Design Document

Ahmedabad
University

CSE300 Software Engineering

Submitted to Faculty: Prof. Khushru Doctor

Name	Enrollment Number	Program
Digant Patel	AU2040086	B Tech (CSE)
Madhvendrasinh Jhala	AU2040162	B Tech (CSE)
Yug Patel	AU2040181	B Tech (CSE)
Deep Patel	AU2040250	B Tech (CSE)
Parth Zinzuwadia	AU2140233	B Tech (CSE)

Table of Contents

1 Introduction

1.1 Purpose of this document

The purpose of this Software Design Document is to remain focussed on the goals of the project and to provide a clear vision of what needs to be done and to provide a descriptive enough documentation of the design of the system to allow for software development to proceed with an understanding of what is to be built and how it is expected to be built.

1.2 Identification

1.2.1 Database Design

For the database design, MySQL has been chosen as the preferred database management system. MySQL is a relational database management system (RDBMS) that provides a robust and structured solution for handling diverse and large datasets. Its relational model allows for the definition and manipulation of data in tabular structures, ensuring data integrity and consistency.

The choice of MySQL is particularly advantageous for our project, as it aligns well with the structured and well-defined nature of the data requirements. The relational approach facilitates the establishment of relationships between different tables, enabling efficient data retrieval and management.

Additionally, MySQL is not limited to data storage; it also serves as a powerful tool for creating and executing queries. MySQL's querying capabilities enable the formulation of complex queries for efficient retrieval and manipulation of data, contributing to the overall performance and functionality of the system. The use of SQL (Structured Query Language) in MySQL further enhances the ability to interact with and manage the database seamlessly.

1.2.2 Frontend Prototyping

For frontend prototyping, Next.js has been employed. Next.js is a JavaScript library maintained by Vercel, designed for building user interfaces with a focus on component-based development. Its declarative and efficient nature allows for the creation of interactive and dynamic user interfaces. Next.js facilitates a smooth and responsive user experience, making it an ideal choice for prototyping and developing the client-side of our application.

The component-based architecture of Next.js promotes modularity and reusability, streamlining the development process and ensuring a consistent and maintainable codebase.

1.2.3 Backend Development

The backend of the system is developed using Node.js and Apollo GraphQL. Node.js is a server-side JavaScript runtime known for its event-driven, non-blocking I/O model. This makes it well-suited for building scalable and high-performance web applications. Node.js, coupled with its extensive package ecosystem (npm), allows for rapid development and deployment of server-side functionalities.

The choice of Node.js for the backend ensures a unified language (JavaScript) throughout the entire application stack, fostering easier communication and collaboration between frontend and backend developers. Additionally, Node.js's asynchronous nature enhances the system's responsiveness, making it suitable for handling concurrent operations and real-time functionalities.

In summary, the technology stack selected for this project leverages the strengths of MySQL, Next.js, Apollo GraphQL and Node.js to create a robust, scalable, and responsive software solution. This strategic combination aims to fulfill the project's requirements efficiently and provide a foundation for seamless future development and maintenance.

1.3 Intended Audience

1. **End Users (Developers)**

   Purpose: Utilize the platform to seek answers to technology-specific and general tech questions, connecting with like-minded individuals globally.

2. **Project Manager**

   Plan milestones and delivery dates, ensuring the development team adheres to the project timeline.

3. **Designer**

   Base system design on the document, continuously referencing it to ensure alignment with customer needs.

4. **Developer**

   Use the document as a foundation for system development, linking requirements to software development and ensuring customer needs are met.

5. **Tester**

   Derive test plans and test cases from documented requirements, conducting tests at various stages of development to ensure compliance with the document.

1.4 Languages and Tools

Language/Frameworks

- Node.js
- Next.js
- Apollo GraphQL

Tools

- Apollo Sandbox
- Visual Studio Code

2. General Overview And Design Consideration

2.1 General Overview

In response to the evolving needs of Ahmedabad University's hostel facilities, the Hostel Management Software is conceived to streamline and enhance the management of various amenities provided to students. Acknowledging the dynamic nature of student living, the software aims to facilitate a seamless experience for both students and hostel administrators.

2.2 Current Challenges:

The administration of hostel facilities often encounters challenges related to room assignments, maintenance requests, dining services, and overall facility management. The manual processes in place may lead to inefficiencies, miscommunication, and difficulties in keeping accurate records.

