DormEase

A Comprehensive Digital Solution for SRH Heidelberg's BS13 Community

Master Thesis

by

Syed Nabeel Azeez

Matriculation number: 11017585

2023-10-23

SRH University Heidelberg

School of Information, Media, and Design

Applied Computer Science (ACS)

Internal Supervisors: Prof. Dr. Gerd Moeckel

Prof. Christoph Hahn

Affidavit

Herewith I declare:

- I have composed the chapters for the Master Thesis for which I am named as the author independently,
- I did not use any other sources or additives than the ones specified,
- I did not submit this work at any other examination procedure.

Heidelberg, 2023-10-23

(Syed Nabeel Azeez)

Acknowledgment

I take great pleasure in expressing my gratitude to many people for their invaluable guidance that helped me to successfully complete my master's research.

My sincere gratitude goes out to my mentor, Prof. Dr. Gerd Moeckel of SRH Hochschule Heidelberg. His constant effort in structuring the Applied computer science course to make it better and more practical oriented gives a master's student like me a solid foundation to make a mark in the field of software. His advice and ideas he shared during thesis seminars were invaluable in helping me through the process of writing my thesis. His support during every stage of my research enhanced my work and helped me focus on important elements like the thorough literature review, application development, and iterative feedback.

Prof. Christoph Hahn's mentorship not only elevated the project but also inspired me to think creatively and develop problem-solving skills. His insights and guidance were instrumental in enhancing my app's development. His constructive feedback and innovative ideas consistently improved its design and functionality.

I would like to express my deepest gratitude to my parents, for without them, I wouldn't be where I am today. Their unwavering support and belief in me have been my driving force. I also want to thank my sister for her invaluable support during challenging moments.

Finally, I owe a tremendous debt of gratitude to my grandfather. His guidance and teachings have been a guiding light in my journey, and I wouldn't have been able to accomplish anything without his wisdom. Though he's not here with me today, I know he's watching over me from above.

Abstract

The BS13 dormitory at SRH Hochschule Heidelberg is a student residential community that currently relies on outdated manual methods for everyday activities and operations. The master's thesis is aimed at delivering an IT solution by creating a web-based application called DormEase to replace these traditional methods of carrying out daily operations and activities.

Dorm Ease is a user-friendly web-based application that aims to bridge the digital gap with an efficient backend which is also graphical user interface (GUI) focused, dependable, and more efficient than current solutions. The application was developed according to the traditional three-tier architecture, with the interactive front end built with HTML, CSS, Bootstrap, and embedded JavaScript, and the backend built with Node.js and Express.js to create a RESTful API for data delivery and communication between the frontend and the server. MongoDB is used to store and manage data, and Mongoose is used to facilitate interaction with the MongoDB database.

The features presented in the prototype allow residents to perform their everyday operations with a few clicks rather than physically going to places to complete their activities. As a result, the students save time and effort. One of the features included also supports sustainability by decreasing and reusing products the residents no longer need. The application directly or indirectly contributes to the enhancement of their lifestyle.

It is about people and their needs, not simply technology. We understood the experiences of the BS13 community, listening to their concerns and challenges. With compassion as our guide, we created a prototype that focuses on tackling the difficulties of the residents of the BS13 Dormitory.

Table of Contents

1. Introduction

1.1 Overview

Dormitories, also known as hostels, are built in universities and colleges to give students a comfortable place to stay that is adjacent to academic facilities. John. M Braxton provides insights into the impact of higher education on students, including factors related to dormitory life. It offers contemporary evidence and findings that can be valuable for discussing how student dormitory experiences affect academic performance **(Braxton 2016).**

One such student dormitory is BS13 dormitory which houses international students from different parts of the world. These students pursue courses in different domains at SRH Hochschule Heidelberg. The master's thesis is dedicated towards providing an IT solution for the students residing in this dormitory which would directly impact their daily lifestyle and overall university experience in a positive manner. It has been noticed that the management and residents of the dormitory use conventional methods for daily crucial chores. The major goal is to assist the BS13 community by developing a web application to provide an IT solution which is more graphical user interface (GUI) focused, dependable and more efficient than the conventional techniques currently in use by the residents of the dormitory.

The Master thesis includes recognizing the old conventional methods in use which could be replaced by a digital solution. The first phase of this master's thesis involves identification of the problem statement which was recognized by conducting various surveys and interviews involving the residents of the BS13 dormitory. The main idea was to empathize and understand their pain points and areas where the application can bring value.

The different problem statements were noted down, next step involved feature extraction where a few problem statements were picked out of a pool of many different existing problem and using design thinking approach we tried to find

an efficient way to solve the problem. A user centric theory by Ames highlights user-centered design's importance and aligns with the thesis on enhancing BS13 dormitory residents' experiences. Emphasizing usability as a success criterion reinforces the goals of an efficient IT solution **(Ames 2001).**

The next step in the thesis involved kicking off the project where the implementation was carried out. Technical aspects of the thesis were decided including various things like picking the right technical stack by weighing pros and cons for developing the prototype. This step also involved choosing the software development methodology, Architecture, flow diagram, designing wire frames for front end, selecting frameworks for both front end and backend and databases. The interactive front end is developed using HTML, CSS, Bootstrap, and embedded JavaScript for the user interface and interactivity. The backend is developed using Node.js and Express.js to create a RESTful API for data delivery and communication between the frontend and the server. MongoDB is used as the database for storing and managing data, and Mongoose, as an ODM (Object Data Modeling) library, is employed to facilitate interaction with the MongoDB database." The development of the prototype was carried out and tested by users, the feedback was taken and the protype was refactored to make it a better experience for the user.

1.2 Problem Description and Research Questions

The BS13 student dormitory at SRH Hochschule Heidelberg faces numerous challenges due to the reliance on traditional, non-digital methods for daily tasks and management processes. These challenges result in inefficiencies, inconvenience, and a lack of transparency for the dormitory residents. During the surveys and interviews conducted it was noticed the dorm lacks digital solutions in a lot of areas. Below are problems identified in different areas which can be improved by providing a digital solution which would reduce time and effort.

There is no digital solution for a BS13 resident to carry out daily operation such as:

- **Booking of Laundry machine:** A resident of BS13 to do laundry has to physically visit the warden's office and pay money to get a physical coin which is needed to activate the machine. This is one of the many old conventional methods in use which can be replaced by digital solution. The importance of streamlining laundry processes for students is addressed by Andrej Kokelj in his paper and the challenges posed by traditional laundry methods, such as crowdedness and waiting times, highlight the need for efficient and user-centric solutions **(Kokelj 2018).**

- **Create/ Raise maintenance tickets:** Consider a scenario where a resident has a problem in his room and it needs an urgent fix, for example the breakage of the window, heating system stopped working or the light is dead. In such cases the resident usually raises a complaint by writing an email requesting an urgent fix or visiting the warden in person to explain the problem which consumes a lot of time. There is no quicker way of handling such events which impact the lifestyle of the residents if the problem is not resolved within the optimal time.

- **Furniture Giveaway:** In the case of residents moving out of the dormitory to migrate to different far-off cities restricts them from carrying all the furniture, equipment, and appliances they own, which mostly ends up in the trash. Instead, the things which are left behind can be donated to the other residents promoting sustainability and contributing to the environment. Maria Isabel Ordonez delves into the significance of creating communal spaces in residential areas to promote waste reduction through product reuse. It reviews five existing spaces where residents can share and repurpose items. The experiences from establishing such spaces in Gothenburg's HSB Living Lab highlight the importance of factors like location and accessibility, although the direct impact on waste reduction can be variable **(Ordonez et al. 2017).**

• **Centralized notifications:** There is no existing system that provides a centralized notification solution that collects and displays all alerts, messages, and notifications from many sources and apps in a single location or interface. The current method does not allow the warden to manage and keep residents informed about various events or updates without having to check several platforms owing to a centralized system that brings together the notifications from many apps or services into one area. As a result, the lack of organized and productive methods of handling information that comes in creates a communication gap between residents and the warden.

The next step involved framing the research questions based on the problem noticed:

Research question 1: How to digitalize the booking of laundry system to reduce the time and effort required by the residents using current conventional methods?

Research question 2: What technical stack and architecture would be the right pick? How to make the frontend more user centric and user interactive?

Research question 3: What is the most optimal and efficient solution to digitalize the maintenance ticket handling?

Research question 4: How can a digital solution promote sustainability and resource sharing by facilitating the donation of furniture and equipment by residents moving out of the dormitory?

Research question 5: Which approaches can be utilized to establish a unified notification system and messages, with the aim of improving communication between dormitory residents and wardens?

Research question 6: How do user-centred design principles, as emphasized in the paper by Ames, which was also previously discussed, aligns

with the development of an IT solution for enhancing the BS13 dormitory experience **(Ames 2001)** ?

Research question 7: What challenges and considerations should be addressed in the development and deployment of the web application's technical stack, software architecture, and database for optimal performance?

Research question 8: What are the best practices for ensuring the success and usability of the proposed IT solution for BS13 dormitory residents?

1.3 Methodology

Choosing the right methodology is a critical decision in the development of any technical product prototype. The technique provides a structured framework for project planning, execution, and management, ensuring that it proceeds effectively and efficiently.

The methodology guides the entire development process, establishing the project's structure, workflow, and interactions. The technique used affects many aspects of the project, from risk management to client satisfaction and product quality. Choosing the right methodology is critical for effective development. It guarantees that the project proceeds in a systematic manner, assisting the development team in setting clear goals, planning tasks, allocating resources, and managing time effectively. Furthermore, the approach adopted has a considerable impact on the product's quality and user happiness. The approach determines the extent to which end users are involved throughout the development process.

1.3.1 Comparison of different methodologies

When working as a lone developer producing a prototype, the ideal software project methodology is determined by several criteria, including the project's size, complexity, and desired working style. In the next sections, I'll compare three generally used techniques - Waterfall, Agile, and Rapid Application

Development (RAD) - in the perspective of an individual developer working on a prototype:

1.3.1.1 Waterfall model

The Waterfall model, created by Dr. Winston W. Royce developed the Waterfall model in 1970 as a traditional and sequential approach to software development. It separates a project into distinct phases, each one building on the preceding one's results. Progress is made in a linear fashion, much like a waterfall, making it difficult to return to previous stages once they have been accomplished. Figure 1.1 shows that waterfall model instills structure and orderly progression throughout planning, design, implementation, and testing, this methodology is best suited for well-defined projects.

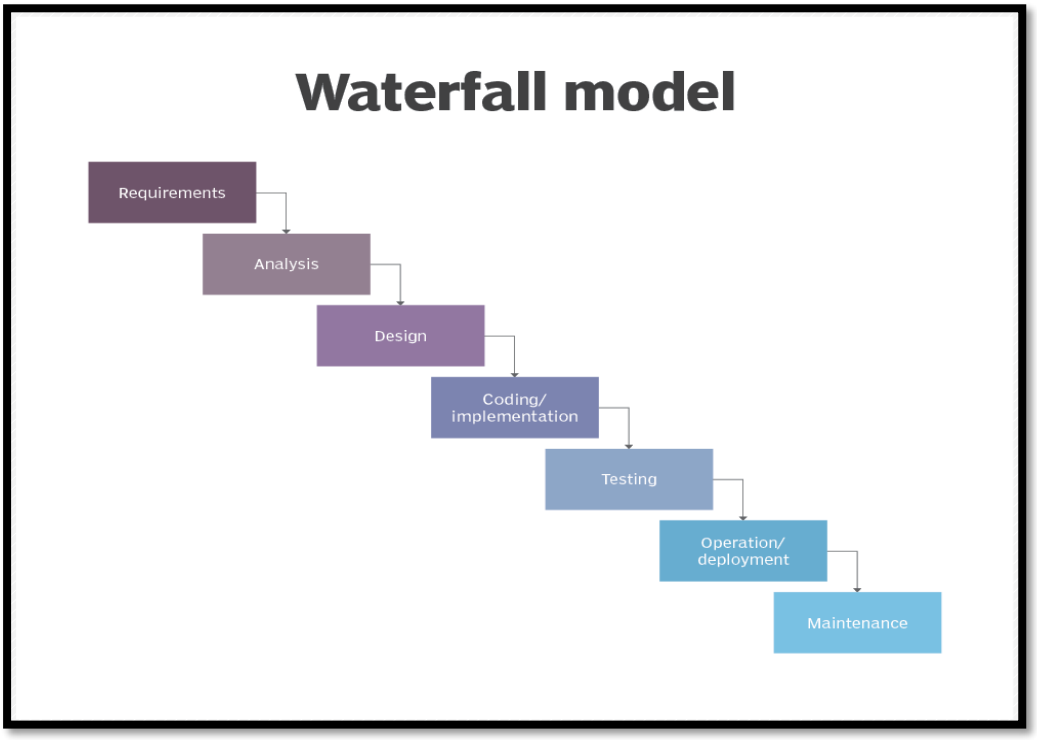


Figure 1.1 Different Phases of Waterfall model (Source: cdn.ttgtmedia.com)

Key points on Water-fall model:

- **Sequential Phases:** Waterfall consists of separate, sequential phases such as requirements, design, implementation, testing, deployment, and maintenance. Each phase must be completed before moving on to the next, ensuring a logical progression. This rigorous sequencing imposes a methodical development approach.
- **Well-Defined Requirements:** The model emphasizes well-defined and stable needs from the start of the project. Changing criteria later in the process can be complicated and costly. Clear requirements are essential for effective planning and execution.
- **Minimal Customer Involvement:** Customer involvement is often limited to the initial requirements gathering and final product delivery phases. There are few opportunities for continuous input and modifications. This reduces potential scope modifications but may result in less customer input.
- **Documentation-Heavy:** The methodology places a significant emphasis on documentation throughout each phase. While improving clarity, this documentation procedure can be time-consuming. Detailed documentation aids in keeping a clear project vision.
- **Quality Assurance:** Waterfall's testing step is critical for discovering and correcting faults before moving on to the next phase. This concentration on testing ensures a high-quality final product. Thorough testing helps to the product's overall quality and dependability.
- **Suitability:** Waterfall is generally regarded acceptable for projects with well-defined requirements and little uncertainty, which is common in industries with stringent regulatory standards.

Reason not to choose Waterfall model:

When a single developer builds a software prototype, the rigidity and limited adaptability of the Waterfall model can restrict the flexibility required for quick iterations and revisions. Agile approach and Rapid Application model (RAD), benefit solo developers by allowing for real-time feedback and efficient prototype creation. Due to the linearity of the Waterfall approach, agile alternatives are suitable for dynamic development settings and lone developers. Furthermore, Vishal Chandra has explained the lack of feedback mechanisms between stages in Waterfall can limit the flexibility required for dynamic development, making it less suitable for constantly changing and experimental projects like prototypes **(Chandra 2015).**

1.3.1.2 Agile

Agile software development is an iterative and flexible methodology that emphasizes adaptation and collaboration. Referring to Figure 1.2 it makes it clear that agile enables developers to work in cycles, incorporating improvements and feedback as they progress through the project.

