchant, system, database

Includes U1, U2

3.3.7 Use Case #7 (U7- Providing loan to customer)



Author – Chirayush Mohanty, Shrilakshmi S K

Purpose – Allows the merchant to let customer borrow money as per the request from the customer.

Requirements Traceability – Interface for merchant to enter the roll number of customer, interface to enter the amount to be borrowed by customer, interface for customers to accept/ deny the borrowing offered by merchant, database.

Priority - High

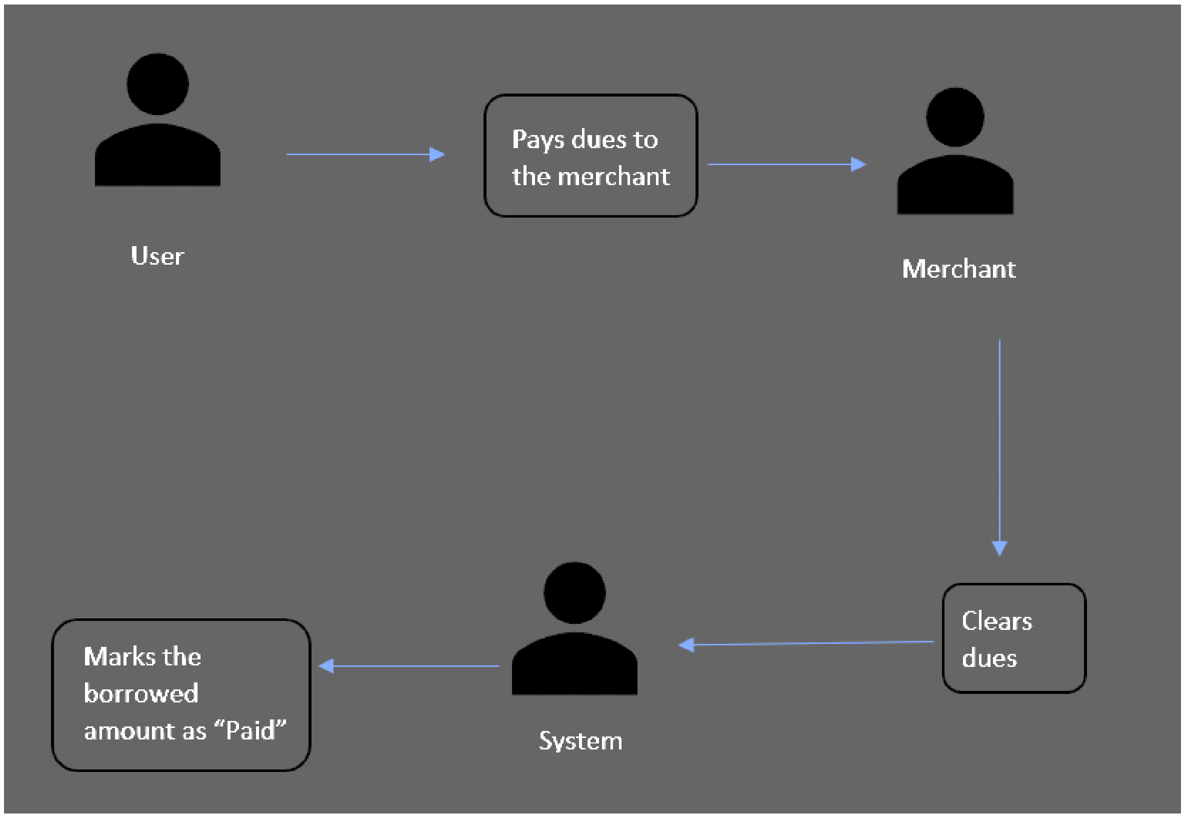
Preconditions – Merchant must be providing products/ services in IITK campus. Merchant can only provide loan to customer holding IITK Roll No.

Post conditions – A notification will be sent to customer and customer must accept it for the loan to be approved. If customer denies, merchant would not be able to provide loan to customer and database would not be updated.

Actors – customer, merchant, system, database

Includes U1, U6

3.3.8 Use Case #8 (U8 – Clearing dues of customers)


User
Pays dues to the merchant
Merchant
Clears dues
System
Marks the borrowed amount as "Paid"

Author – Shrilakshmi S K

Purpose – When customer pays the amount that they have borrowed from merchant, the merchant will be able to clear the customer's dues.