2.2.1 Proposed Solution - Statement of Need

The Hostel Management Software is designed to address these challenges by providing a comprehensive platform that automates and centralizes key hostel management processes. It aims to create a more organized and efficient environment for both administrators and students, fostering a positive living experience within the university hostel.

3. Design Consideration

3.1 Goals and Guidelines

3.1.1 Architecture

The solution should be such that all the requirements stated in the requirements document both functional as well as non-functional, should be accomplished. Also, the software should be coded in such a way that it could incorporate required future changes.

3.1.2 Ease of Use

The features of the application should follow the concepts of human-computer interaction, and different categories of users should find it easier to use.

3.1.3 Development Environment

The application development is supposed to stay consistent. Adapting to technological advances, this project aims to build the application in the latest versions of technology with backward compatibility for better performance.

3.1.4 Extensibility

The application is aimed to be extensible to adding new features, also providing backward compatibility.

3.2 Development Methods & Contingencies

The fundamentals of a good architecture involve layering the application into numerous autonomous means that components are more cohesive and loosely coupled and it can be replaced separately and keep the application running while we work on a particular layer.

3.2.1 Scalability

Ensure that the architecture can be horizontally scaled across numerous servers and locations. The solution should allow you to add and remove new servers as needed as your traffic increases.

3.2.2 Availability

The architecture should support a high-availability environment. Redundant infrastructure is necessary. As a result, even if some servers or an entire data center fail, the solution will still be available.

3.2.3 Extensibility

The architecture must be flexible enough to replace modules, add layers, and expand the application without worrying about the underlying data contracts.

3.2.4 Security

The solution architecture should reveal as little code as feasible. The majority of the supporting components ought to be hidden. Additionally, any system's security should have multiple layers of protection.

3.2.5 Functionality

When used in accordance with predetermined guidelines, the software is capable of delivering features that satisfy both explicit and implicit needs.

3.2.6 Separation of responsibility

The system should be sufficiently modular such that each piece of code only has the necessary responsibilities. Both the front end and the back end should provide business logic in the front end.

3.3 Development Environment

Software	Description
Apollo GraphQL	Backend server
Node.js	Programming Language at server side
Next.js	Programming Framework for Web
MySQL	Database

4. System Architecture And Architecture Design

4.1 System Descriptions

	Framework/Tools	Description
Backend Server	Apollo GraphQL/Node.js	Apollo is a GraphQL platform that simplifies data management in client applications. It provides tools for building GraphQL APIs, connecting them to front-end applications.
Database	MySQL	MySQL is a source-available, general-purpose, relational database management system (RDBMS) designed to meet the needs of modern application developers and the cloud era. It is classified as a relational database program and employs a structured approach to data management.
Frontend Framework	Next.js	Next.js is a react framework used for building reusable UI components. Next.js can also render on the server using Node, and it can power native apps using React Native.