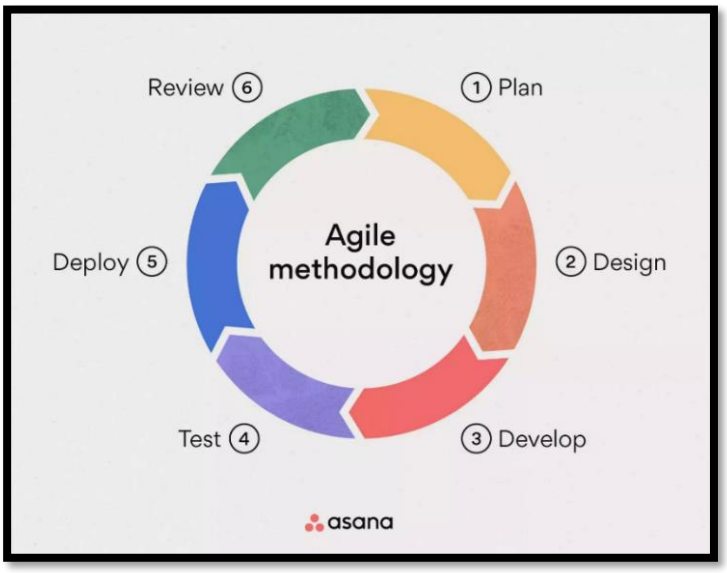


Figure 1.2 Agile Framework (asana.com)

Key aspects of Agile include:

- **Iterative Work:** Agile breaks down the project into smaller iterations, allowing for incremental development and regular feedback.
- **Customer Involvement:** Throughout the development process, customers or stakeholders are actively involved to ensure their demands are addressed.
- **Collaboration:** Cross-functional teams collaborate, promoting cooperation and quick problem-solving.
- **Adaptability:** Agile enables for shifting requirements and quick adjustments.
- **Consistent Delivery:** It encourages the delivery of functional components in each iteration.

Reason not to choose Agile model:

In conclusion, because of its flexibility, client engagement, and gradual approach, Agile is great for single developer building prototypes. Riesener emphasizes the benefits of Agile approaches in the software business, emphasizing client participation and flexibility **(Riesener et al. 2020).** When it comes to producing prototypes for technological systems, however, these methodologies face difficulties due to complex technical dependencies and evolving requirements. This involves the use of alternate ways to efficiently manage time constraints and prioritize complicated specifications. Other techniques, such as Rapid Application Development (RAD), may provide more simplified processes for independent developers.

1.3.1.3 Rapid Application development

Rapid Application Development (RAD) is a software development process that emphasizes rapid prototyping and development. As shown in the figure 1.3, it tries to shorten the development cycle by emphasizing rapid iterations, frequent feedback, and constant close collaboration with end users. RAD

simplifies the development by making it an appealing option for situations when speed is of the essence. Key Aspects of Rapid application Development:

- **Rapid prototyping:** Rapid prototyping is central to RAD, allowing developers to quickly construct working models of the application. This offers stakeholders with an actual version and assists in the refinement of requirements.
- **Iterative Development:** RAD uses an iterative approach to development, in which the project is broken into smaller modules or prototypes. Each cycle produces a functional component, which is eventually integrated into the final product.
- **User Involvement:** In RAD, user feedback is critical. End users participate actively in the evaluation of prototypes, giving remarks, and assessing that the application meets their needs.
- **Reduced planning:** In contrast to traditional approaches, which require considerable upfront preparation, RAD emphasizes on reducing early planning and documentation. It is more adaptive to shifting requirements and can take in new insights with greater speed.
- **Rapid Development (RAD):** As the name implies, RAD is all about speed. It is intended to speed up the development process, making it a great choice when time is of the essence.

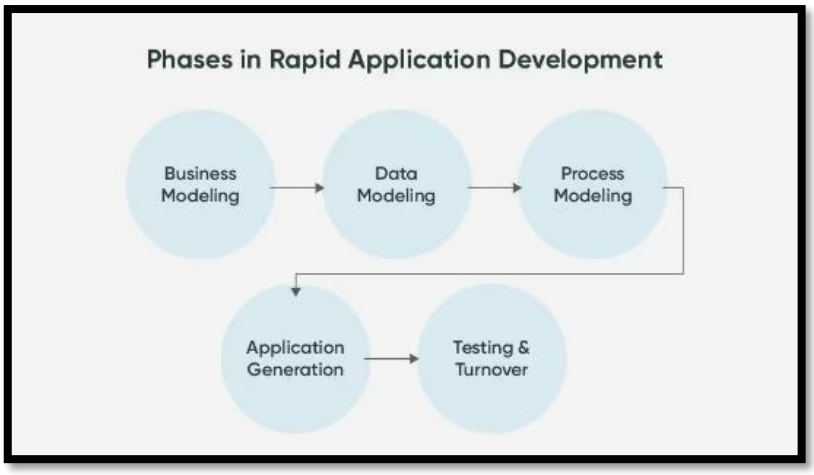


Figure 1.3 Phases in Rapid Application development (servicenow.com)

Reason to pick Rapid Application development:

Rapid Application Development (RAD) is ideal for single developers working on prototypes. Its emphasis on rapid prototyping, iterative development, and user participation enables a single developer to design and refine a prototype efficiently. There is less overhead in terms of extensive preparation and documentation with RAD, which could prove extremely beneficial for single developers. Nik Marsyahariani Nik Daud explains how RAD enables rapid progress and quick adjustments in response to changing needs or user feedback. The iterative technique of the RAD methodology, according to Nik Daud, enables quick development. Using RAD for DormEase is ideal as it is quick to adapt to changing requirements, improves user satisfaction, and accelerates the prototype development process **(Nik Marsyahariani Nik Daud et al. 2010).**

1.4 Personal Motivation

We were inspired to launch DormEase after personally witnessing the daily struggles of BS13 dormitory students. As an international student living in a different student residence, we are aware of the drawbacks of outdated systems.

What drives us is the enthusiasm that technology has the potential to revolutionize our daily environment. I'm enthusiastic about implementing IT solutions to make our community more sustainable and connected. Reduced waste and resource sharing are two things that are important to me. We encourage the ideals of reducing, reusing, and recycling by allowing residents to donate items they no longer need. The personal purpose for launching DormEase is to make a significant difference in the lives of our fellow students by making living in a dormitory an easier, more fun, and sustainable experience.

1.5 Technical Motivation

On a technical level, developing DormEase is an excellent opportunity for advancement as a programmer. This project serves as an opportunity for us to experiment, learn, and polish the coding skills. I'm getting better at problem-solving with each line of code. It forces us to think critically, break big projects down into simple steps, and come together with simple and effective solutions. The obstacles that we address along the way not only improve the programming skills but also give us a sense of accomplishment by contributing to a social cause.

1.6 Thesis Structure

The first chapter gives an overview of the master's thesis, mentions all the research questions which were addressed, gives us a comparison between different models and an explanation why RAD was chosen, followed by thesis structure.

The second chapter introduces design thinking and its' different phases. It explains in detail how each phase influenced the research phase. The third chapter dwells into the technical aspects of the development of the prototype by evaluating framework. It gives detailed information on the software architecture, server-side development, client-side development, database, the technical stack and mainly the reasoning behind choosing the tech stack.

In the fourth chapter, we discuss the working model of the prototype and the different features it offers for the user and admin. We use critical thinking to understand the strengths and weakness of the application. In chapter five gives us insights on the potential features which can be implemented and the areas where the application can be improved in the future to make it more user centric. Finally, it highlights the findings and brings this study to an end.

2. Related work and Surveys

2.1 Existing Systems

While researching on elated works, we came across a project “Dormitory Management System" developed by a group of students at JAHANGIRNAGAR UNIVERSITY in 2018, referring to their project report **(Khan 2018).** This project offers search capabilities as well as information regarding students' housing conditions and other aspects.

The project enables administration to provide a dorm room to a student and regulates room allocation. It is a digital platform that allows administrators and students to carry out all dorm-related activities online. This includes functions such as maintaining the details of old and new students, student payments, room distribution among students, and hall program management. The main source of inspiration for us is how the backend has been developed in a manner that it was highly secure to store and manage information on their database. There was one shortcoming, which we observed in their project is the front-end and how it could have been improved by making it more user-centric to provide a more interactive and immersive experience for the user **(Khan 2018)**.

When we started gathering Ideas on how to tackle the laundry booking features and to come up with an efficient solution to automate and digitalize it while eliminating the current method in use at BS13 which uses coins to start with the laundry machine. Setthawong illustrates how Bangkok's expanding urbanization has resulted in an increase in high-density living, particularly in apartment/condominium buildings. These buildings tend to have public coin-operated washing machines, which are popular among renters. However, the limited availability of machines during peak hours has resulted in long waiting lines. To solve this, the study presents a smart public coin-operated washing machine platform that uses IoT apps to improve tenant convenience. This platform consists of infrastructure, new features, and vacancy detection,

allowing residents to schedule their laundry even during busy hours and, as result, enhancing their quality of life **(Setthawong et al. 2023).**

The above-mentioned research gave us a foundation on how to tackle the laundry problem in BS13. This encouraged us to develop an online booking feature for the residents to book and plan their laundry appointments and avoid waiting for long hours for their turn to use the laundry machines. The digital solution We are proposing for BS13 can automated even further by integrating with IOT technology where the residents can book their laundry slot and pay for it online to get a unique code, which is needed to feed into the completely smart washing machines to activate them. This eliminates the use of physical coins to activate the machine. We have added this point in the future scope on how the DormEase feature we are proposing can be integrated with IOT technology to make the washing machines completely smart and automate it from booking of slots to payment, to getting the unique code required for activation of the laundry machines. This model eliminates the involvement of warden in the laundry process.

In conclusion, this section was necessary to understand and evaluate the pros and cons of the state-of-the-art model. With these insights We started the master's thesis research phase by having a clear idea on how to approach the problems currently faced by BS13 community and solve them with a prototype that was developed in the later stage of master's thesis.

2.2 Design thinking

According to Rikke Dam, Design Thinking is a method of problem solving that begins with a thorough understanding of the people experiencing the problems. It is about not taking things for granted and thinking differently to generate fresh ideas. This strategy is all about being creative and hands-on. It's useful for dealing with difficult or ambiguous topics since it pushes you to test out a variety of new ideas and changes your perspective on the problem. It also entails regularly testing and refining your ideas. Design Thinking is a user-

centred, creative problem-solving technique that incorporates empathy, asking lots of questions, and experimenting with ideas to solve the currently existing problems **(Dam and Siang 2021).**

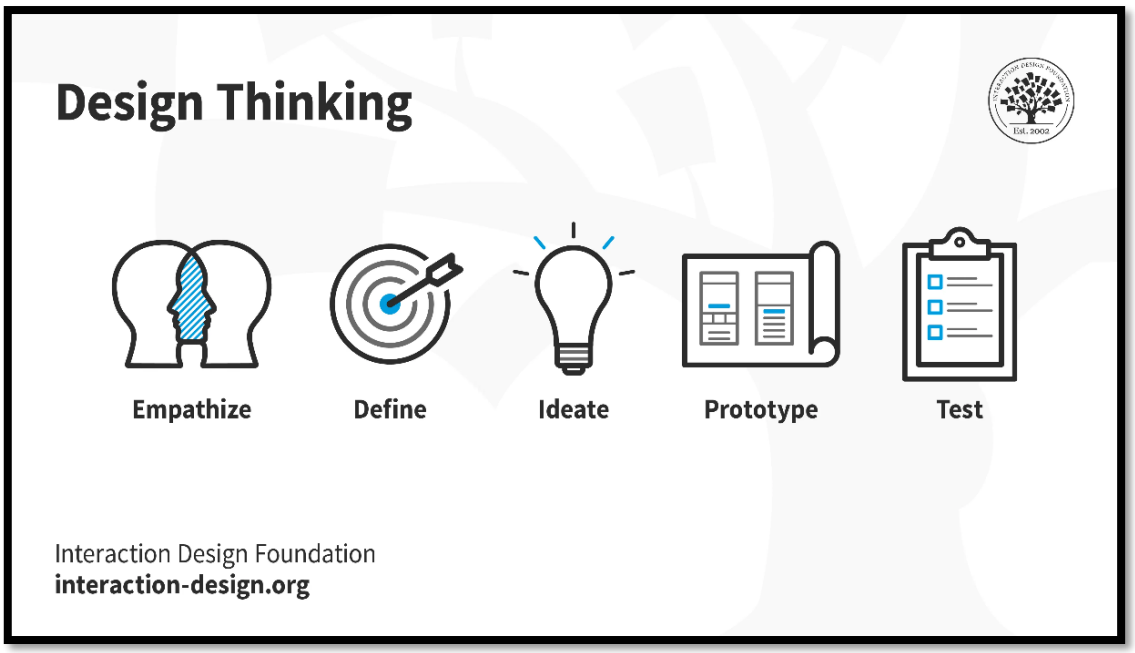


Figure 2.1 Phases in Design Thinking (Source: interaction-design.org)

As shown in the above figure 2.1, Design thinking typically encompasses the following stages:

1. **Empathize:** This stage requires recognizing the end users' viewpoints, emotions, and needs. Design thinkers perform activities that include interviews, observations, and surveys in the users' surroundings to obtain an in-depth understanding of their challenges.
2. **Define:** The issues and possibilities are defined in this phase based on the insights obtained throughout the empathy stage. It must be done to properly explain the problem statement, ensuring that it aligns with the needs and pain points of the users.
3. **Ideate:** Diverse teams create a wide range of creative concepts and practical solutions to the given difficulties during ideation. At this stage,

there are no incorrect answers, and brainstorming, mind mapping, and other techniques are employed to promote creative thinking.

4. **Prototype:** Prototyping is the process of turning selected proposals from the ideation phase into actual prototypes. These prototypes might be low or high resolution, ranging from simple sketches and paper models to interactive computerized mock-ups. Prototypes are crucial to evaluating and enhancing concepts.
5. **Usability Testing:** This stage involves putting prototypes in the hands of end users to gather feedback and insights. Testing allows for iterative adaptations and modifications by identifying what's effective. The input of users plays an important part in describing the outcome of the prototype.

Design Thinking is not a linear process; it often means revisiting to and iterating on prior phases in response to developments and user feedback. This dynamic approach places an emphasis on user fulfillment and is frequently employed in a variety of sectors, including product design, software development, business strategy, and social innovation. According to Matthew Sadiku, Design thinking skills will be required in future workplaces. Design thinking can lead to innovation Unfortunately, most designers still have a poor understanding of design thinking. So, it is important for us to understand every step of design thinking before taking complete advantage of the design thinking approach **(Sadiku et al. 2019).**

2.3 Question Driven Development

Question Driven Development is an approach that entails breaking down your development project into distinct questions and finding answers to them systematically. It acts as a step-by-step way to tackling your project, assuring clarity, and understanding at each stage of development.

