SOFTWARE Requirement Specification

for

DBMS

Version 2.1

Prepared by

Group #: 5 **Group Name: DevDynamos**

Vishal Kumar	221201	vishalkmr22@iitk.ac.in
Yash Pratap Singh	221223	yashps22@iitk.ac.in
Shriya Garg	221038	shriyag22@iitk.ac.in
Aditya Gupta	220065	adityagu22@iitk.ac.in
Abhishek Kumar	220041	kumara22@iitk.ac.in
Nandini Akolkar	220692	anandini22@iitk.ac.in
Rishikesh Sahil	220892	rishikeshs22@iitk.ac.in
Udbhav Singh Sikarwar	221150	udbhavss22@iitk.ac.in
Anshu Yadav	220171	anshuyadav22@iitk.ac.in
Kushagra Singh	220572	kushagras22@iitk.ac.in

Course: CS253

Mentor TA: Bharat

Date: 22/04/2024

Revisions

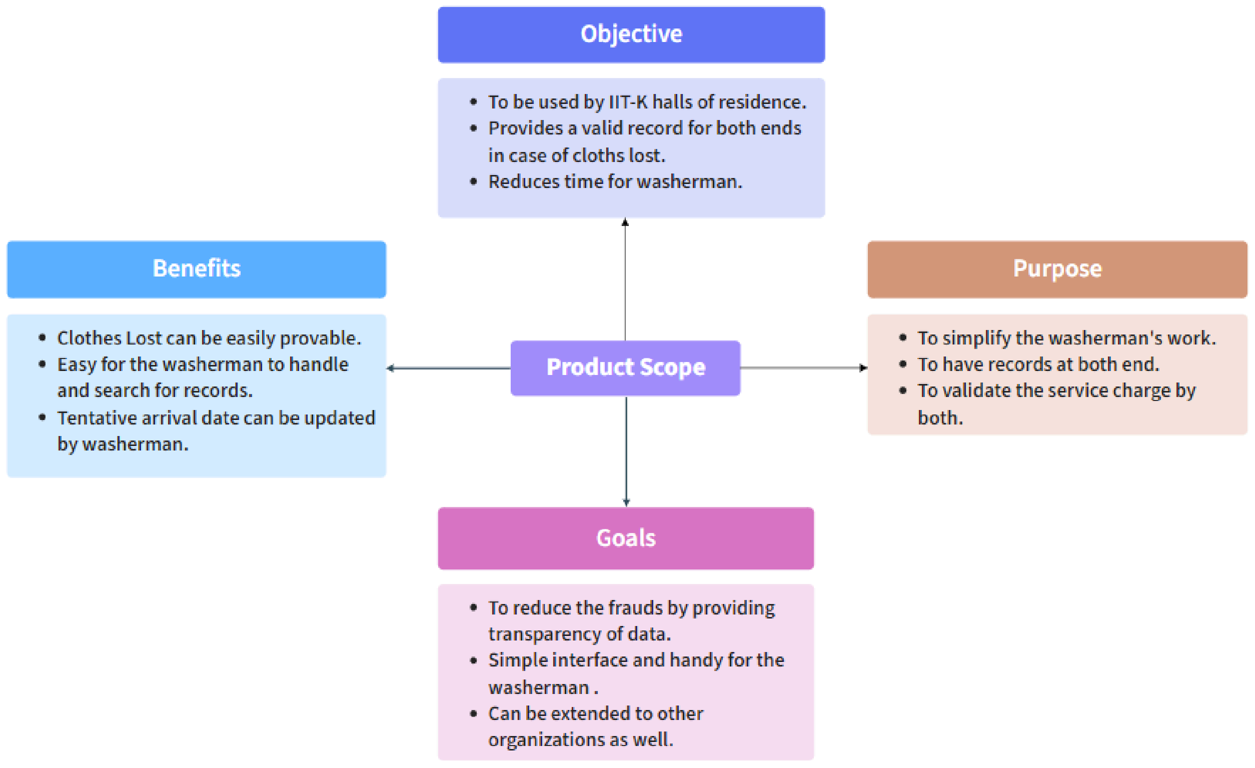
Version	Primary Author(s)	Description of Version	Date Completed
1.1	**Group 5 - DevDynamos**	-A basic version of the document was drafted. -Introduction and overall description was completed.	20/01/24
1.2	**Group 5 - DevDynamos**	-Specific requirements and other non-functional requirements were formalised.	24/01/24
1.3	**Group 5 - DevDynamos**	-Data dictionary and appendix log was added in the document.	25/01/24
1.4	**Group 5 - DevDynamos**	-Final document was formalised after few changes in format of the document.	26/01/24
2.1	**Group 5 - DevDynamos**	After completion for the final web application, we reviewed and made necessary changes. These new changes have been highlighted in yellow or ~~crossed out~~.	22/04/24

1. Introduction

1.1 Product Scope

The product, DBMS, aims to replace the antiquated, manual, and unreliable record keeping by the washermen across IITK hostels using a simple and easy-to-use interface. The application simplifies the life of the students as well as the washermen. It addresses the hassles related to clothes count, payment history, etc. The product's main Benefits are:

- Maintain a user-friendly interface for the students and the washerman.
- Give a better user interface for the washerman, who has to record clothes given by each room in a wing (a total of 10 rooms with three
- /two students each), and replace his old notebook.
- Ensure that the interface is responsive for mobile devices to facilitate easy access.
- Include a help or support section with FAQs and contact information for assistance.