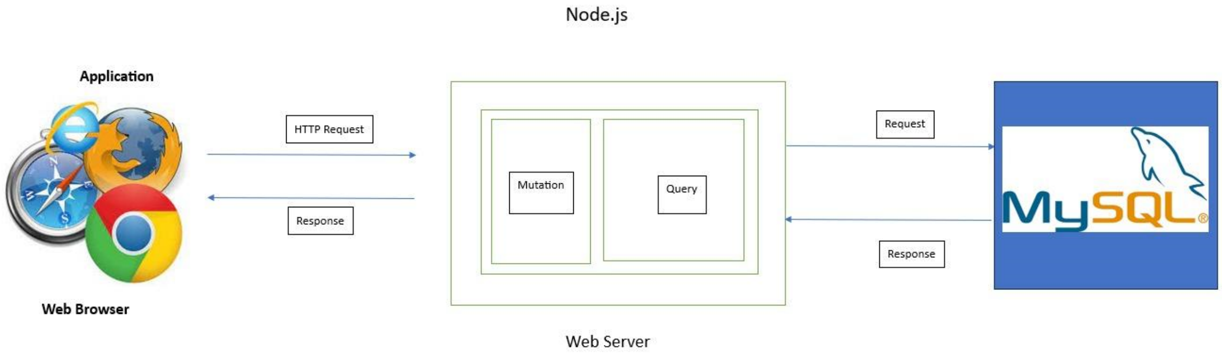
4.2 Software Architecture

4.2.1 Software Element

Function	Description
General Access	Users can access the hostel management system through a web interface, accessible from any browser.
Student Registration	Admin can register new students, capturing and storing personal and contact information.
Room Assignment	Admin can assign rooms to new students and maintain an updated record of room occupancy.
Room Cleaning Booking	Students can request room cleaning services through the system, and housekeepers can manage and fulfill these requests.
QR Code Generation	Generate QR codes for students, potentially for identification and access purposes.
Dining Services Management	Monitor student counts at the University Cafeteria and Residence during meal times, ensuring efficient dining services.
View Room Records	Admin can view records of cleaned rooms, pending requests, and complaints uploaded by students.
Mess Management	Mess manager can monitor the count of students having meals at the University Canteen and Hostel.

View Student Profiles	Admin can view details of individual student profiles, including room assignment.
Housekeeping Requests	Housekeepers can view and manage requests made by students for room cleaning and bedsheet changes.
See Details	Admin can see details on room occupancy, housekeeping requests, and dining service usage.