How we Used QDD in Prototype Development:

- **Identifying Key Questions:** We began by compiling a list of questions relevant to the dormitory web application prototype. These questions were about specific functionalities and features that we needed to add to the prototype. For instance, we had a question, "How can we streamline laundry machine bookings?" With the question we could frame another question "What's the most efficient way to handle maintenance requests?". We repeated this process in a loop which helped us consider even the tiniest details.
- **Prioritizing and Addressing Questions:** We structured the list of questions, focusing on the most actionable ones first. we made a priority list of which ones need to be addressed in the initial stages of development. This method allowed us to focus on the most basic or foundational components of the prototype before moving on to more complicated features.
- **Iterative Problem-Solving:** Question Driven Development encouraged us to carefully answer each question. We sought solutions through studying study, learning from accessible resources, like documentations and asking colleagues and minimized watching video. Based on the responses, we were able to determine the structure and functionality of the web application.
- **Continual Progress:** The prototype developed incrementally as we addressed one question after another. This step-by-step approach enables to stay on track while learning more about the project's requirements.

How QDD was Beneficial:

Mary Paskhaver in her blog explains that Question-Driven Development is a strategy, which is focused on asking and responding to questions, substantially accelerated the development learning. It was also crucial in the efficient creation of the DormEase prototype. QDD provided with the opportunity to improve problem-solving abilities, break complex concepts down into simple

steps, and actively engage with programming foundations. This method enabled us to break free from tutorial hell and pursue practical by working on innovative tasks. Finally, QDD's simplicity and emphasis on exploration accelerated our learning and the development process which allowed to successfully produce DormEase prototype **(Paskhaver 2023).**

2.4 Summary

In this section we carried out a literature survey to study the existing models and different solutions which gave us an understanding of the state of the art and how we can use it to solve the current problems faced by BS13 residents. We introduced the readers to Design thinking methodology and its different phases. In the later stages of the report, we will see how we have used each phase to breakdown the problem into several steps and find the solution using a design thinking approach. A simple principle, Question driven development is explained in depth, how asking the right questions, and finding the answers along the phase of development can help us to learn things in a practical manner, using this principle we approach the development phase in the next section.

3. Development

3.1 Implementing Design thinking.

3.1.1 Empathize

The "Empathize" stage in the development of DormEase was essential and enlightening. It was all about gaining a deep understanding of the BS13 community's needs, concerns, and challenges. During this phase, we focused on various activities and approaches to imagine ourselves in the residents' shoes and truly understand their experiences.

In the research conducted by Fleischmann to explore the availability and impact of mobile apps designed for the elderly, focusing on those aged 60 and above, considering the development of empathy among student designers for their target audience, helped them to come up with creative and better solutions **(Fleischmann et al. 2012).** So, learning from that research we understand how important it is to truly empathize and performed the following steps to empathize with the residents:

Surveys and Interviews: we began by conducting online surveys but didn't get much response, so we focused more on one-on-one interviews with the residents of the BS13 dormitory. These interactions allowed us to interact with the students personally, hear their experiences, and learn about their day-to-day challenges. It was an important step in gathering their views, feelings, and experiences regarding the dorm's existing systems and processes.

Observation: In addition to surveys and interviews, we attempted to observe the activities of the residents within the dormitory. This included spending time in the dorm, seeing how they engaged with the existing system and with one another, and identifying any obvious complications or inefficiency. Observations provided us with information that we could capture in the one-on-one interview on their lives.

Empathy-Building Activities: To further understand the BS13 community, we engaged in activities that allowed to see the world from their perspective. To better understand the issues they faced, they briefly adopted some of the current manual methods they use.

Identifying Pain Points: We found various pain areas and issues faced by the residents because of this sympathetic engagement. These ranged from the time-consuming process of using laundry machines to the time-consuming manner of raising maintenance tickets. We also saw the possibility for sustainability by addressing furniture giveaway as a waste reduction strategy.

Active listening, open-mindedness, and a need to understand the residents needs and problems identified during the "Empathize" stage. We were able to take down the features that users would like to see in the prototype by gaining these useful insights, as shown in Figure 3.1. the Column "Features" reflects the features that the user would want to see in the prototype, and Column "Occurrence" shows the number of times the feature was requested by residents.

Features	Occurance
Online Laundary Booking	9
Online Kitchen Booking	4
Warden at Office status	3
Centralised notifcation	7
Wifi Sharing	5
Community Channels similar to reddit	2
Frequently asked Question section	6
On-Boadrding/ Off-boarding	2
Sepearte Chatrooms	3
Shared Cooking Plan	5
Online Market place to give away things	7
Online platform like meetup for Weekend events	3
Online platform to rasie maintenance tickets like JIRA	7
Blog on Garbage segregation	9
Blog on Your Guide to Fire Emergency in Student Dorms	9

Figure 3.1 Features desired by residents and the occurrence count of each feature. (Source: Author)

3.1.2 Define

At this stage, we identify specific issues and opportunities based on the understandings gained throughout our study of the needs of BS13 residents. It is critical to clearly define the issues we are going to address, assuring that they align with the demands and challenges of the users.

3.1.2.1 Feature Extraction

The next stage was to describe the exact obstacles we wanted to address after we had a solid understanding of the residents and their needs. This requires an in-depth examination of all the data gathered through surveys, interviews, and observations.

We looked through this data with the goal identify the most pressing and significant issues that the DormEase application could address. In figure 3.2, we segregate the user requirements in different categories to prioritize the requirements. It was critical that we focus on issues that would make significant impacts in the lives of the residents.

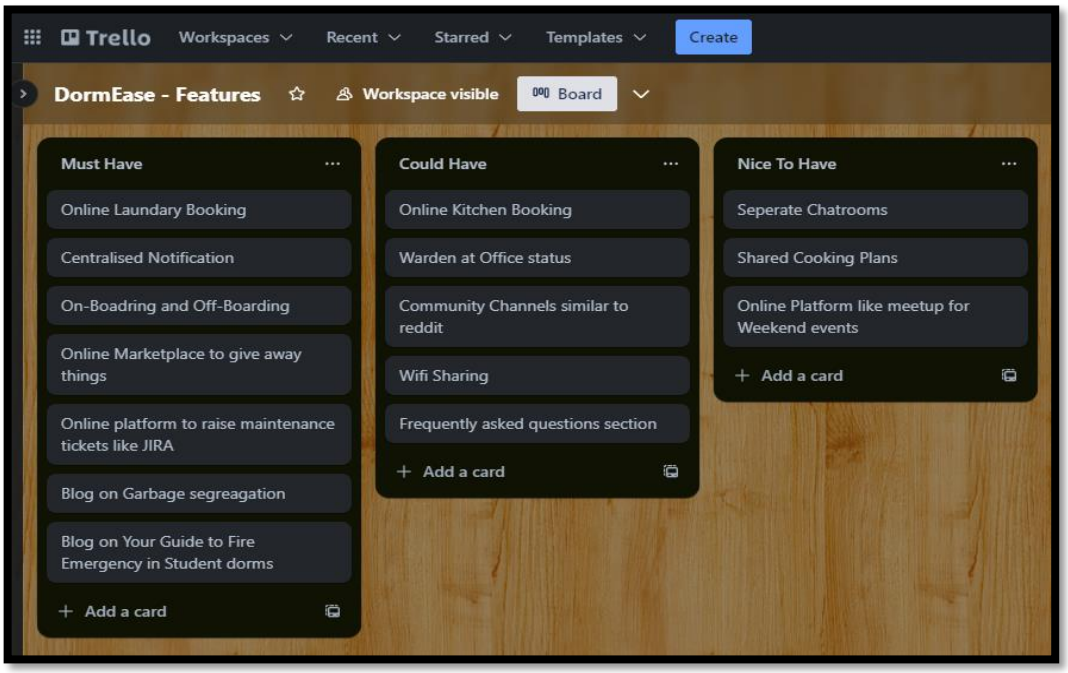


Figure 3.2 Prioritizing the features. (Source: Author's Trello Board)

Three main challenges stood out:

1. **Booking Laundry Machines:** we observed that residents had to physically visit the warden's office to pay for a physical coin to activate the laundry machines. This process was cumbersome and inefficient. We aimed to simplify and digitize the laundry machine booking system, saving residents time and effort.

2. **Maintenance Ticket:** When residents had issues in their rooms, such as a broken window or malfunctioning heating system, they had to either write an email or visit the warden's office in person to report the problem. This manual process consumed a lot of time. The goal is to streamline the creation of maintenance requests, ensuring that urgent issues can be reported and addressed more efficiently for the residents of BS13 community. The study conducted by Christa explains the importance of software maintenance and service delivery in ensuring client happiness. It stresses the importance of examining the large number of incident-related tickets, which contain useful information. The study investigates ticket analytics with the goal of identifying patterns and proposing techniques for resolving tickets with varied levels of relevance and priority **(Christa and Suma 2017).**

3. **Centralized Notifications:** The existing system lacked a unified method for collecting and displaying alerts, messages, and notifications from various sources and apps. This meant that residents had to check multiple platforms to stay informed about events or updates. We aimed to bridge this communication gap by providing a centralized notification solution.

   Identifying these key problems was a crucial step in the development of DormEase. It allowed us to narrow our focus and work on solutions that would significantly enhance the daily lives of the residents.

3.1.2.2 Creating Personas

Persona creation is an important phase in the DormEase development process. Based on genuine user data and insights, these personas show fictional, yet highly real personalities. According to Bagnall, personas reflect average users and are based on real-world research, are valuable tools in the design and execution of interactive systems. These personas guide the design process and allow effective communication across design, marketing, and engineering teams, enabling a user-centered approach to interactive system development **(Bagnall 2007).**

Personas contribute to DormEase being a customized solution that caters to the unique requirements of BS13 residents and the house warden. Personas, in essence, make DormEase more user-centric, ensuring that it caters to the people for whom it was developed.

Persona 1:

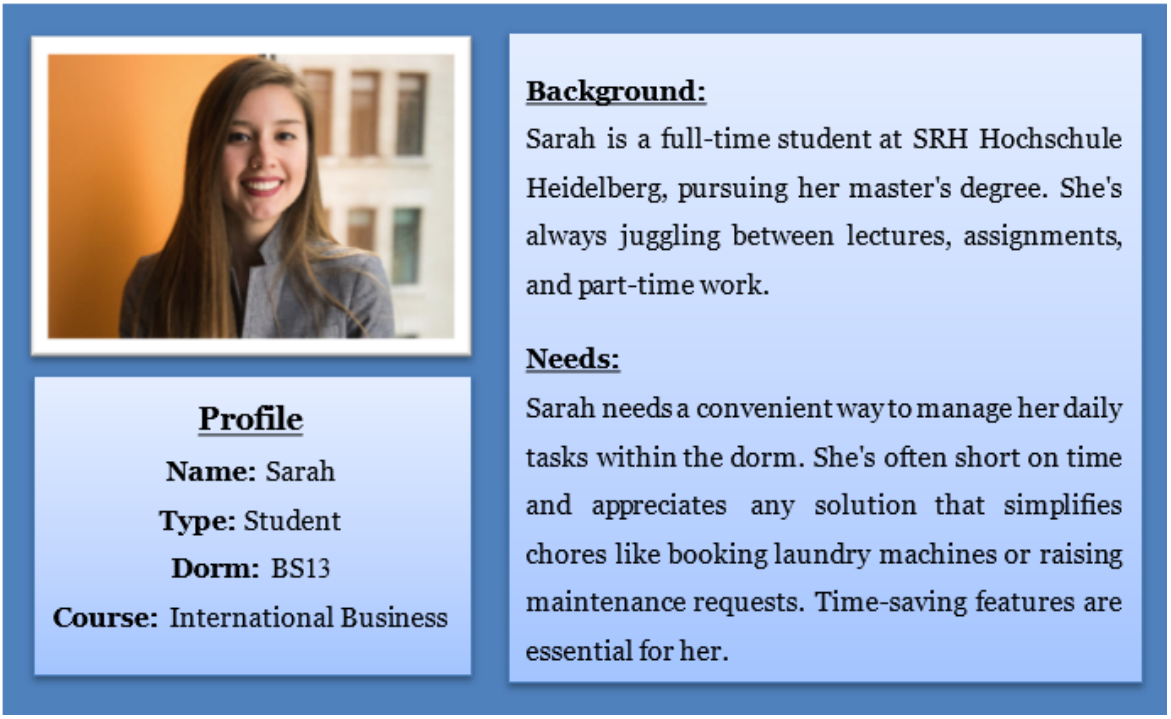


Figure 3.3 Persona 1 (Source: Author, Persona image: unspalsh.com)

Persona 2:

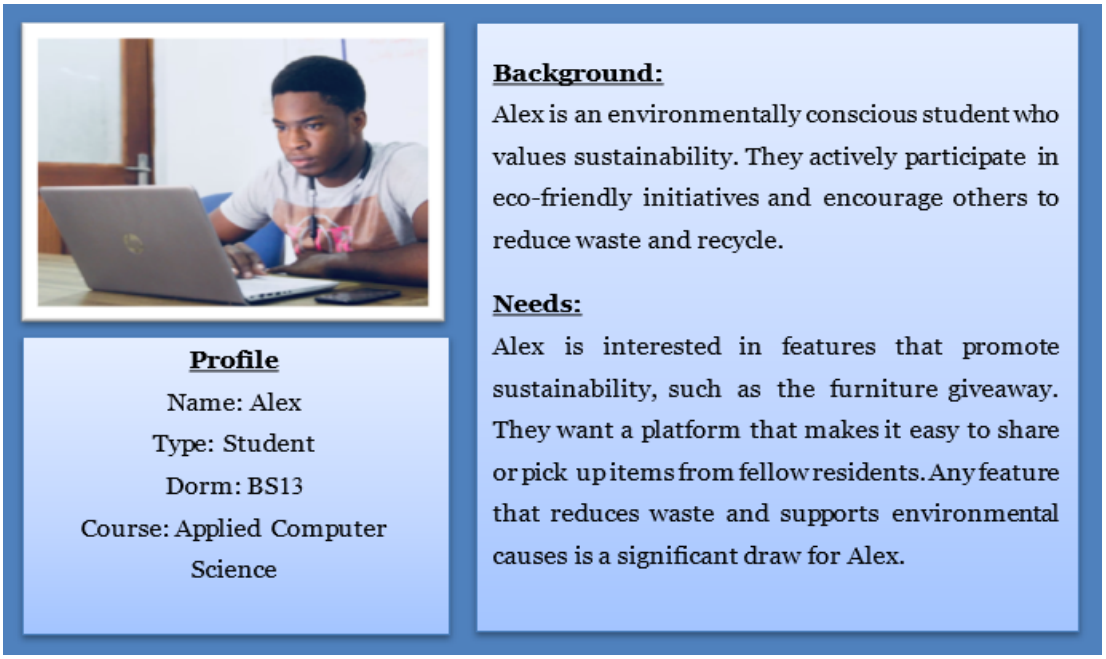


Figure 3.4 Persona 2 (Source: Author, Persona image: unspalsh.com)

Persona 3:

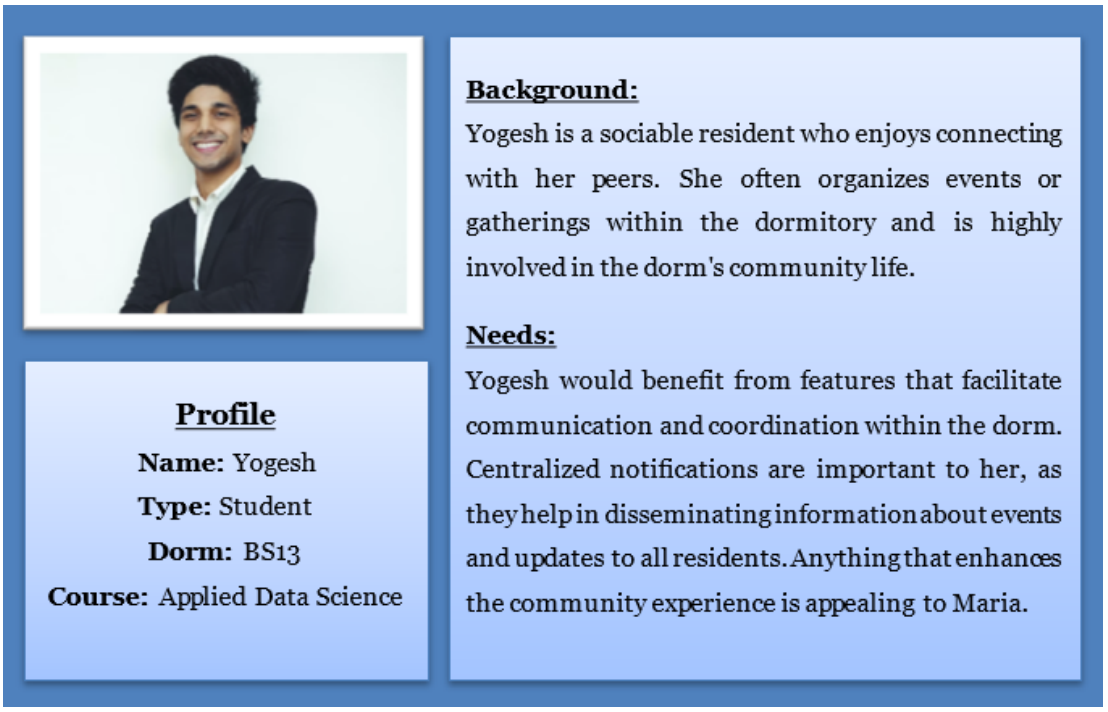


Figure 3.5 Persona 3 (Source: Author, Persona image: unspalsh.com)

Incorporating clearly defined personas into the DormEase development process has given us a more precise and user-centric approach. We've established a solid foundation to develop a solution that caters to our fundamental users' wants, decisions, and pain points by fully understanding and empathizing with their needs, practices, and pain points - the residents of BS13 dormitory and the house warden. Persona's act as an important guidance, assisting us in making sensible choices at each stage of development. DormEase is a simple but very effective user-friendly application that addresses the issues experienced by the BS13 community, thanks to this user-focused thinking. We proceed to the next phase "Ideate" with these personas into consideration.

3.1.3 Ideate

The Ideation phase of DormEase development is an exciting journey in which creativity has no boundaries. It is the stage at which different teams gather to brainstorm and develop a wide range of unique ideas and potential solutions to the problems we've identified. Consider it a room full of eager minds, all focused on developing novel methods to make DormEase a more user-friendly, efficient, and solution-driven application.

During this stage, there are no wrong answers, and the more of them the better. It's all about promoting creative thinking and applying methods that include brainstorming and mind mapping to promote creativity. These approaches allow us to examine an extensive variety of opportunities, allowing us to study multiple viewpoints and describes while solving the issues confronting the residents and house warden at BS13 dormitory.

In this long discussion, we'll look at how the Idea generation phase works out, the value of collaborative thinking, and how different ideas eventually create the way for creative solutions. Let's look at ideation and how it performs an

integral part in the creation of DormEase, making it more than just a practical solution that enhances the lives of its users.

3.1.3.1 Brain Stroming

Brainstorming was an important element of the Ideation step throughout the journey developing DormEase. The Research done by Al-Samarraie examines the implementation of brainstorming strategies such as traditional, nominal, and electronic brainstorming in higher education. It assesses their success, purpose, problems, and potential solutions in discipline-specific contexts. The findings provide educational decision-makers and students with insights into how to improve brainstorming techniques in university settings **(Al-Samarraie and Hurmuzan 2018).**

As we discovered the problems and opportunities based on the understandings we had gained earlier, this innovative approach allowed us to explore a wide range of ideas and potential solutions. The beauty of brainstorming is that there are no bad answers, and it's a space where brain could think endlessly, and our creativity had no limitations.

Approach to Brainstorming in DormEase Development:

As a single developer working on the project, the brainstorming sessions usually began with a clear problem statement based on the insights we obtained during the Empathize and Define phases. This problem statement worked as a guide for me, directing the creative energies in an accurate goal.

The Rules of the Brainstorming:

- **Quantity Over Quality**: The main purpose was to develop an extensive list of ideas. Initially, it was about allowing every idea, no matter how strange, to find its place in the notepad. More ideas meant more opportunities to discover significant solutions.

- **No Self-Critique:** All ideas were dealt with consideration and enthusiasm during brainstorming, with no self-criticism or judgment. This enabled us to openly submit ideas and shared it with our friends without worry of being rejected.
- **Building on Ideas:** A strange idea can sometimes create a moment of inspiration. Ideas were frequently used as building blocks in the brainstorming sessions. One thought could inspire another, leading to new combinations.

Mind Mapping:

We used mind mapping as a technique to visualize our thoughts. It benefited us in organizing our thoughts and connecting them. Mind mapping helped DormEase see the whole picture, find connections between distinct concepts, and learn how they related to a complete solution.

3.1.3.2 Proposed Solution

1. **Booking of Laundry Machine:**

   Proposed Solution: Introduce a Digital Reservation System

- Develop a feature within DormEase that allows residents to reserve laundry machines through a user-friendly interface on the application. According to the research paper by Li Hesen, that investigated the laundry challenges experienced by college students in dorms, used various data collection methods, and underlined the need of categorizing group lifestyles in building a more effective and environmentally friendly laundry system. The study sheds light on how symbiotic design and shared services might meet students' laundry needs in a hostel setting **(Hesen and Ying 2019)**.
- Eliminate the need for physical coins by integrating an online payment system. Residents can make payments for machine usage directly through the app.

- Implement a notification system to alert residents when their reserved time slot is approaching, minimizing the need for physical presence, and waiting.

2. Create and Manage Maintenance Tickets:

Proposed Solution: Streamlined Maintenance Request System

- Integrate a maintenance request feature within DormEase, enabling residents to report issues and request repairs directly through the app.
- Include categories and urgency levels for maintenance requests, allowing residents to specify the nature and importance of the issue.
- Develop a system that assigns, prioritizes, and tracks maintenance requests, ensuring timely resolution and effective communication between residents and maintenance staff.
- Implement status updates to keep residents informed about the progress of their requests, enhancing transparency and user satisfaction.

3. Furniture Giveaway:

Proposed Solution: Marketplace Platform

- Create a dedicated section in DormEase for furniture sharing, where residents can list items, they no longer need.
- Implement a search and request system that enables residents to find and request available furniture.
- Include a communication channel for residents to coordinate pickup or delivery of shared items.
- Promote sustainability by encouraging residents to share and reuse furniture, reducing waste and benefiting the environment.

4. Centralized Notifications:

Proposed Solution: Unified Notification Hub

- Develop a centralized notification hub within DormEase that aggregates alerts, messages, and updates from various the house master.
- Enable residents to customize their notification preferences, allowing them to receive relevant information according to their interests and needs.
- Create a feature for the house warden to send announcements and updates to all residents through the notification hub.
- Implement notification management tools to ensure residents can stay informed without the need to check multiple platforms, bridging the communication gap between residents and the warden.

These proposed solutions align with the defined problem statements and address the specific challenges faced by BS13 dormitory residents. By combining such features into DormEase, the application becomes a complete package that simplifies daily tasks, improves user experience, and promotes the dormitory communities towards sustainability and effective communication.

3.1.4 Prototype

In the development of DormEase, the "Prototype" phase indicates the transition from conceptual concepts to actual, tested representations of the proposed solutions. This phase is important for converting the submitted ideas into something concrete that can be evaluated and improved. The study conducted by A. Berglund emphasizes the need for prototyping in design initiatives, particularly in product innovation engineering. It is a crucial instrument to foster creativity and innovation. Collaborative prototyping allows for the incorporation of feedback from peers, resulting in useful input and increased innovative abilities. This highlights the importance of

prototyping in product innovation engineering learning experiences **(Berglund and Grimheden 2011).**

We overcome the gap between concept and reality at the "Prototype" phase. It involves creating prototypes that vary from simple sketches to interactive digital mockups, which are then tested, improved, and iterated upon. This method assures that the proposed solutions are not only feasible, but also user-friendly and applicable to the BS13 community's particular requirements.

3.1.4.1 Low Fidelity

An interesting Blog by **Dam and Siang (2023)**, Low-fidelity prototypes are the first drafts of our design ideas, and they are a critical building component for the development of DormEase. These prototypes are easy and basic, usually consisting of primitive sketches, paper models, or wireframes. They provide an essential picture of our established features without getting into technicalities. The main aim of low-fidelity prototypes is to set up the application's basic structure and layout. They can be an entry-point for concepts and evaluating the overall flow of DormEase.

The figure3.6, is the low fidelity model of Admin panel which was initially designed. Figure 3.7 shows the low fidelity model of Resident home including the navbar, footer and the card structure, buttons that would be later implemented in the final prototype.

Figure 3.8 gives us a rough sketch of the marketplace page that lists all the products and figure 3.9 gives us insights on how a detailed page of a single product would look like in the final version of the software.

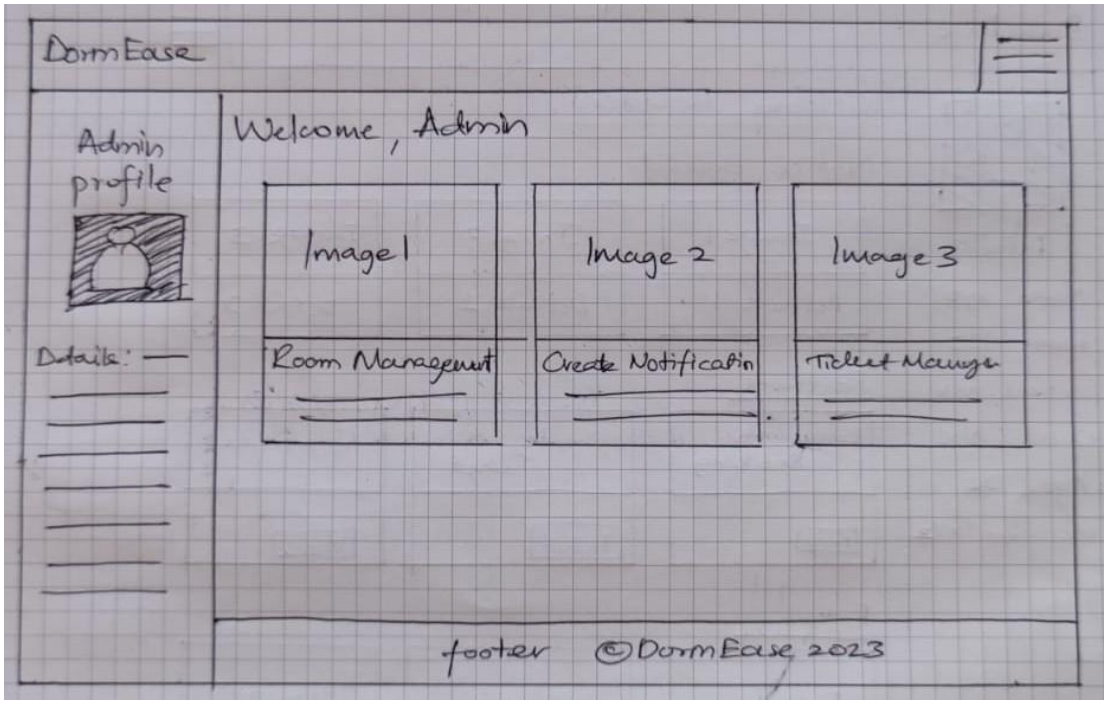


Figure 3.6 Low fidelity prototype of Admin home (Source: Author)

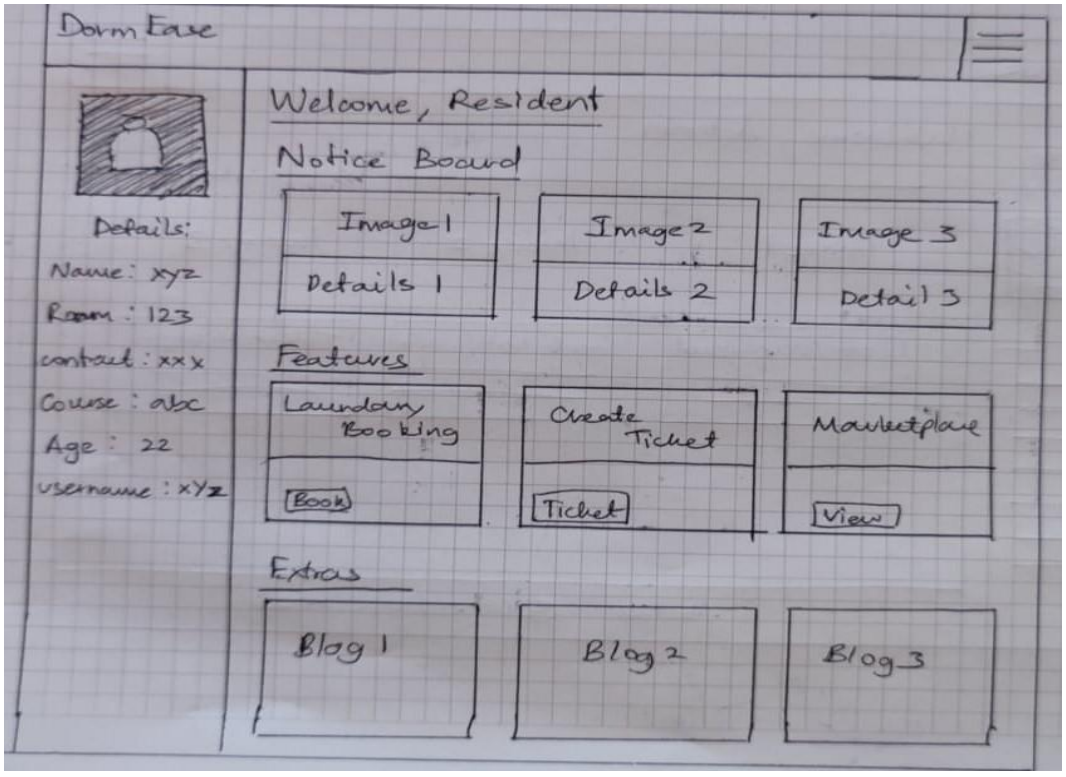


Figure 3.7 Low fidelity prototype of Resident home (Source: Author)