Objective
• To be used by IIT-K halls of residence.
• Provides a valid record for both ends in case of cloths lost.
• Reduces time for washerman.
Benefits
• Clothes Lost can be easily provable.
• Easy for the washerman to handle and search for records.
• Tentative arrival date can be updated by washerman.
Product Scope
Purpose
• To simplify the washerman's work.
• To have records at both end.
• To validate the service charge by both.
Goals
• To reduce the frauds by providing transparency of data.
• Simple interface and handy for the washerman .
• Can be extended to other organizations as well.

1.2 Intended Audience and Document Overview

This Software Requirements Specification (SRS) is intended for a diverse audience involved in various aspects of the software development lifecycle. The primary readers include:

- **Users: Students and washerman**
- **Documentation Writers**
- **Developers**

Sections and Organisation

The SRS starts with sections that describe its importance and how to use it, then it moves forward with the overview and functionality of the product. Then, it covers a guide towards the interfaces, which makes it easier for a user to understand how they need to interact with the product, covering some use cases to wind up.

Sequence for Important Sections

- A **Developer** or a **Tester** can start reading the document with a focus on **overview** (2.1)→functionality (2.2)→design and implementation (2.3)→ interfaces (3.1) → functional requirements (3.2).
- The **users(students** and **washermen)** should focus on the **scope (1.1**)→ **overview (2.1)** and **functionality (2.2)**→ **specific requirements**, going through the **sections 3.1.1, 3.1.3** and **3.2**.
- The **washerman** and the **students** can find their interfaces in **section 3.1.**

1.3 Definitions, Acronyms and Abbreviations

DBMS	Dhobi Management System
UPI	Unified payment interface.
DB	Database
UI	User Interface
SRS	Software requirements specification
FAQs	Frequently asked Questions
Wing	Set of all students living on the same floor of a block.
ACID	Atomicity, Consistency, Isolation, Durability
API	Application programming interface

1.4 Document Conventions

<u>**Formatting Conventions:**</u>

● This document is written with an Arial font of **size 12** with single spacing and **1-inch** margins.
● Words highlighted with bold in the same font space represent terms whose explanations are either given in footnotes or separately in the same section.
● Italics highlight has been used for comments.
● Underline has been used for headings in subsections.
● Bullet point ordering has been used as a listing typesetting tool.

<u>**Naming Conventions:**</u>

- **Users**: washermen and students

1.5 References and Acknowledgments

The following websites were referred to in the process of making this document:
● Use-Case Model, washerman usage of the system - **Moqups**
● User interface - **Figma**

We'd also like to **acknowledge** the help of our **TA**, **Mr. Bharat Singh Kayarat**, for their valuable input in the creation of this document. We also would like to thank Prof. Indranil Saha for providing the SRS template and teaching the concepts.

2. Overall Description

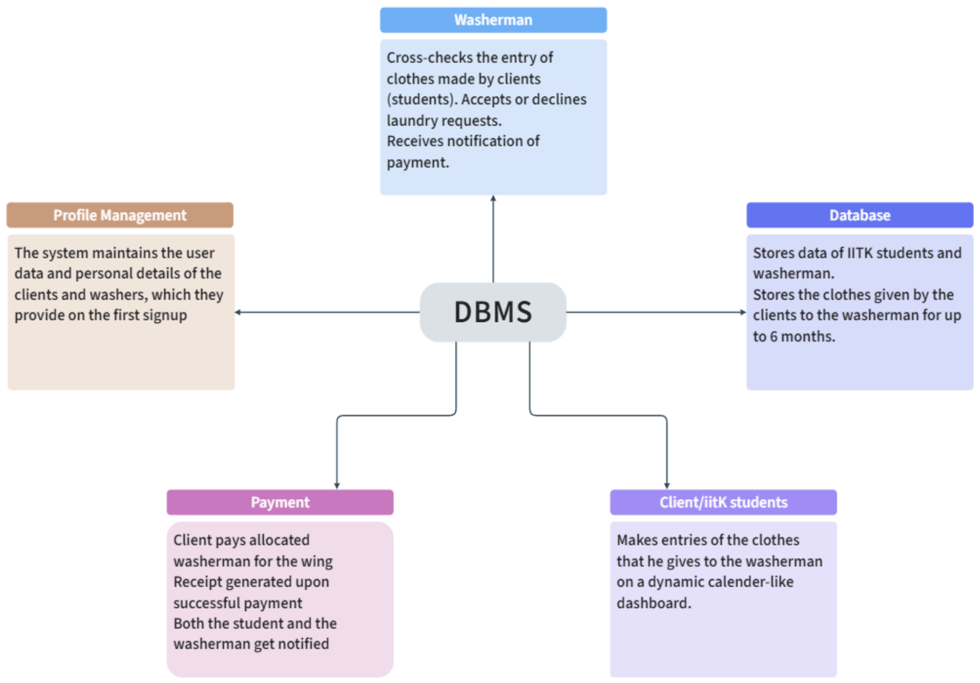
2.1 Product Overview

Our project "**Laundry Management system**", addresses the common problem across the hostels of IITK .