Requirements Traceability – database, interface to see the dues of all customers, interface to clear the dues.

Priority – High

Preconditions – User must be authenticated and must hold an IITK Roll No. User must have borrowed money from said merchant and must have cleared the dues.

Post conditions – Customer will also be able to see if merchant has cleared the dues. The borrowed amount will be marked as “Paid” in the database.

Actors – customer, merchant, system, database

Includes U1, U6, U7

4 Other Non-functional Requirements

4.1 Performance Requirements

PR-1: System should response within 1 second and while adding data to database it should be done and showed to user by not more than 2 seconds.

PR-2: All Web pages generated by the system shall be fully downloadable in no more than 10 seconds over a 500KBps modem connection.

4.2 Safety and Security Requirements

SE-1: Users shall be required to log in to the system for all operations.

SE-2: The system shall permit only the shopkeepers to verify that customer has paid the arrear dues.

SE-3: The system shall permit shopkeepers to view only the data related between user and shopkeeper and not of other shopkeepers.

SE-4: The system shall permit users to view data only related to them and not someone else's data.

SE-5: The system shall allow the password to be recovered via a password recovery mail that is valid for thirty minutes from the time the password is sent.

4.3 Software Quality Attributes

4.3.1 Flexibility

The web app is to be designed so that it is flexible. Also, it allows the incorporation of new requirements in any module of the system. The application will be designed in a modular format such that any future changes (additions or deletions) will be easily incorporated into the system. Each functionality is divided into several folders so make it flexible.

4.3.2 Portability

The application will be easily portable on any window-based system. The website front is designed using React JS, making it a responsive and progressive web app, which ensures that the application can run on different platforms.

4.3.3 Maintainability

The architecture, design, implementation, and documentation of the product should minimize the maintenance labour of the software system. The maximum person-time required to fix a security defect (including regression testing and documentation update) must not exceed two persons per day.

Otherwise, the software system must be taken offline, or the offending feature disabled which should only be done in case of emergencies. The average person-time required to make a minor enhancement (including testing and documentation update) must not exceed one person per week.

4.3.4 Usability

The frontend of the software application is designed to be user-friendly such that the user can utilize the system effectively. It will be based on the well-known principle of usability -As Simple as Possible. Also, after logging in the user can view his account without logging in again even after the website is closed and reopened.

4.3.5 Security

The application is password protected and any upgradation of new entries and deletions is done by only privileged users. All the users will be able to log on to their accounts using their CC username and custom password (created by them). This password will be saved in an encrypted (hashed and salted) format in the database to ensure security. Also, the password is ensured to be strong enough while signing up, to ensure if someone gets an encrypted password, it will too be difficult to decrypt.

4.3.6 Reliability and Availability

Web app will store data in MongoDB Atlas Cloud which is reliable and always available. The system will always be running and ready to carry out its task whenever the user needs it.
Also, the time to exchange data between web app and database is less which will give users a smooth experience.

5 Other Requirements

*<This section is **Optional**. Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

Appendix A – Data Dictionary

Abbreviation	Abbreviation definition
hashed	It is a method of encryption of data such that it converts password into alphanumeric string of constant length.
Salting	A string which is added to data during its encryption to make hashed string difficult to decrypt.
Server	It is a computer that can be accessed remotely.
Database (DB)	It is an organized collection of structured information stored in the server.
User Interface (UI)	By which the user and the computer system interact, by use of input and output devices.
OTP	One Time Password, string which is sent to user for verifying user.

Appendix B - Group Log

SL. No.	Date	Timings	Venue	Description
1.	06/01/2023	6:00 PM –7:00 PM	CCD	Brainstormed various project ideas
2.	11/01/2023	6:00 PM –7:00 PM	CCD	Finalised our project idea
3.	16/01/2023	6:00 PM –7:00 PM	CCD	Discussed the features, requirements, and software we will work on.
4.	17/01/2023	6:30 PM –7:00 PM	Microsoft Teams	Discussed our project with our mentor Abhinav
5.	23/01/2023	6:00 PM –7:00 PM	CCD	Distributed work and decided UI of the project
7.	25/01/2023	10:00AM- 11:00AM	Microsoft Teams	Discussed about the specific work each is doing in the SRS document.
8.	25/01/2023	6:00PM- 6:30PM	CCD	Discussed about SRS document.
9.	27/01/2023	5:30PM- 6:30PM	CCD	Finalised the SRS document.