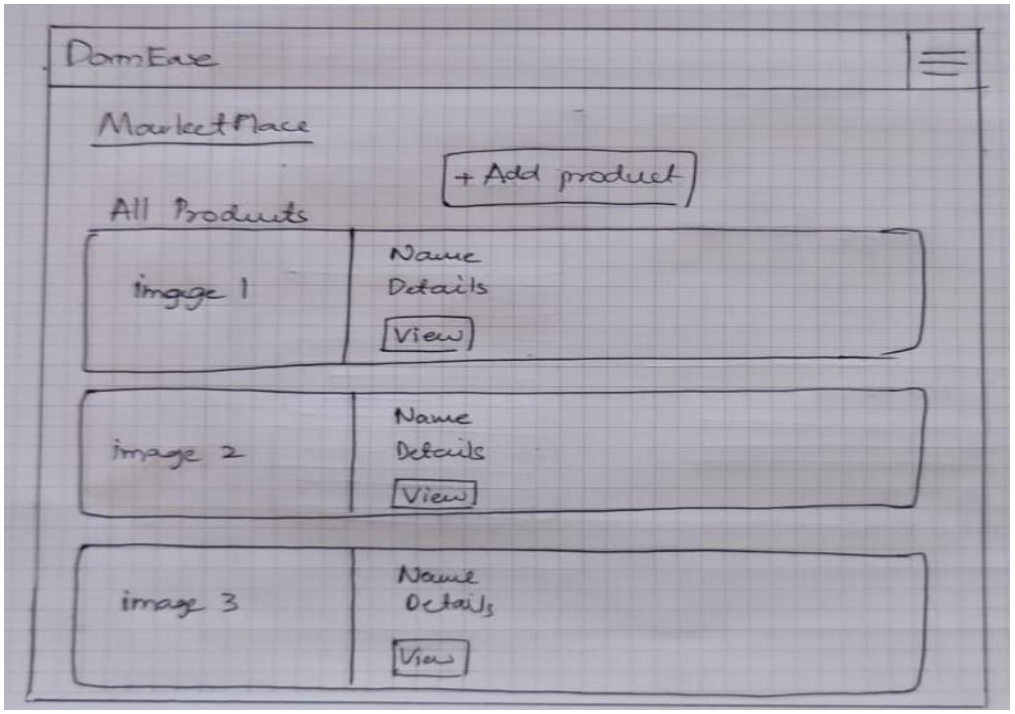


Figure 3.8 Low fidelity prototype of Marketplace main page (Source: Author)

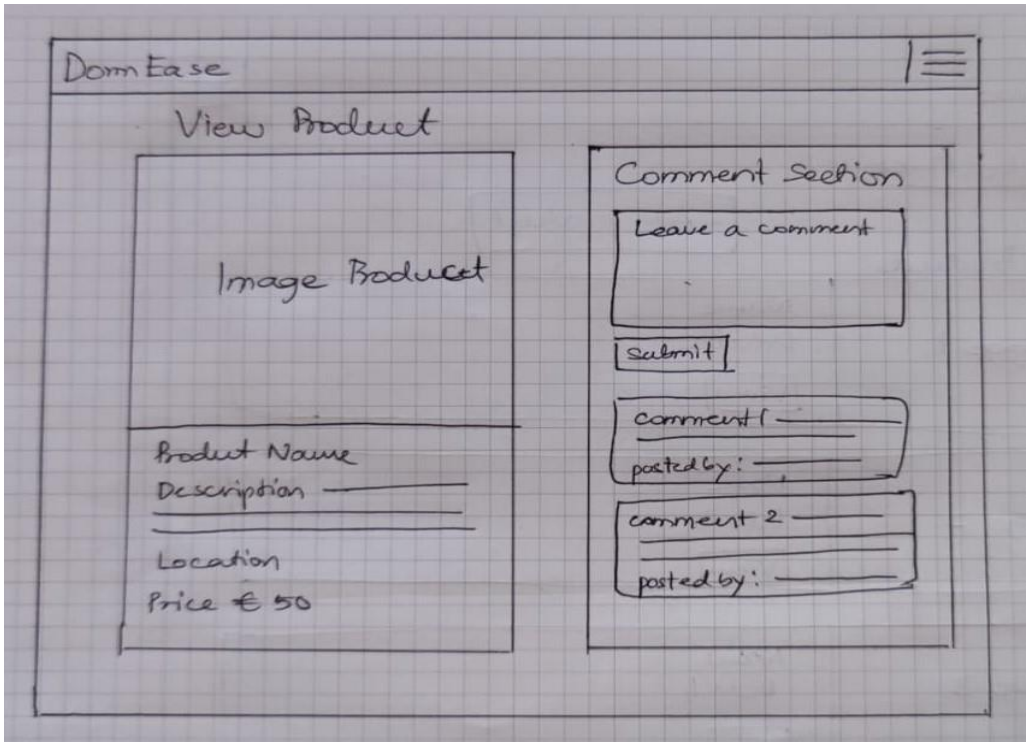


Figure 3.9 Low fidelity prototype of Product view page (Source: Author)

3.1.4.2 High Fidelity

According to Zarmeen Zehra's blog, High-fidelity prototypes are where we start developing the DormEase concept to life. These prototypes are more complicated and interactive, and they often look like the finished product. They expand on the basic structure by including additional elements such as colors, graphics, and interactive features. High-fidelity prototypes allow us to go even more into the application's functionality and user interface, providing a more precise representation of the user experience. These prototypes are essential in evaluating and improving all aspects of DormEase to ensure it meets the needs and expectations of its residents **(Zehra 2023).**

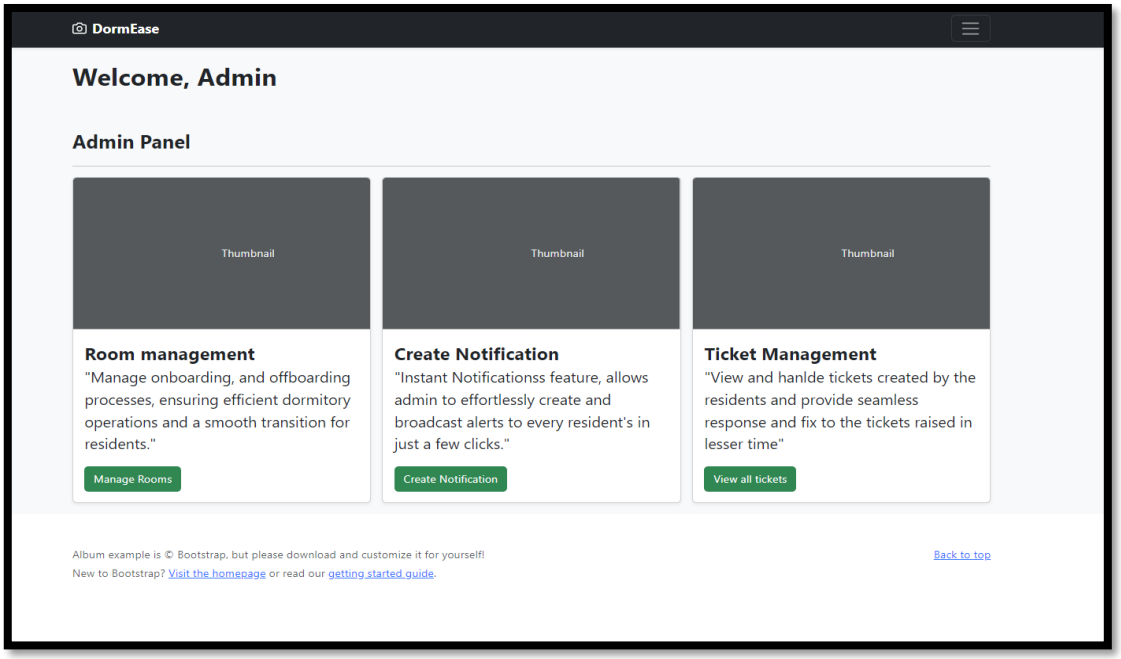


Figure 3.10 Admin Homepage (Source: Author)

Figure 3.10, Shows the high-fidelity prototype for Admin homepage. It shows admin having the cards on the page namely Room Management, Create Notifications and Ticket Management.

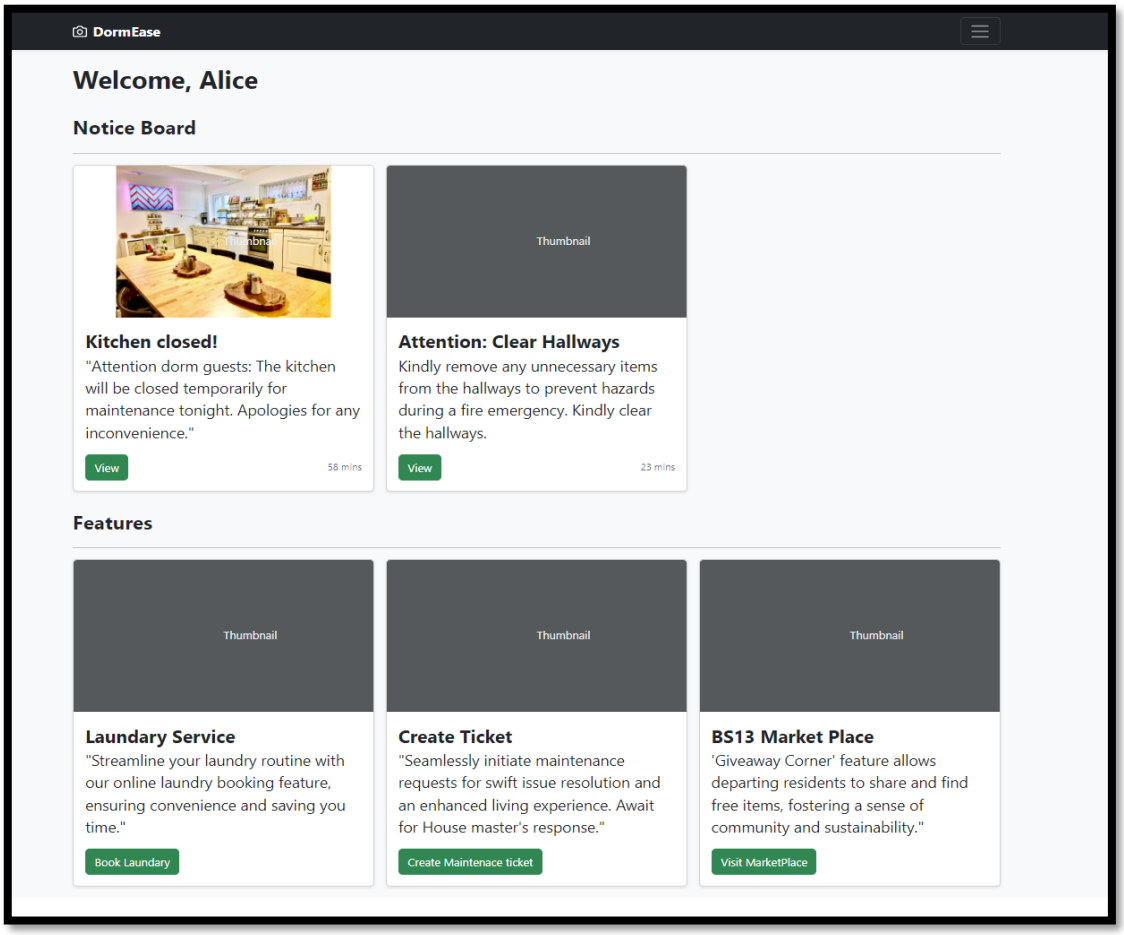


Figure 3.11 Resident Homepage (Source: Author)

Figure 3.11, Shows the high-fidelity prototype for Admin homepage. It shows users having a notice board on top to be updated with all the latest notifications, the second section is features which enables user to book online Laundry, create maintenance ticket and Marketplace to buy and sell goods.

For creating the DormEase prototype, we opted for a combination of HTML, CSS, and Bootstrap. This choice was driven by our familiarity with these technologies and our comfort in using them to achieve the goal of developing a high-fidelity prototype. While we may not have been well-versed in using frontend wireframing tools like Figma or other similar applications, we found that HTML, CSS, and Bootstrap allowed us to effectively translate our design concepts into a practical and visually appealing prototype for DormEase. These

technologies served as our tools for crafting a high-fidelity representation of the application's user interface and functionality.

3.1.5 Usability Testing

Usability testing is an important phase in the development process for DormEase. It involves displaying prototypes to end users, in this case BS13 residents, to get useful feedback and ideas. This stage's primary purpose is to make sure that the web-based application not only meets their needs but also provides an outstanding user experience.

In the Second Edition by **(Barnum),** the author has shown the importance of usability testing. According to the author, in the modern era of digital age, users have minimal tolerance for poor user experiences, as initial impressions on websites are created in just 50 milliseconds. Most product returns occur not because of problems, but because users can't figure out how to utilize them or have buyer's remorse. This emphasizes the significance of designing products with the user in mind. This explains how important usability is for developing a good prototype.

Importance of Usability Testing:

1. **User-Centric Design:** DormEase was conceived and developed with a user-centric approach. Usability testing is the bridge that connects the design and the residents' expectations. It ensures that the application aligns perfectly with their preferences, behaviours, and objectives.

2. **Validation of Assumptions:** Throughout the development stages, assumptions were made about the features, layout, and functionalities. Usability testing serves as a reality check, either confirming those assumptions or revealing areas that require adjustment.

3. **Identification of Pain Points:** Usability testing helps pinpoint any pain points or challenges that the residents might face when using DormEase. Whether it's the ease of booking laundry machines or navigating the application, these insights guide improvements.

The Usability Testing Process:

1. **Selecting a Diverse Group of Test Users:** It's essential to include a diverse group of BS13 residents as test users to ensure a comprehensive assessment. This group should represent various backgrounds, preferences, and tech-savviness levels.
2. **Setting Clear Testing Objectives:** Before testing, clear objectives and success criteria are established. What should the residents be able to do? What are the expected outcomes? This ensures that testing remains focused and structured.
3. **Guided Test Sessions:** Each test session is guided but not prescriptive. Test users are encouraged to explore the application naturally, while a facilitator observes and records their interactions. This approach allows for unbiased feedback.
4. **Feedback Collection:** Users are encouraged to express their thoughts, likes, dislikes, and difficulties during the testing. These feedback sessions are documented and analysed systematically. This phase is the most important phase.

Iterative Improvements:

DormEase receives a series of incremental improvements following feedback from users. Based on the feedback, this might involve changes to the user interface, product updates, or issue repairs. DormEase continues to evolve as an outcome of the iterative nature of the process, becoming more user-friendly and efficient with each round of testing and modification.

Shaping the Final Solution:

DormEase's final version is greatly influenced by the insights and feedback collected during usability testing. By responding to user issues, improving movement, and fine-tuning features, the application becomes an accurate representation of what residents ask for and desire. It precisely aligns with the primary goal of streamlining everyday activities and enhancing their overall BS13 experience.

The Usability testing phase is explained in the later part of the report as we need to look deep into the development and technical aspects of the prototype before we do the testing. After we have a prototype, we handle it users to play around with it and give us valuable feedback which helps us to make the app.