4.2.3 Topology Diagram



4.2.3 ER Diagram

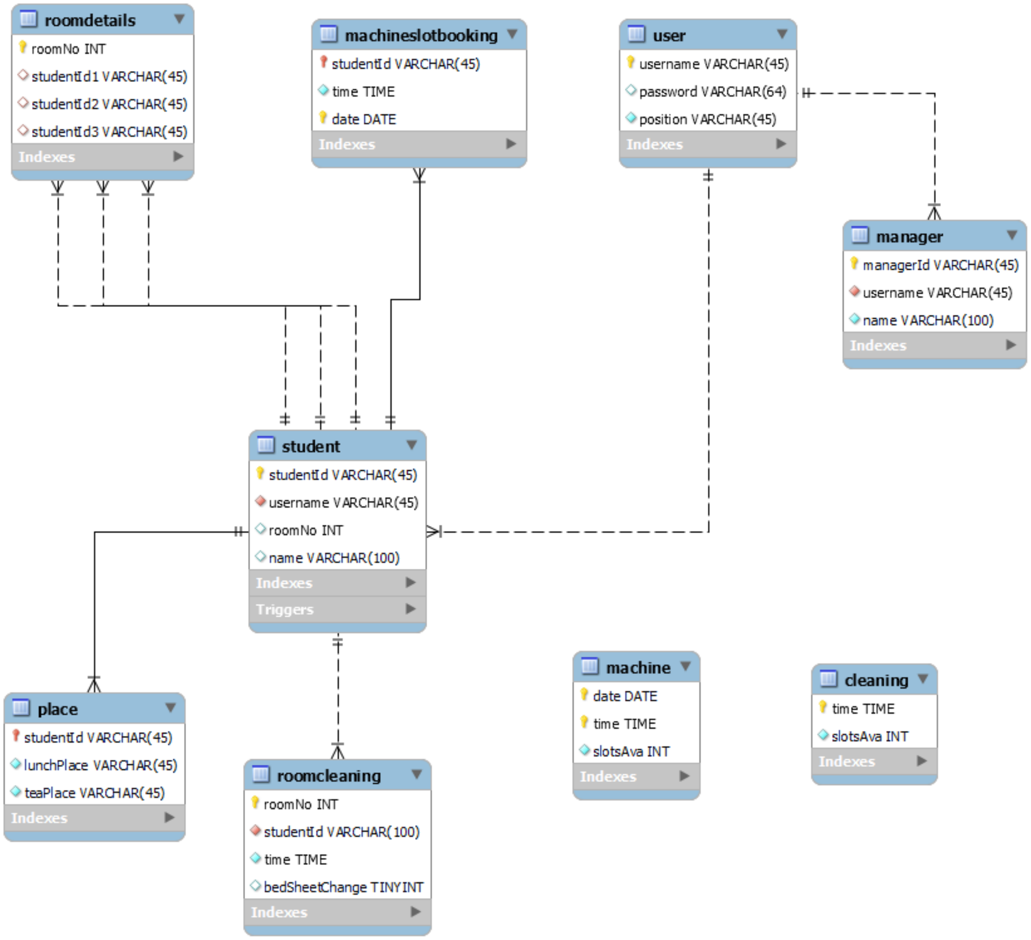
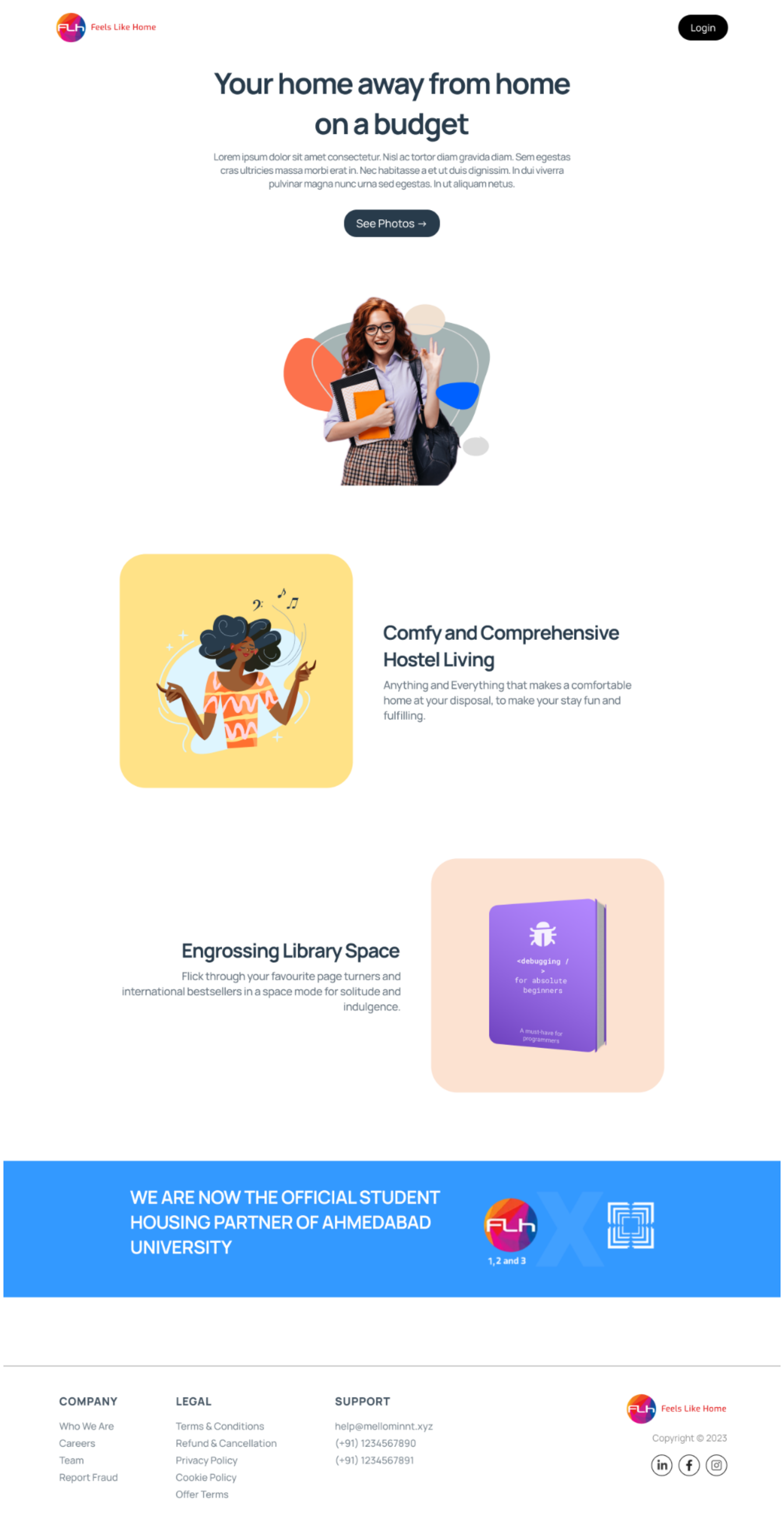


Figure1: Home-screen Design



Your home away from home on a budget

Lorem ipsum dolor sit amet consectetur. Nisl ac tortor diam gravida diam. Sem egestas cras ultricies massa morbi erat in. Nec habitasse a et ut duis dignissim. In dui viverra pulvinar magna nunc urna sed egestas. In ut aliquam netus.

Comfy and Comprehensive Hostel Living

Anything and Everything that makes a comfortable home at your disposal, to make your stay fun and fulfilling.