- The aim is to digitalize the manual, unreliable, and tedious record keeping by the Dhobi.
- Often, discrepancies arise between the dhobi and students, such as number of clothes given, amount left to be paid, etc.

Our project provides a solution to such problems by use of an easy-to-use website.

Context Diagram


Washerman
Cross-checks the entry of clothes made by clients (students). Accepts or declines laundry requests.
Receives notification of payment.
Profile Management
The system maintains the user data and personal details of the clients and washers, which they provide on the first signup
DBMS
Database
Stores data of IITK students and washerman.
Stores the clothes given by the clients to the washerman for up to 6 months.
Payment
Client pays allocated washerman for the wing
Receipt generated upon successful payment
Both the student and the washerman get notified
Client/iitK students
Makes entries of the clothes that he gives to the washerman on a dynamic calender-like dashboard.

2.2 Product Functionality

The product would allow every user to classify as a student or washerman
The product shall have the following functionality for the student :

- Account creation and profile details input
- User login via email/phone number
- Home page with a calendar showing total laundry submitted
- Color-coded calendar for easy status checking
- ~~Price rate chart for each clothing item~~
- Payment options: cash or UPI

The product shall have the following functionality for washermen

- User-friendly login page
- Monthly summary file displaying dues
- Accept or deny student laundry requests
- Notification and receipt upon receiving payment
- Translation feature for increased convenience

2.3 Design and Implementation Constraints

- The system must be designed to scale effectively to accommodate an increasing number of users and laundry transactions as the college grows.
- Email services are crucial for user logins,necessitating their reliable functioning.
- Due to storage constraints, we can only store data for one year at most.
- Automated Maintenance.

2.4 Assumptions and Dependencies

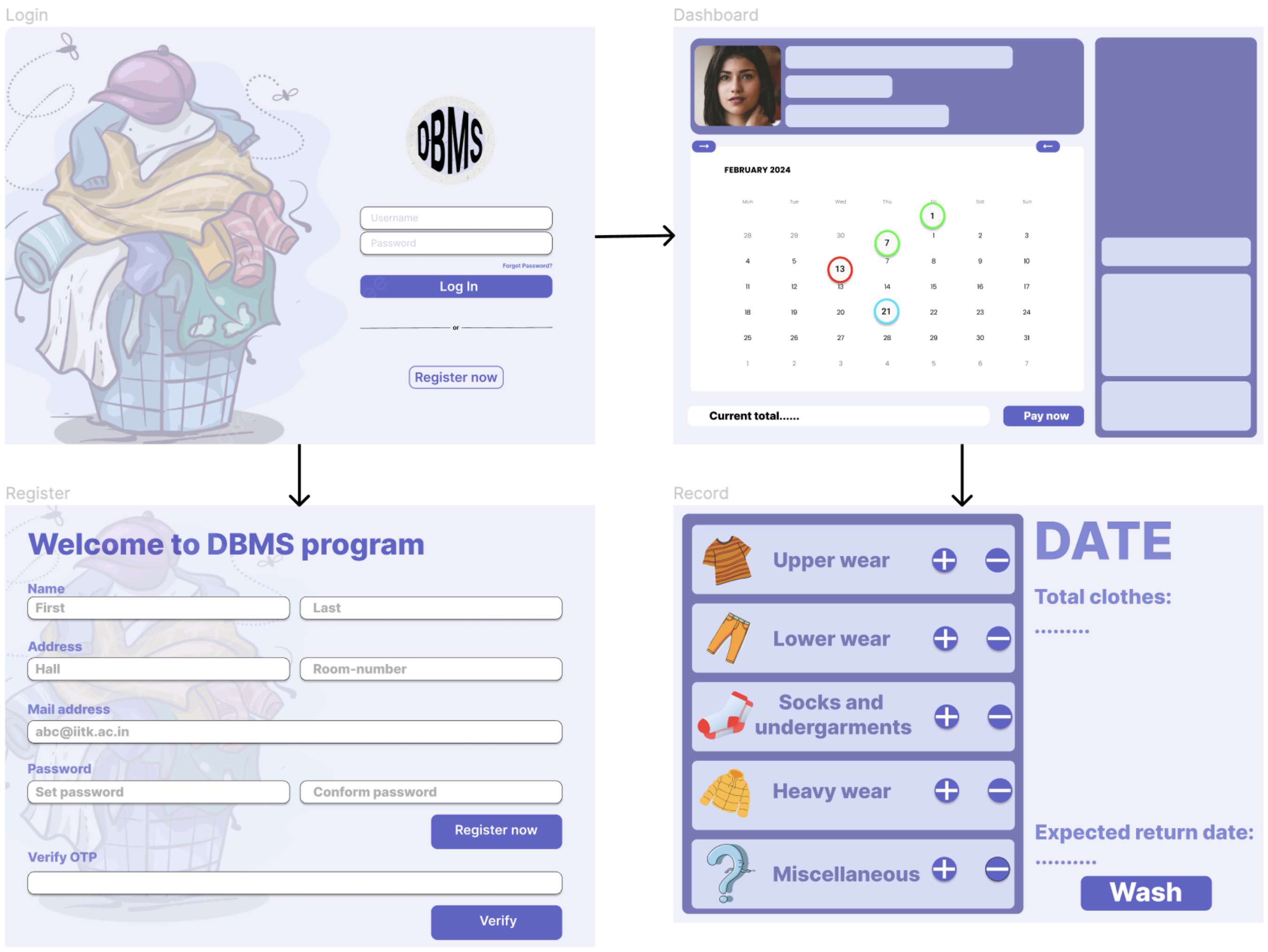
- Washerman possesses a smartphone, with good internet connectivity.
- Optimal performance is ensured on modern web browsers; older versions may result in inefficient interface usage.
- Dependency on third-party payment gateways.
- Dependency on third-party Translators to translate the contents in Hindi for washerman.
- An Admin exists whose role is to register new washermen and add new hall and wing details. (In context of Campus, Admin could be someone from hall administration or a student representative)