3.2 Design

3.2.1 Module Description

The module description provides in-depth information about how the prototype works. In this approach, the housemaster serves as the application's administrator, responsible for its administration and management in accordance with university rules and regulations. On the other hand, we have the Residents, who we simply refer to as Users because they are the major users of DormEase.

At the initial workflow stage, both the Admin and User are directed to the login page. It's essential to note that users only have the option to log in, while the registration feature is intentionally omitted. This measure enhances application security, ensuring that only residents of BS13, onboarded by the admin using a dedicated feature discussed later, can access the system. This approach effectively prevents non-residents or external users from gaining unauthorized access to DormEase.

At the beginning of the workflow, both administrators and users are directed to the login page. Referring to Figure 3.12 below, let's explore admin module:

- Once the admin logs in, the admin is greeted by admin panel page, offering three primary functionalities. The first feature, "Onboarding," empowers the admin to initiate the onboarding process for new residents, granting them access to the DormEase application.
- The admin panel also includes a second feature, "Create and Manage Notifications," where administrators can compose new notifications for distribution to DormEase users. They can also view the list of applications from which they can remove outdated notifications.
- The third feature in the admin's toolkit is "Maintenance ticket Management." These tickets are generated by residents when they require assistance or when maintenance is needed to address a specific issue. Once a user has submitted a ticket, the admin can dispatch a maintenance team for immediate service. Upon completion of the service, the admin can return the ticket to the user for authorization. Residents have the option to close the ticket if they are satisfied with the service or reopen it if further attention or improvements are required.

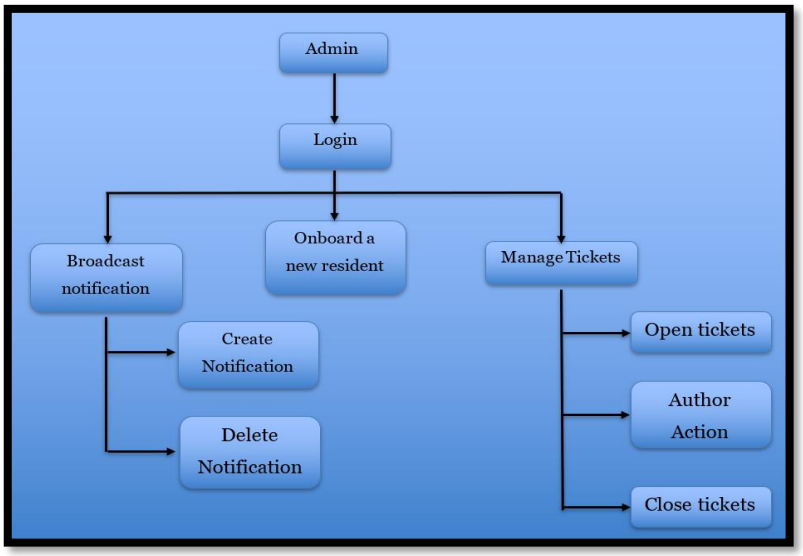


Figure 3.12 Admin module workflow (Source: Author)

Referring to the figure 3.13, On the user's side, the journey begins with the login page, where, upon entering the correct credentials, the user is granted access to the application's home page. The user is presented with a range of features, each tailored to address specific problem statements.

- At the top of this page, the user has visibility into notifications shared by the admin, ensuring they stay informed about the latest news and updates in the BS13 community.
- One of these features is the "Laundry Booking" functionality, which simplifies the process of reserving laundry machines.
- With just a few clicks in their web browser, the user can easily book a laundry machine. Additionally, the user can conveniently access a record of their past and current bookings.
- The second feature empowers the user to initiate maintenance requests promptly. This functionality allows the user to generate maintenance tickets when there is a need for urgent repairs due to breakage or malfunction.
- The user's creation of a maintenance ticket serves as an alert to the admin, indicating that service is necessary.
- Upon receiving the maintenance request, the admin takes action by moving the ticket to the "Author Action" section once the service has been provided. The user then has the option to either close the ticket if they are satisfied with the service or reopen it if further maintenance is required.
- Additionally, the user can create multiple tickets and conveniently review their current tickets, which are categorized into three sections: "Open Tickets," "Author Action Tickets," and "Closed Tickets."
- The third feature, known as the "Marketplace," grants users the ability to list their products for sale, whether it's for free or at a specific price. Users can also make edits to their product listings if corrections are needed. Additionally, users can browse through products posted by other users. If a

user is interested in purchasing an item, they can engage in the comment section to express their interest or make inquiries.

- It's important to note that all users can leave comments on product listings, but they can only delete comments that they have posted from their own profile.
- Users also have access to a section where they can explore various blogs covering topics such as waste segregation and guidelines to follow in the event of a fire emergency.
- Additionally, there are numerous other engaging blogs available that are directly relevant to student dormitory life. These blogs aim to educate users on how to comfortably navigate dormitory living while also promoting environmental cleanliness and sustainability.

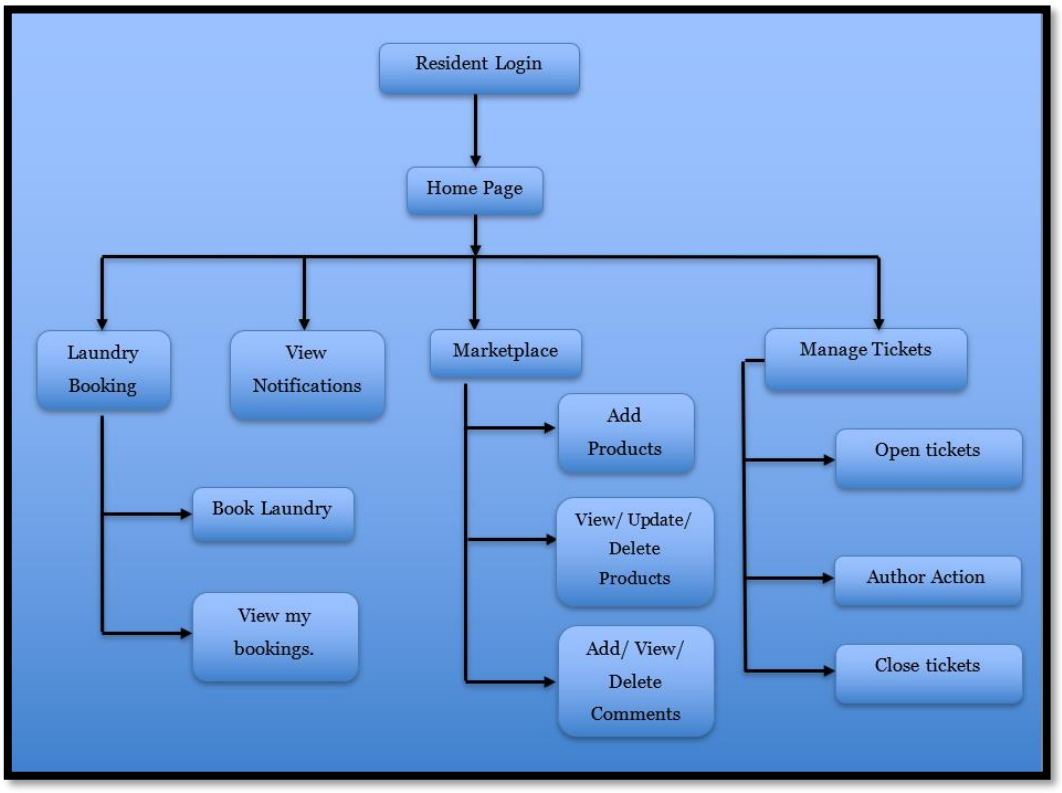


Figure 3.13 User module workflow (Source: Author)

3.2.2 Architecture

DormEase's architecture was carefully created and integrated in order to develop a user-friendly, dependable, and efficient system for the BS13 dormitory's residents and administration.

We have made certain that the front-end, which is what you see and interact with, looks excellent and responds swiftly. It was designed to be dynamic and eye-catching. The back end is powerful and capable. It is the component that ensures everything works smoothly and all features function properly.

We chose a flexible and scalable database solution for storing all the information. It maintains the data secure and well-organized.

These architectural choices are all aimed at making DormEase a system that prioritizes the requirements and convenience of its users, with the goal of improving daily living at the BS13 dormitory.

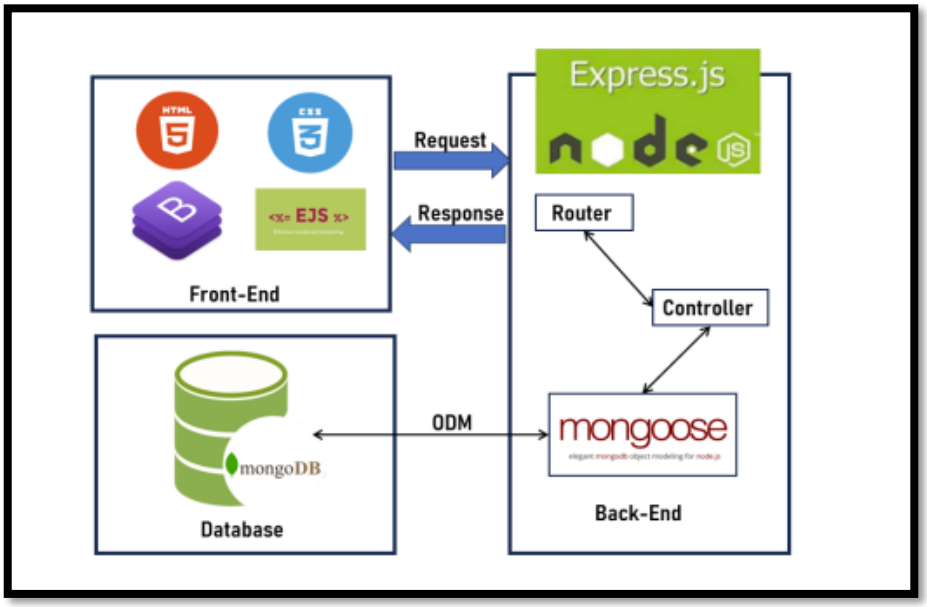


Figure 3.14 Architecture (Source: Author)

In conclusion, DormEase's architecture as shown in the above Figure 3.14, was carefully crafted to prioritize user experience for both residents and administrators at BS13. The front-end offers an engaging, visually appealing

interface, while the robust back end ensures smooth feature functionality. Our flexible, secure database keeps data organized and safe. These architectural choices collectively work towards enhancing daily life at the BS13 dormitory.

3.3 Coding

As we discussed in the previous section 2.3, we followed a simple but important principle in our coding approach: Quality-Driven Development (QDD). The key to breaking free from tutorial dependency is Quality-Driven Development (QDD). It stresses problem-solving and hands-on abilities. With QDD, you find that a solid foundation in programming principles, syntax, and understanding how to write a function definition is all you need for starting building. It all boils down to asking the proper questions and skillfully researching the answers by Googling or asking colleagues for help. This technique promotes hands-on learning and self-reliance, preventing the loop of tutorial dependence.

3.3.1 Backend

Backend server-side development is the backbone of modern web and mobile applications. This critical component is responsible for handling data, logic, and communication, making it an integral part of any software ecosystem.

Key Aspects of Backend Server-Side Development:

- **Programming Languages and Frameworks:** Backend development often involves a variety of programming languages such as Python, JavaScript (Node.js), Ruby, Java, and more. Frameworks like Express.js, Django, Ruby on Rails, and Spring provide pre-built structures and libraries to streamline development.

- **Server Management:** The backend typically runs on a server, whether physical or cloud based. Developers need to set up and manage these

servers to ensure their applications are available and responsive to user requests.

- **Data Handling:** Managing and storing data is a fundamental aspect of the backend. This includes working with databases (SQL, NoSQL), processing user inputs, and ensuring data security.

- **API Development:** Backend developers create APIs (Application Programming Interfaces) that enable communication between different parts of an application or between different applications. These APIs allow for data exchange and integration.

- **Authentication and Authorization:** Implementing secure user authentication and authorization systems is vital. This ensures that only authorized users can access certain parts of an application and that user data remains confidential.

- **Scalability:** Backend developers need to design systems that can handle growing numbers of users and data. Scalability is essential to maintain application performance as it becomes more popular.

- **Data Protection:** Data breaches are a major concern. Backend developers must employ encryption and other security measures to protect sensitive data.
- **Documentation:** Creating comprehensive documentation is crucial for future development, maintenance, and collaboration with other team members.

In conclusion, Backend server-side development involves handling the server and databases, and assuring data integrity and security, allowing communication between various components of an application, and

improving performance. It serves as the foundation for the user interface (front end), making it an essential part of every software project.

3.3.1.1 Nodejs vs Flask

Flask and Node.js represent different approaches to building web applications. Flask is a Python-based web framework that primarily follows a synchronous (or blocking) programming model. On the other hand, Node.js is a runtime environment that uses JavaScript and is known for its asynchronous (or non-blocking) programming model. A comparison study by Challapalli gives us the significant differences between Flask and Nodejs, they compared both frameworks performance using Locust which an open-source software and Autocannon by testing them under similar condition **(Challapalli et al.).**

Flask:

1**. Synchronous:** In Flask, operations are typically carried out sequentially, which means that if one operation is slow (e.g., a database query), it can block the entire application until it's completed.

2. Python: Flask uses Python as its primary programming language, which is known for its simplicity and readability.

3. Simplicity: Flask is praised for its ease of use and minimalism, making it a good choice for small to medium-sized applications.

4. Community: It has a large and active community, providing a wide range of extensions and libraries.

Node.js:

1. **Asynchronous:** Node.js uses an event-driven, non-blocking I/O model, which allows it to handle multiple tasks simultaneously without blocking the application.

2. **JavaScript:** Node.js relies on JavaScript, which is a widely used language, especially in web development.

3. **Scalability:** Its asynchronous nature makes Node.js well-suited for scalable and real-time applications, such as chat applications or streaming services.

4. **Community:** Node.js has a growing community with a wealth of modules available through npm (Node Package Manager)

The choice between Flask and Node.js is mostly determined by your project's specific requirements, your expertise with programming languages, and your desire for synchronous versus asynchronous development. Flask might be a suitable choice if you're developing a simple web application with Python. Node.js may be a better alternative if you're working on a real-time, highly parallel system and have JavaScript experience.

3.3.1.2 Reasons to select Node.js.

We chose Node.js as the primary foundation for DormEase for a variety of reasons. DormEase's backend server-side development provides important services such as database management, data security, effective communication between system components, and performance optimization. Given this essential role, it is critical that you choose an efficient and robust technology.

According to Krause the reason behind the fast implementation of Nodejs is because Nodejs is an Open-Source Platform used for JavaScript server page execution. The core of Node.js is the in C/C++ written and hence fast JavaScript implementation V8, which compiles the JavaScript code before execution in native machine code **(Krause 2017).**

Node.js distinguishes itself by providing an integrated development experience that uses JavaScript for both the frontend and backend. This approach simplifies code sharing, reduces context switching, and improves development efficiency, making it suitable for DormEase's prototype. This choice ensures that developers can collaborate across the entire application.

DormEase's development is supported by the huge Node.js ecosystem, which is managed by npm (Node Package Manager). Developers can take advantage of pre-built features with a variety of open-source modules at their disposal, saving essential time and resources during the DormEase development process.