Engrossing Library Space

Flick through your favourite page turners and international bestsellers in a space mode for solitude and indulgence.

<debugging / > for absolute beginners
A must have for programmers

WE ARE NOW THE OFFICIAL STUDENT HOUSING PARTNER OF AHMEDABAD UNIVERSITY

COMPANY
Who We Are
Careers
Team
Report Fraud

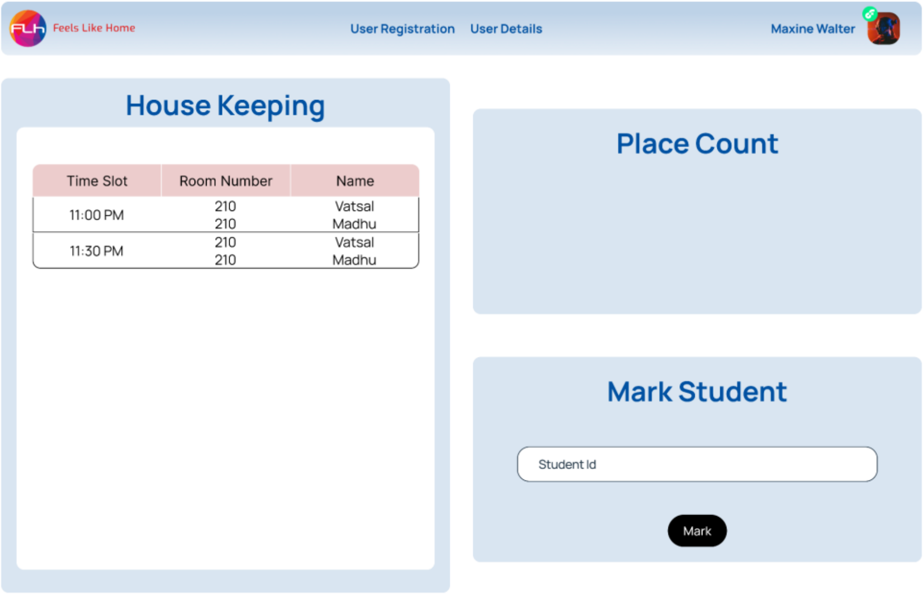
LEGAL
Terms & Conditions
Refund & Cancellation
Privacy Policy
Cookie Policy
Offer Terms