3. Specific Requirements

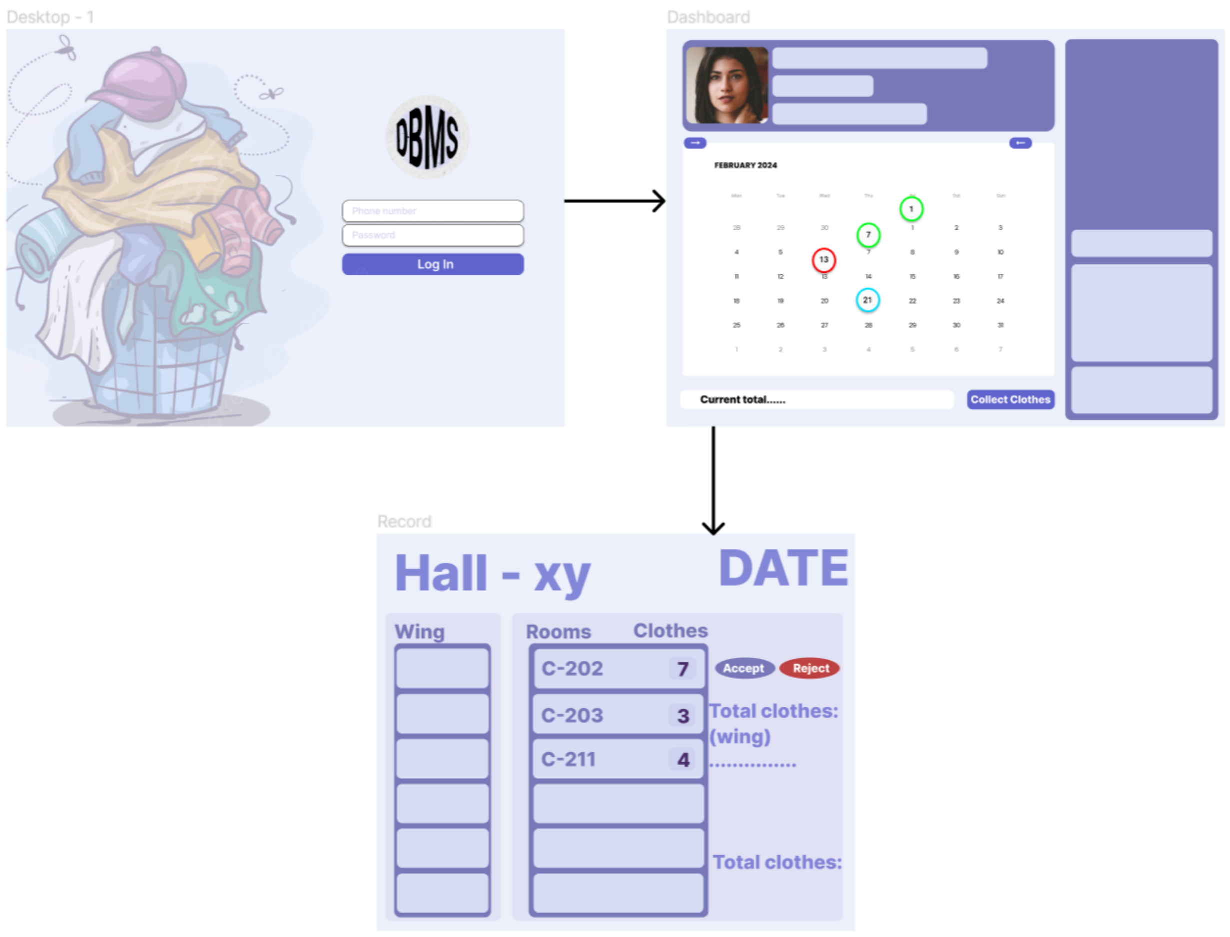
3.1 External Interface Requirements

3.1.1 User Interfaces(user end)

These images are just for reference, actual interfaces vary .



3.1.2 User Interfaces(Washerman end)



3.1.2 Hardware Interfaces

To access the website, clients and washermen need the necessary devices and an internet connection.
On the server side, our website will be hosted on a web server.