The built-in features of Node.js are preferably aligned with DormEase's RESTful API delivery design. The platform facilitates the establishment of APIs which allow seamless integration between frontend and backend components. This kind of communication is fundamental to the DormEase user experience.

Furthermore, Node.js enables you the capability of determining the most suitable database system for DormEase, including choices like as MongoDB. This flexibility ensures that the database of choice can support the application's specific data storage and retrieval requirements.

Finally, Node.js appears as a suitable development option for DormEase. Its real-time capabilities, unified technological stack, scalability, huge module ecosystem, and rapid development features are all aligned with the DormEase project's specific demands and objectives, ensuring the project's success.

3.3.1.3 Express

Express.js contributes significantly to the development of the DormEase application by seamlessly complementing Node.js. R. Archer introduces ExpressJS (also known as Express) it as a Node.js web application framework in this research paper. It assists developers with various aspects of web

application development, such as error handling, database connectivity (in particular, with MongoDB), app development via the Express application generator, middleware usage, interacting with template engines, handling processes, dealing with proxies, debugging, and migrating applications from Express version 3 to 4. Express is essentially a framework that streamlines and simplifies the process of developing Node.js web applications **(Archer 2015).**

Here's a more complete explanation of how Express fits the requirements of DormEase:

- **Efficient Routing:** DormEase makes effective use of Express's powerful routing technology to define endpoints and routes. This streamlines HTTP request handling, resulting in an ordered and structured API. Due to the well-defined route structure, users and administrators get an efficient routing experience.
- **Modular Middleware:** Express adds middleware, which allows DormEase to run functions during the request-response cycle. User authentication, logging, data parsing, and other duties are handled by these middleware routines. DormEase uses middleware to maintain modular and code that is reusable, ensuring that all areas of the app run seamlessly.
- **Streamlined HTTP Handling:** Express includes facilities for managing standard HTTP functions including setting headers, managing cookies, and handling session data. This reduces the demand for low-level coding in the creation of DormEase, increasing efficiency and security.
- **RESTful API Development:** Express is an obvious choice for DormEase's RESTful APIs. It makes it easier to define routes for different HTTP methods and to work with JSON data. This is critical for DormEase's front-end and back-end modules to communicate effectively and securely.
- **Structured Error Handling:** Express introduces an organized approach to error handling. This is useful for DormEase as it enables it easier to handle and respond to various types of faults that may arise during the

application's execution. This provides a consistent and seamless user experience.

In summary, Express.js is a critical component of DormEase's development, easing numerous complex web development processes and enabling the development team to focus on developing the application's unique features and logic. The collaboration between Node.js and Express improves the capabilities of the web application, making them an ideal match for DormEase's success.

3.3.1.4 RESTful API

The In the context of the DormEase application, developed using Node.js with Express.js for the backend, the REST API is critical in providing seamless communication and data interchange between system components. Belkhir focuses on how Android apps connect with the internet using REST APIs, that operate as communication rules. It discovered that developers used to mostly utilize "HttpURLConnection," but now prefer tools like "Okhttp" and "Retrofit." However, other critical procedures, such as dealing with internet outages and errors, were often neglected. According to the report, developers should utilize the correct tools and follow certain best practices to improve their apps **(Belkhir et al.).**

Here's an explanation of how the REST API operates in the DormEase context:

- **Data Exchange:** The REST API serves as a vital bridge connecting the DormEase frontend, which users interact with, and the backend, where data processing and management occur. It facilitates the exchange of data, encompassing user profiles, content, and other resources essential for the application's functionality.

- **Request Processing:** When users interact with the DormEase interface, such as filling out forms or requesting information, the frontend sends HTTP calls to the REST API. These requests cover an extensive variety of

functions, from data retrieval to task execution within the application's framework.

- **Backend Coordination:** The REST API takes in and handles incoming HTTP requests using Express.js inside the Node.js environment. It then effectively routes these requests to the appropriate backend URLs and functions. Database access, calculations, and particular application features are examples of backend tasks.

- **Structured Data Exchange:** DormEase's REST API communicates using conventional data formats such as JSON. This ensures that data is organized and accessible by both the frontend and the backend, irrespective of the technology or programming languages used.

- **Response Management:** After the backend has finished processing, it returns a response to the REST API. The REST API then sends this response to the DormEase client side. Depending on the nature of the user's request, the response may provide data, status updates, or error messages.

In conclusion, the REST API is an essential component of DormEase since it provides a consistent, efficient, and structured means of communication. It successfully separates the frontend's duties for user interactions and display from the backends' responsibilities for data and application logic. This separation improves the application's maintainability, scalability, and adaptability, ensuring DormEase's success.

3.3.1.5 Other JavaScript libraries

JOI: Joi is a JavaScript package that is used on the server side for validating request data. It is critical in ensuring that data delivered from the front end, such as user input or API queries, fits the specific DormEase requirements and limitations. Joi aids in the verification of the integrity and validity of incoming

data, lowering the possibility of wrong or malicious data entering the system. DormEase can preserve data consistency, security, and dependability by setting validation criteria using Joi, resulting in a more powerful and reliable experience for both residents and administrators.

```
const validateProduct = (req, res, next) => {
  const productSchema = Joi.object({
    product: Joi.object({
        title: Joi.string().required(),
        price: Joi.number().required().min(0),
        image: Joi.string().required(),
        location: Joi.string().required(),
        description: Joi.string().required()
    }).required()
  })
  console.log(req.body)
  const {error} = productScheama.validate(req.body, camp);
  // console.log("**************************",error,"**********
  if(error){
    const msg = error.deatils.map(el => el.messgae).join(',')
    throw new AppError(msg, 400)

}
  else{
    next();
  }
}
```

Figure 3.15 Joi middleware function to validate creation of new product in marketplace. (Source: Author)

The middleware function in figure 3.15 is used for validating incoming data from a request, particularly for product-related information. Here's a breakdown of what it does:

- It defines a validation schema for product data using Joi. The schema specifies that a valid product should have attributes such as "title," "price," "image," "location," and "description," each with specific validation rules.
- The function logs the request's body (the data sent in the request) to the console.
- It attempts to validate the request's body (req.body) against the defined product schema using Joi's validate method.

- If there is an error in validation, the function generates an error message based on the details of the validation error (e.g., missing required fields or invalid data) and throws an "AppError" with a status code of 400. The error message is constructed by joining the error details.
- If there are no validation errors, the function calls the "next" function, allowing the request to proceed to the next middleware in the Express.js request-response cycle.

To summarize, this middleware function determines whether the new product data given in the request complies with specific validation standards. If the data is correct, the request is processed; otherwise, an error is generated.

Morgan: Morgan is a well-known logging middleware for Node.js applications. It is a necessary tool for developing web apps with frameworks such as Express.js. Morgan's primary goal is to build extensive logs that capture important information about incoming HTTP requests and responses. Users can choose from established log formats or develop bespoke ones that are tailored to their individual requirements. These logs contain information including request methods, URLs, status codes, and response times. Morgan is a useful source for developers looking for insights on the behavior of their web applications because it plays a critical role in improving debugging, monitoring, and performance analysis.

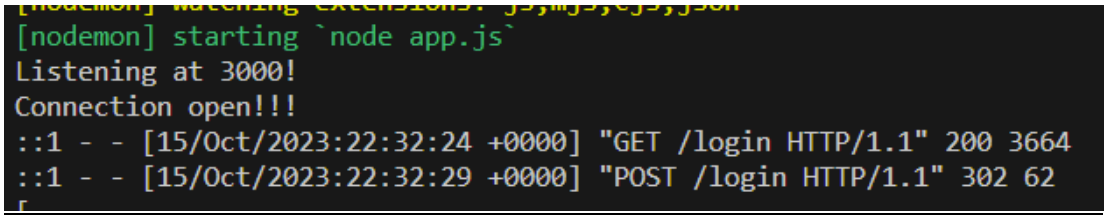


Figure 3.16 Request getting logged and displayed in the terminal. (Source: Author)

In the Figure 3.16, the logging of requests and responses is prominently displayed in the terminal. This feature provided invaluable assistance during

the development of the DormEase application, enabling effective debugging and troubleshooting on various instances.

Connect-Flash: "connect-flash" can be seen as a middleware that integrates with the Express.js web application framework. Its primary role is to handle flash messages within the DormEase application. These flash messages are brief, temporary notifications that are shown to users in response to specific events or actions. For example, when a user successfully logs in entering the right credentials as shown in figure 3.17 or encounters an error while entering the wrong credentials as shown in figure 3.18.

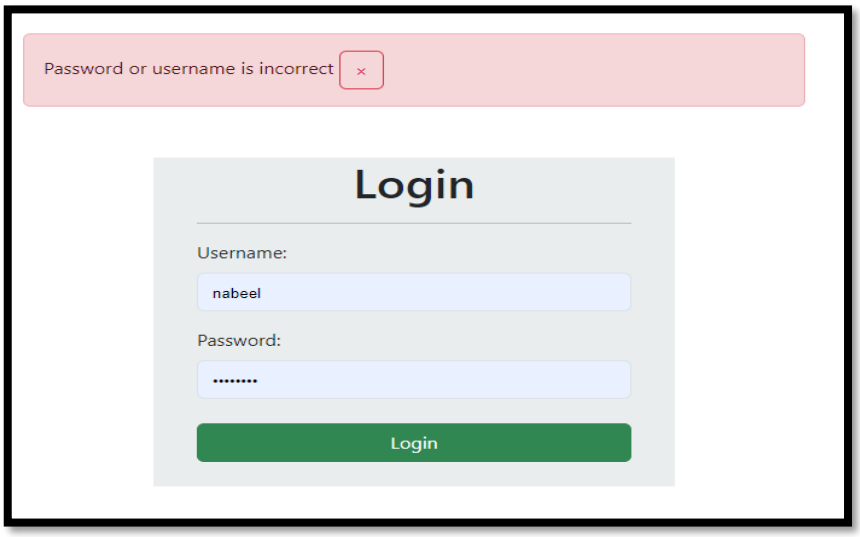


Figure 3.17 Success Flash message (Source: Author)

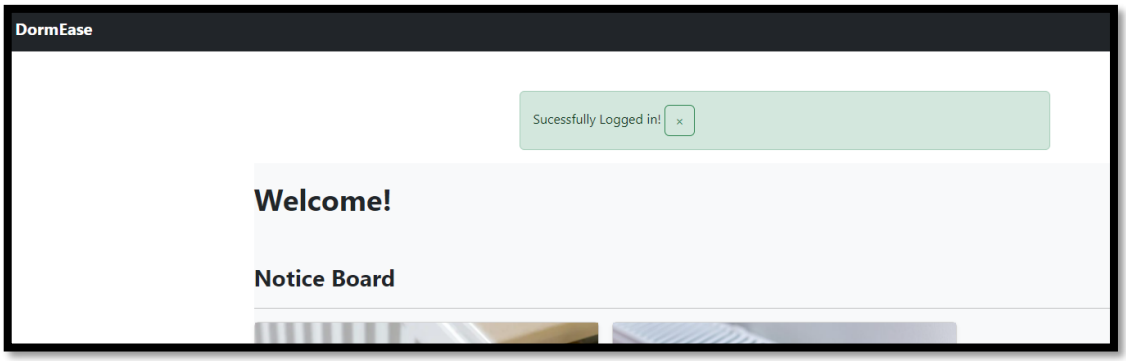


Figure 3.18 Error Flash message (Source: Author)

Passport: According to the author of the paper In DormEase, Passport integration strengthens security and user access control. It supports various authentication methods, including local credentials and social logins. For instance, residents can register with their university email, ensuring secure access. Passport manages user identities, sessions, and feature access, enhancing data security and user privacy for a hassle-free login experience. This code snippet in the below figure 3.19 configures and sets up the Passport.js authentication middleware in an Express.js application. It includes the following key steps:

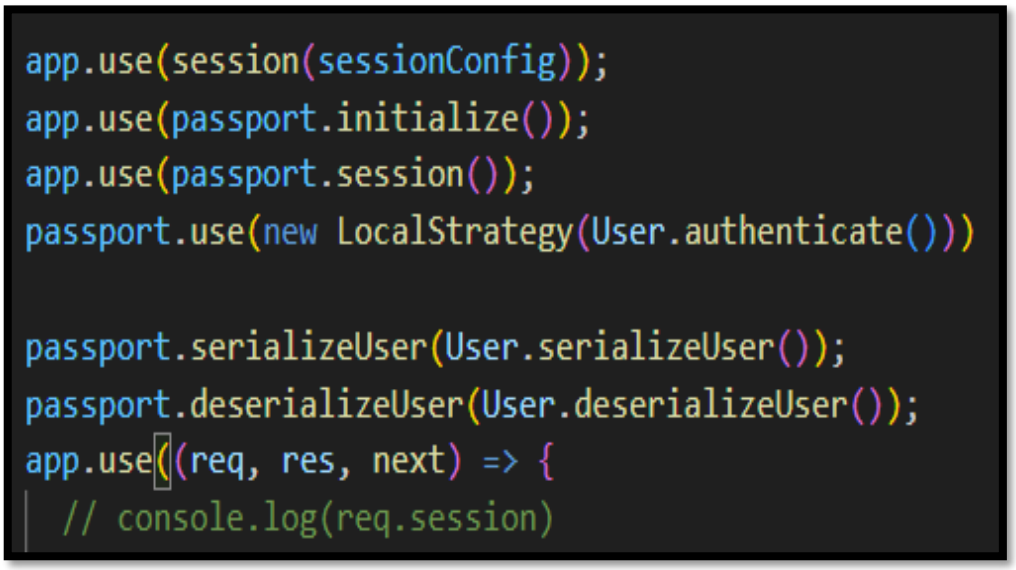


Figure 3.19 Passport Authentication middleware (Source: Author)

- **app.use(session(sessionConfig)):** This line sets up session handling using the provided sessionConfig. Sessions are essential for storing user authentication data and maintaining user state across requests.
- **app.use(passport.initialize()):** This line initializes Passport, preparing it for authentication.
- **app.use(passport.session()):** This line sets up Passport to use sessions to persist login state between requests.
- **passport.use(new LocalStrategy(User.authenticate())):** Here, a LocalStrategy is created for handling username and password-based

authentication, typically against a local user database (e.g., in the User model).

- **passport.serializeUser(User.serializeUser()):** This line defines how user data should be serialized to be stored in the session. It's necessary for maintaining user sessions.
- **passport.deserializeUser(User.deserializeUser()):** This line specifies how to deserialize user data from the session.

Together, these configurations set up Passport.js to handle user authentication and session management within the Express application. Passport's LocalStrategy is used, which is common for username and password-based authentication against a local database.

3.3.2 Front End

According to **Hensley (2022)'s** blog, front-end development is critical for combining website design with back-end functionality. It enhances the user experience by combining design and data. A well-executed front-end promotes user-friendliness and intuitiveness, increasing the accessibility and user-centeredness of websites.