SUPPORT
help@mellominnt.xyz
(+91) 1234567890
(+91) 1234567891

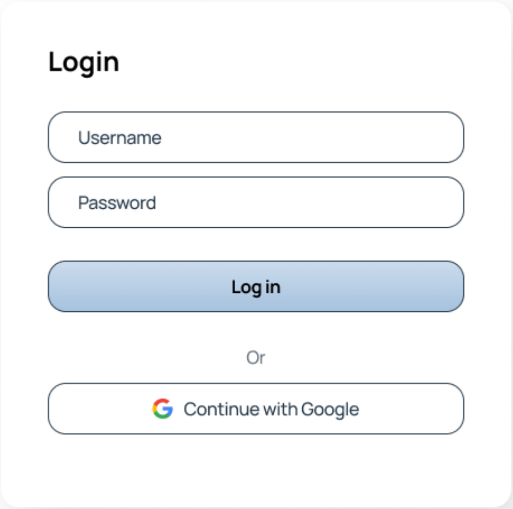
Copyright © 2023

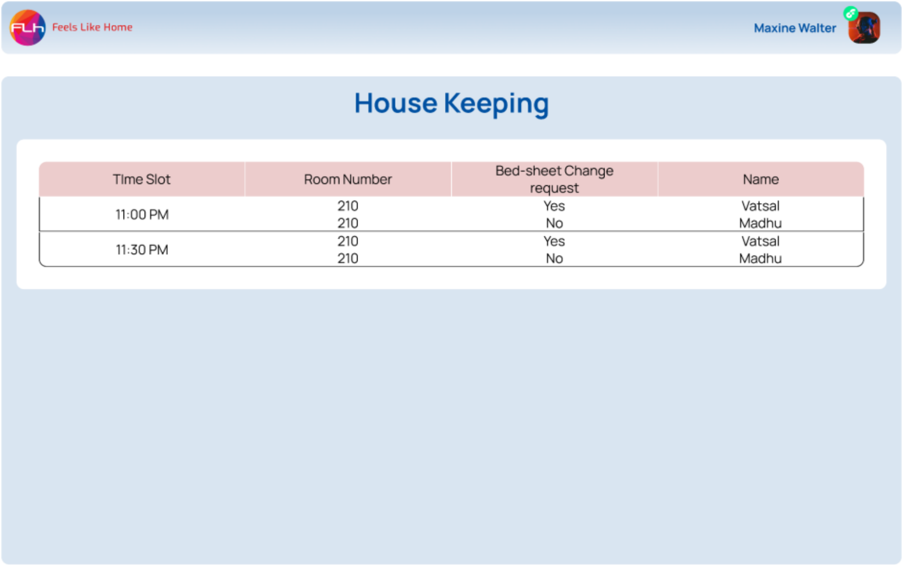
Figure 2 : Admin View Panel

Figure 3: Login Screen view



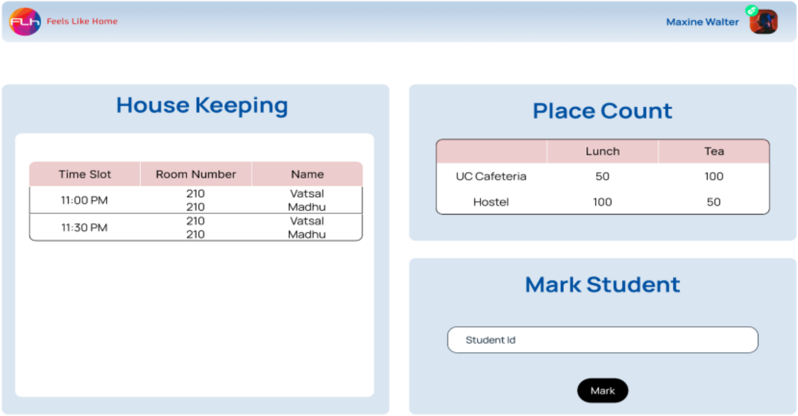
Time Slot	Room Number	Name
11:00 PM	210 210	Vatsal Madhu
11:30 PM	210 210	Vatsal Madhu





House Keeping

Time Slot	Room Number	Bed-sheet Change request	Name
11:00 PM	210 210	Yes No	Vatsal Madhu
11:30 PM	210 210	Yes No	Vatsal Madhu



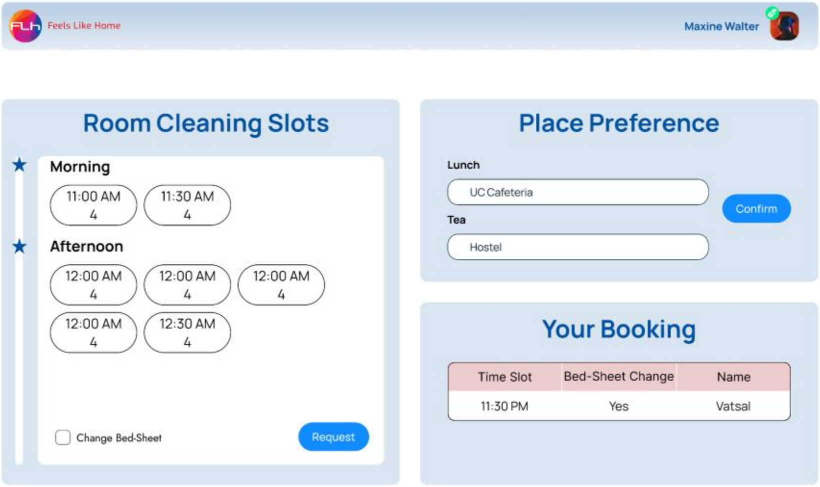
House Keeping

Time Slot	Room Number	Name
11:00 PM	210 210	Vatsal Madhu
11:30 PM	210 210	Vatsal Madhu

Place Count

	Lunch	Tea
UC Cafeteria	50	100
Hostel	100	50

Mark Student



Room Cleaning Slots

Morning

Afternoon

Place Preference

Lunch: UC Cafeteria

Tea: Hostel

Your Booking

Time Slot	Bed-Sheet Change	Name
11:30 PM	Yes	Vatsal

Figure 4: House Keeper’s Screen view.

Figure 6: Students screen for Housekeeping service booking.

Figure 5: Hostel Admin’s Screen view.