3.1.3 Software Interfaces

On the client side, users require an operating system (such as Windows, macOS, Linux, etc.) and a web browser. On the server side, the necessary software interfaces include a web server and a database.

3.2 Functional Requirements

3.2.1 Account creation and profile details input for students:

- The student shall create an account using his email, wing, and room number.
- The student will create a password for the website.

3.2.2 Student login via email/phone number:

- There will be a login page for the student to log in using his IIT-K mail and password.
- There will be a provision for a Forget password, which students can use to verify their identity and recreate the password.

3.2.3 Student Home page with a calendar showing total laundry submitted:

- A Dynamic Calendar for the current month will be displayed on the user dashboard.
- On each date, there will be the total number of clothes given.
- There will be an option to view past month's records.

3.2.4 Color-coded calendar for easy status checking:

- The dates in the calendars will be presented in different colors for students' convenience.
- The days on which the washerman did not come would not be colored.

- The days with clothes returned by the washerman will be green in color.
- The days on which the given clothes had not been returned will be displayed with a blue background.
- The days on which the washerman came and no clothes were given would be displayed in red color.

~~3.2.5 Price rate chart for each clothing item:~~

- ~~The student will have the option to view the rates of washing different pieces of clothing as given by the washerman.~~

3.2.6 Payment options: cash or UPI:

- Student's pending dues will be displayed on their dashboard.
- The student shall be allowed to pay the dues using UPI or cash.
- Upon selecting UPI, they can pay directly to the washerman using a third-party payment gateway via UPI.
- Upon selecting cash, the user shall pay cash physically to the washerman, which the washerman shall verify.
- A receipt will be generated by the site specifying the payment made.

3.2.7 User-friendly login page for Washerman:

- The washerman will have a simple login page on which they can log in using their phone numbers.

3.2.8 Monthly summary file displaying dues:

- The washerman will receive a summary at the end of each month in .pdf format.
- The summary shall contain the total number of each item of clothing he washed.
- The summary will also contain which students have pending dues for the month or the previous month.

3.2.9 Accept or deny student laundry requests:

- When the student sends a request, the washerman will cross-verify the number of clothes and either accept or deny the request.

- If he accepts the request, the entry will be added.
- If he rejects the request, the entry will be deleted, and the student can add a new entry.

3.2.10 Notification upon receiving payment:

- The washerman will receive a notification on his registered mobile number regarding a new payment made to his UPI ID.

3.2.11 Translation feature for increased convenience:

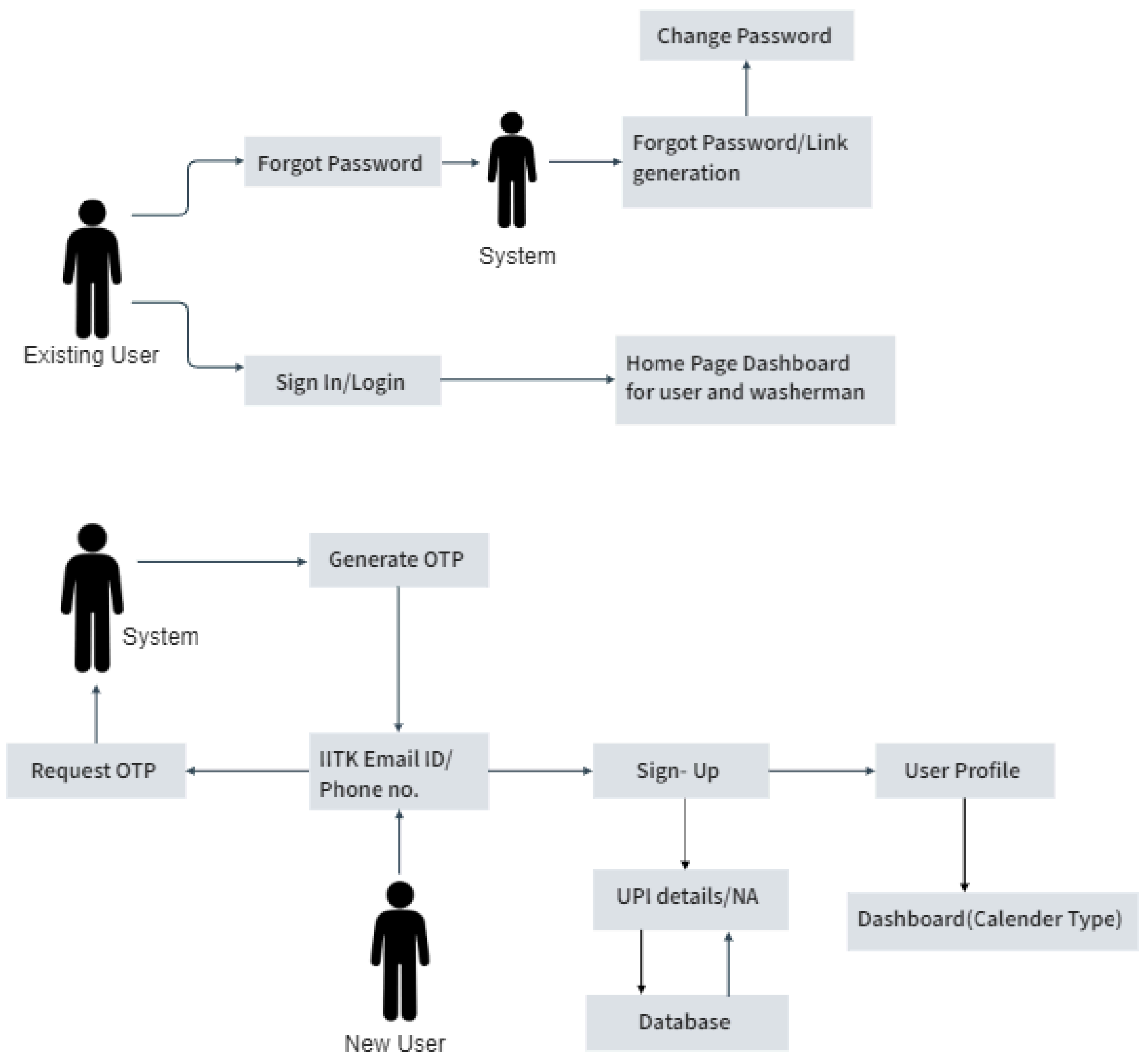
- Given that the washerman might not be comfortable with English, our website will make it convenient for him to use it by translating it to Hindi, by using a third-party translator.

3.2.12 Students will receive a copy of their payment receipts on their mail IDs.

- On payment via UPI, students will receive a receipt for further reference**.**

3.3 Use Case Model

3.3.1 Use Case #1 (Authentication)



Author – **Kushagra, Vishal, Aditya**

Purpose - To authenticate valid users(IITK students and washerman)

Requirements Traceability- Signup interface, Login interface, Reset password interface, DB.

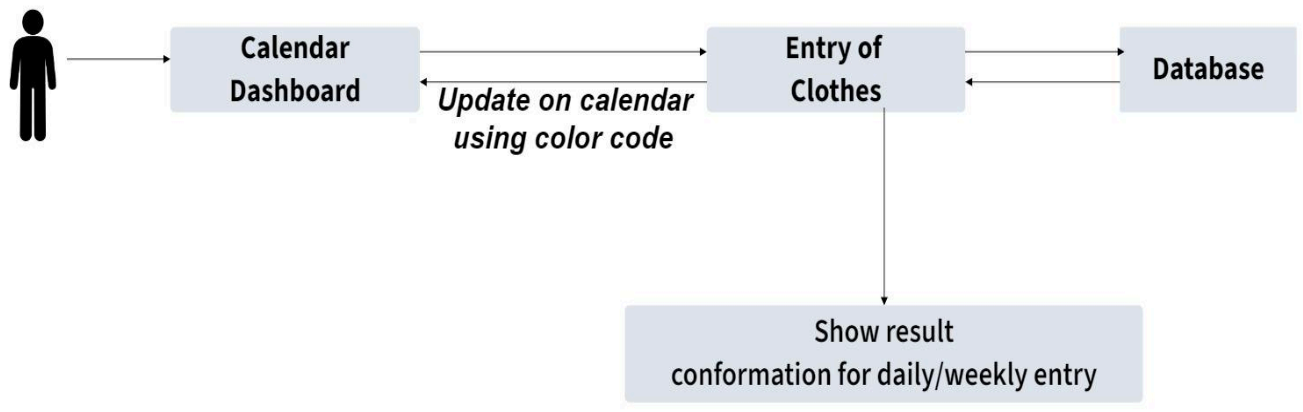
Priority - High

Preconditions - Clients must have IIT email id(for students only)

Post conditions - The user is logged in and is able to interact with the system

Actors – Human, System

3.3.2 Use Case #2(IITK Student-end)


Calendar Dashboard
Entry of Clothes
Database
Update on calendar using color code
Show result conformation for daily/weekly entry

Author – Kushagra, Vishal, Udbhav

Purpose - To enable the user to make entry of clothes,dashboard for the user to cross check the total dues and make payment online/cash.

Requirements Traceability- Database,Calendar type dashboard interface.

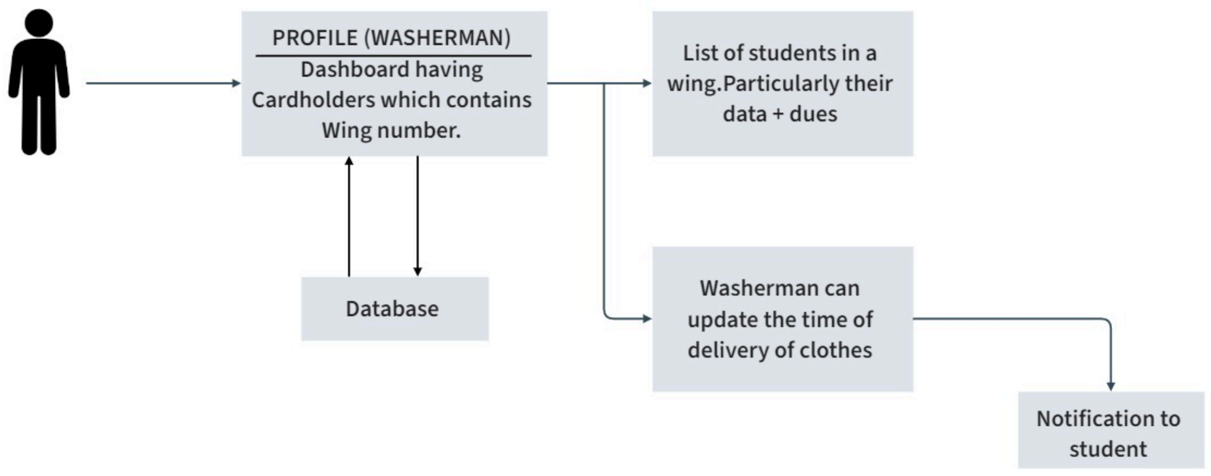
Priority - Medium

Preconditions -Authenticated user must login the website , the user must have an IITK email id.

Post conditions -The calendar dashboard should reflect the updated status of the laundry service, indicating completion or delivery dates.

Actors – Human,System

3.3.3 Use Case #3(Washerman-end)


PROFILE (WASHERMAN)
Dashboard having Cardholders which contains Wing number.
List of students in a wing.Particularly their data + dues
Database
Washerman can update the time of delivery of clothes
Notification to student

Author – Rishikesh, Shriya, Nandini

Purpose -. Washerman can see the number of laundry given by students. He can also check their remaining dues and update the delivery time of clothes.

Requirements Traceability- Database

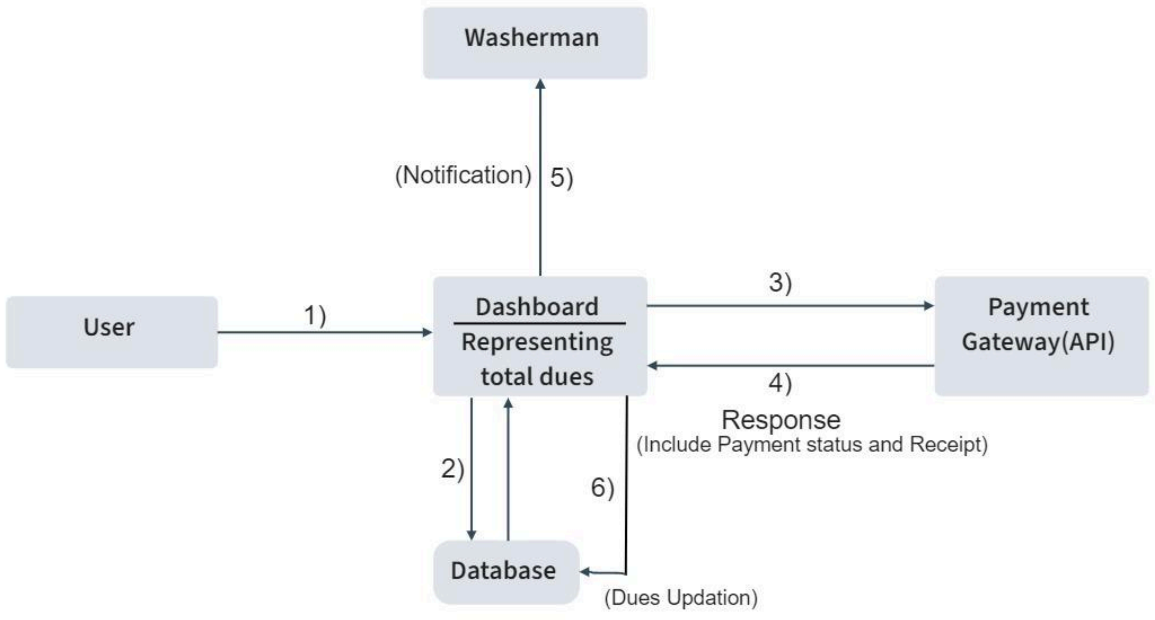
Priority - Medium

Preconditions -.Washerman must have logged in the website and be able to interact with the system.

Post conditions - Washerman should be able to notify the students regarding delivery dates and their dues if and when needed.

Actors – Human,System

3.3.4 Use Case #4(Payment)



◉ Server is the implicit middleware in all the above communication channels

Author – Yash, Abhishek, Anshu

Purpose -. The primary purpose is to facilitate secure online transactions between clients and washers. By integrating a reliable payment system, the website ensures that financial information is handled securely, instilling user confidence.

Requirements Traceability- Ensures that only authenticated users (clients) can access and utilize the payment feature.

Priority - High

Preconditions -A confirmed service request or an assigned washerman should exist before initiating the payment process. The client must have valid and up-to-date payment information (credit/debit card, digital wallet, etc.) stored in their account.

Post conditions -After a successful payment transaction, the client and the washerman should receive confirmation of the payment. The washerman receives a notification while the client gets the printed receipt.

Actors – User, Washerman,System

4.Other Non-functional Requirements

4.1 Performance Requirements

Performance of the website depends on several factors like:

1) **Concurrency Management**: The database read and write operations must adhere to ACID properties, given the simultaneous access by multiple users. Ensuring there are no concurrency issues is imperative for a seamless user experience.

2) **Latency Optimization**: The preference is for API requests to be fulfilled in less than 100ms, contributing to the smooth functioning of the application. This aligns closely with the industry standard for expected latency.

4.2 Safety and Security Requirements

- **Safety Requirements**

a) **Mandatory login**: All students must create an account with an IITK email id to ensure the authenticity of them.

b) **Privacy** ; All the data given by users (bank account /upi details etc .) is of utmost importance and should not be disclosed in any form .

- **Security Requirements**

a) **Confidentiality** : Only the students and corresponding washerman should be able to get the particular laundry and payment details.

b) **Integrity:** The software should not corrupt any sort of transactions/data between the students and the washerman.

c) **Encryption:**
All sensitive information such as passwords should be stored in an encrypted format using SOTA encryption schemes.

4.3 Software Quality Attributes

4.3.1 Availability:
The website will be accessible to all users within the campus, allowing every washerman or student in IITK to operate it with an account and without any restrictions.

4.3.2 Consumability:

An intuitive user interface will enhance the website's usability, especially for first-time users. It will be easy for students to add laundry and hassle free payment.The washerman will be able to send notifications in any unforeseen circumstances .

4.3.3 Correctness:

The data entered by the students will be easily verified by the washerman by denying or accepting the given data.

4.3.4 Interoperability:

The operation of this application will not interfere with any other application operating simultaneously on the device.While adding clothes, users can simply add the clothes and remove them if needed, operate any other application simultaneously, and resume adding/removing from the Recent screen/browser tabs. This will ensure interoperability of the application.

4.3.5 Scalability:

The software should easily accommodate any future washerman or students within the IITK campus. Its services should be designed for both horizontal and vertical scalability, allowing for efficient expansion as needed.

5. Other Requirements

Legal data requirements

- The user data obtained through the website must be stored within the regional data center of the country. This is in adherence to data protection and privacy regulations that may be mandated by the national government.

Appendix A – Data Dictionary

- **Washerman class:**

Element name	Washerman
Description	Each washerman is registered with his allotted wings and halls.
Attributes	1. Name - String 2. Phone number - String (used as login id) 3. Hashed Password (String) 4. Halls- (String, Wing Object ref array) -> Wing Obj. - (String, student ref array) 5. Upcoming Date (Date) 6. Account Id (String) - for payment
Operations	1. **register()** : registration of a user using the software for the first time. 2. **login()** : called while the user wants to login.
Relationships	Each washerman has a single account corresponding to their phone number. The washerman has a many-one relationship with the student table, corresponding to the wings allotted to him.

- **Student Class**

Element Name	Student
Description	Each student is registered with their email onto the platform, and logs in every time using their password.
Attributes	1. Name - (String) 2. Roll - (String) 3. Email - (String) 4. Hall - (String) 5. Wing - (String)

	6. Total Dues - (Int) 7. Hashed Password - (String) 8. Records - (Array of [Date, Cloth Obj]) 9. Last Cleared - (Date) 10. Receipt - (Array of Receipt Obj)
Operations	1. **register()** : registration of a user using t2he software for the first time. 2. **login()** : called while the user wants to login. 3. **Forget_pwd()**: Users can use it to generate new passwords in case they forget. 4. **Pay_Dues()**: Users can pay their current dues using this.
Relationships	Each user has a single account corresponding to their Email id. Each student is related to one washerman which has been allotted to their wing. Each student has a many-one relation with the clothes table.

- **Student's Clothes Class**

Element Name	Clothes
Description	Stores the clothes given by a student on each day.
Attributes	1. St_Email: string 2. Number:int 3. Date: date 4. Type: string 5. Status: string
Operations	1. **Add():** Can be used to add new entries in the table. 2. **Delete():** Removes entries from the table. 3. **Change_status()**: Will be used when the washerman returns the

	clothes.
Relationships	Each student is related to the clothes that he gives to the washerman each time he comes. Each washerman is related to the clothes from his wing's customers.

Appendix B - Group Log

SL no.	Date	Timings	Venue	Description
1	6 January 2024	2 pm -4pm	CCD	Brain Stormed various possible ideas for the project.
2	10 January 2024	9pm -11pm	RM building	Finalized the idea for the project and discussed the various aspects of it.
3	11 January 2024	4pm-5pm	RM building	We had a meeting with the professor to share our ideas and take valuable suggestions.
4	12 January 2024	9pm -11:30am	Google Meet	Started working on key aspects of the project and everyone gave their ideas.
5	16 January 2024	8pm-9:30 pm	Student Lounge	We created the preliminary UI design for our product to provide a clear overview of what needs to be developed and the features that are part of the project.
6	17 January 2024	9:30pm-10 pm	Google Meet	Met the TA for the first time and took his valuable tips.
7	20 January 2024	12pm-3pm	RM building	Worked on a draft of Software Requirement Specification document.
8	24 January 2024	9pm-11pm	RM building	Met to combine the work of different persons and cleared each other's doubts.
9	25 January 2024	9pm-10pm	Google Meet	Updated TA with our current progress and discussed the issues we were facing.
10	26 January 2024	5pm-7pm	RM building	Final meet before SRS submission and fine tuned our document.