The DormEase front end serves as the primary interface for the user, providing a straightforward and visually appealing platform for simple interactions. It was designed to fulfill the different demands of both residents and administrators at the BS13 dormitory, and it facilitates tasks like laundry booking, maintenance requests, and marketplace activities. It provides residents with seamless access to services and notifications, while administrators may manage communication and service requests more efficiently.

In this section, we will look at the technologies applied to develop the DormEase front end, as well as the logic behind choosing these technologies to

design an interface that prioritizes customer satisfaction and the particular needs of BS13 dormitory residents.

3.3.2.1 HTML5

Hypertext Markup Language 5 is the latest iteration of the HTML standard, which is used for structuring and delivering web content. It adds various new elements, properties, and capabilities to web pages that improve their semantic structure and functionality. HTML5 is crucial in establishing a well-organized and structured user experience in DormEase.

Form Elements: HTML5 has improved form controls and input types. DormEase takes use of this by adding HTML5 forms, which contain capabilities such as input validation, date pickers, and more, to improve user interactions, such as the "Laundry Booking" and "Maintenance Request" functionalities.

Multimedia Support: HTML5 includes built-in functionality for audio and video elements. These are used in DormEase to integrate multimedia content, such as tutorial videos or informative clips.

Canvas: The < canvas> element is used to draw graphics and generate visual elements dynamically. While it is not commonly utilized in DormEase's front end, it may be useful for future expansion or interactive aspects.

In summary, HTML5 is a fundamental technology that improves the structure and functionality of DormEase's front end. It not only provides semantic meaning to the text, but it also includes capabilities for increased user interactions, multimedia support, and the ability for dynamic graphics. This offers a modern and effective user interface adapted to the demands of BS13 dormitory residents.

3.3.2.2 CSS3

Cascading Style Sheets 3 is the most recent version of the CSS standard, which is used in web development to govern the appearance and styling of HTML elements. CSS3 is used extensively in DormEase to create a visually appealing and user-friendly design.

- **Advanced Styling:** CSS3 introduces a plethora of advanced styling features, including gradients, drop shadows, rounded corners, and custom fonts. These features are used in DormEase to enhance the visual aesthetics and provide a modern look and feel to the application.
- **Responsive Design:** CSS3 enables the implementation of responsive design principles, making DormEase accessible and functional on a variety of devices and screen sizes. Media queries are employed to adapt the layout and styling to different resolutions, ensuring an optimal user experience.
- **Transitions and Animations:** CSS3 supports transitions and animations, allowing for smoother and more interactive user interfaces. These effects can be found in DormEase to provide users with engaging interactions, like animated notifications or transitions between pages.
- **Flexbox and Grid Layout:** CSS3 introduces the flexible box layout (Flexbox) and the grid layout system, which are used in DormEase to create dynamic and responsive page structures. These features help in maintaining consistency and alignment of elements across the application.

The features of CSS3 are essential to DormEase since they guarantee that the application is not only useful but also visually appealing and adjustable to different screen sizes and devices. For the residents of the BS13 dorm, this technology is essential to developing a visually appealing and user-focused front end.

3.3.2.3 Bootstrap

According to Andrew Zola, the open-source framework Bootstrap offers pre-defined grids, CSS classes, and JavaScript components to simplify the process of developing responsive websites. As a result, developing specific code for common design components and functionality takes less time and effort for developers when creating websites. It's an accepted choice in web development because of its extremely accessible features and engaged community **(Zola 2023).** Bootstrap offers the following features:

- **Aesthetic Enhancement:** DormEase's visual aesthetics have been greatly enhanced with Bootstrap's pre-designed CSS components and classes. The BS13 dorm community responds well to the application's new, visually appealing, and modern interface. The application is now visually improved by using Bootstrap's default stylistic selections for typography, colours, and spacing.
- **Responsive Design:** DormEase adjusts effortlessly to a range of screen sizes and devices because of Bootstrap's responsive grid framework. Whether a user accesses the program on a desktop, tablet, or mobile device, responsiveness is essential to provide them a consistent and easy-to-use experience.
- **Form Validation:** Form elements are subject to the "validate-form" class, which is usually given to the form itself or to certain input fields on the form. It acts as a hook for the client-side JavaScript code that manages form validation. This class is used to initiate validation tests to make sure the data provided is correct and satisfies the necessary requirements when users submit data through a form.
- **Input Validation:** Every input field in the form may have its own validation criteria, which the "validate-form" class helps to streamline and organize. For instance, in Figure 3.20 below, the Title and location fields are filled in in accordance with the stated validation; hence, they are

highlighted in green. In contrast, the picture URL and price fields are left empty and are highlighted in red, despite the validations being set.

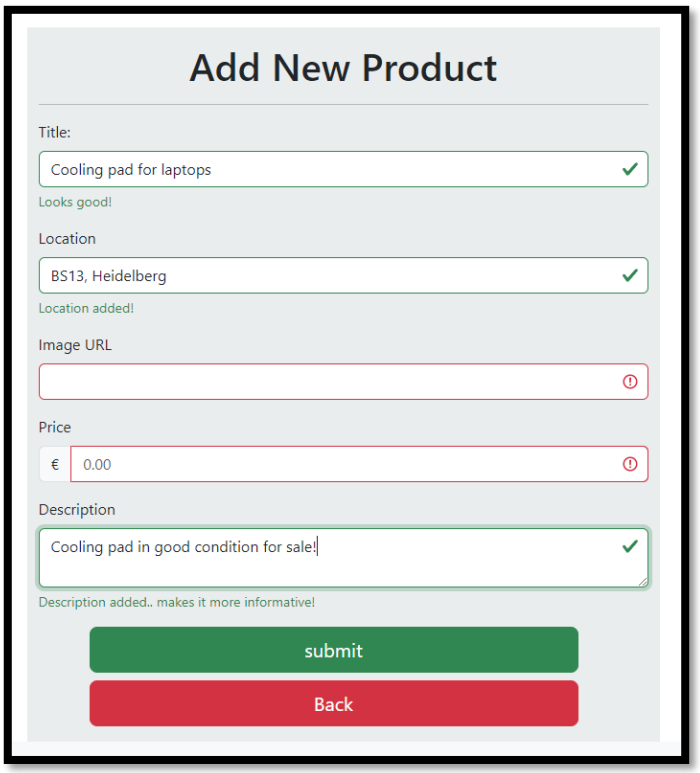


Figure 3.20 Front end form validation (Source: Author)

By offering a foundation for adaptable and aesthetically beautiful design, Bootstrap has improved the visual appeal of DormEase. Users and administrators will find the interface to be user-friendly due to its numerous stylistic options and ease of usage.

3.3.2.4 EJS

EJS was essential to the front-end development of DormEase because it allowed JavaScript to be easily integrated into HTML templates for dynamic content, as seen in Figure 3.21 below. This made it possible for the application to effectively show tailored data and react to user inputs. EJS made it easier for data to move in real time between the frontend and backend, enabling the delivery of up-to-date content and guaranteeing a standardized user interface. The user experience was improved by its conditional rendering and data

iteration features, which improved DormEase's frontend's responsiveness and interaction.

```
    <hr>
    <% for(let product of products){ %>
        <div class="card mb-3" >
            <div class="row">
                <div class="col-md-4">
                    <img style="width: 500px; height: 300px;" class="img-fluid" src="<%=product.image%>">
                </div>
                <div class="col-md-8">
                    <div class="card-body">
                        <h5 class="card-title"><%= product.title %></h5>
                        <p class="card-text"><%= product.description %></p>

                        <p class="card-text">
                            <small class="text-muted"><%= product.location %></small>
                        </p>
                        <a class="btn btn-primary" href="/products/<%=product._id%>">View</a>
                    </div>
                </div>
            </div>
        </div>
    <% } %>

</div>
```

Figure 3.21 Ejs embedded in a HTML file. (Source: Author)

The Blog by Victor Mgbemena, in guide to build dynamic web application using EJS which played a key role in giving DormEase's web pages a consistent and visually appealing style. It made it easier to render elements like footers, forms, navigation bars, and cards while maintaining a sleek appearance. The application's frontend was simple to manage and update because to the uniform structure that EJS templates **(Mgbemena 2023).**

3.3.3 Databases

3.3.3.1 SQL vs NOSQL

There are two distinct techniques for managing and storing data: NoSQL and SQL databases. Each has advantages and applications of its own.

Wisal Khan conducted a thorough literature study to examine the significance of software architecture in large-scale data processing using SQL and NoSQL databases. NoSQL databases are preferred for their scalability with unstructured data, while SQL databases are known for their structured data

and horizontal expansion. Due to the various approaches, the assessment addresses issues with data flow between various cloud platforms. It also emphasizes the applicability of SQL databases for online transaction processing (OLTP) and the potential of NoSQL databases for huge data analytics **(Khan et al. 2023).**

Relational databases with structured query language (SQL) for data definition and management include MySQL and PostgreSQL. They are perfect for applications like financial systems and e-commerce platforms that demand complex inquiries and transactions. High consistency and integrity are provided by SQL databases, ensuring reliable and correct data. They are ideal for applications requiring data consistency since they also support ACID (Atomicity, Consistency, Isolation, and Durability) transactions.

On the other hand, non-relational NoSQL databases, like MongoDB and Cassandra, are designed to handle unstructured or semi-structured data. They are ideal for applications with changing data requirements, such social media platforms, or Internet of Things systems, because of their great scalability and flexibility in data modeling. NoSQL databases can manage enormous volumes of data and traffic because they place an emphasis on horizontal scalability and dispersion. For high availability and partition tolerance, they could sacrifice some data consistency, which is acceptable in some use situations.

Reason to choose NOSQL:

For DormEase, we decided on a NoSQL database since it best meets our needs. NoSQL databases, like the ones found in DormEase, are perfect for handling a variety of data types.

Choosing to utilize a NoSQL database like MongoDB is mostly influenced by its horizontal scalability. As users' unique data volumes increase, DormEase can easily manage them, making it simple to scale the service as needed.

It is crucial to remember that the flexibility offered by NoSQL while creating the database schema can have drawbacks. Although it provides freedom, it also implies that we must take more care when creating an effective design. As opposed to this, SQL databases enforce a set schema and are more structured, which can assist avoid inefficiencies. For DormEase, NoSQL is the best option because of its benefits and our ability to carefully customize the database to meet our unique requirements.

3.3.3.2 MongoDB

The MongoDB, the NoSQL database chosen for DormEase, is leading edge and adaptable, satisfying the needs of our application. Unlike traditional relational databases, MongoDB stores data in a binary representation of documents called BSON that resembles JSON. The dynamic and ever-evolving characteristics of DormEase's data make this framework perfect for it.

Anjali Chauhan's study describes how MongoDB, a well-known NoSQL database, is more reliable and efficient than alternative solutions because of its open-source nature and shared-nothing cluster architecture, it is very effective in constructing data warehouses. Despite being well-supported and documented, its regular modifications necessitate a risk-taking attitude. Schema-free, distributed, horizontally scalable solutions with simple APIs and easy replication are available with next-generation NoSQL databases. Subsequent investigations may examine data access performance, particularly in situations with data hotspots, possibly through the integration of an additional model within MongoDB **(Chauhan 2019).**

We can efficiently store and retrieve data thanks to MongoDB's document-based architecture. This implies that we can handle a variety of data sorts (such user profiles, notifications, and marketplace product listings) quickly and easily within the DormEase framework. Furthermore, because of its horizontal scalability, MongoDB is an excellent option for handling any potential expansion of DormEase's user base in the future.

The useful tools and resources offered by MongoDB improve the development process. MongoDB's graphical user interface (GUI), Mongo Compass, may make database management easier for DormEase developers. We can efficiently do querying, data visualization, and schema analysis with Mongo Compass while maintaining data consistency and integrity. We can also access the database directly through the terminal by using the Mongo shell, which is MongoDB's command-line interface.

3.3.3.3 Mongoose

According to Bin Uzayr, MongoDB is a well-known cross-platform document database that is occasionally paired with other nonrelational data stores in the "NoSQL" categorization, like CouchDB, Cassandra, RavenDB, and others. Because its "records" are kept as simple JSON objects and because its query interface and stored functions are written in plain JavaScript, it is a well-liked option for data storage among Node.js developers **(bin Uzayr et al. 2019).**

Mongoose was selected for the DormEase development in addition to MongoDB because it provides an easy-to-use interface for communicating with databases. Using Mongoose simplifies the processes of data modeling and interaction. Data schemas are specifications that serve as rules for how information should be arranged in databases. This functionality helped to ensure that data consistency and integrity were maintained throughout the development process.

```
models > JS product.js > ...
1   const mongoose = require('mongoose');
2   const Schema = mongoose.Schema;
3
4   const ProductSchema = new Schema({
5       title: String,
6       image: String,
7       price: Number,
8       description: String,
9       location: String,
10      author: {
11          type: Schema.Types.ObjectId,
12          ref: 'User'
13      },
14      comments: [
15          {
16              type: Schema.Types.ObjectId,
17              ref: 'Comment'
18          }
19      ]
20  });
21
22  module.exports = mongoose.model('Product', ProductSchema);
```

Figure 3.22 Defines a Mongoose schema for products in DormEase. (Source: Author)

For features like products and keeping track of the comments for each product in DormEase, Mongoose also provided the ability to create complex queries and handle the linkages between data. By streamlining database operations and facilitating more efficient development, Mongoose completely fulfilled the project's requirements.

The development of DormEase benefited significantly from the combination of Mongoose and MongoDB. In Figure 3.22, the structured data allowed us to construct schemas, relationships, and data restrictions, which made MongoDB behave more like a SQL database. This was especially beneficial since, despite NOSQL, it combined some of the features of structured data with SQL databases, including scalability and flexibility.

3.4 Error Handling

John Kim's 2017 in his blog explains the importance of error handling and how error handling can be implemented using try-catch block. He further adds that Software development requires error handling to ensure that applications can gracefully handle unexpected problems. It's necessary to identify errors and take appropriate action to prevent program crashes or unwanted behavior **(John Kim 2017)**.

```
app.get('/userhome',isLoggedIn, async(req, res, next)=>{
  try{
    const notices = await Notification.find({});
    console.log(notices);
    res.render('users/home.ejs', {notices})
  }
  catch(e){
    next(e)
  }
})

// error handler FINAL hope
app.use((err, req, res, next)=>{
  console.log("******IN FINAL*****")
  const {status=500} = err;
  if(!err.message) err.message = "Oh No, Something went Wrong!";
  console.log(err, status, err.message)
  // res.status(status).send(message)
  res.status(status).render('products/error.ejs', {err})
})

```

Figure 3.23 Try-catch block and error handling middleware. (Source: Author)

As seen in Figure 3.23 below, try-catch blocks, error messages, and status codes are examples of error handling strategies that are used to manage exceptions and convey issues to users or developers. Appropriate error management enhances both program security and user experience.

3.5 Summary

In this section, we illustrated the DormEase development process along with an in-depth analysis of the crucial Design stages. To understand the needs and experiences of the BS13 community, we first had to go through the Empathize stage. Next came the Define phase, in which we described exactly what problems DormEase would fix, focusing on big problems like scheduling laundry machines, creating repair requests, and centralizing alerts. Creative solutions were listed by the creative process of ideation. Prototyping transformed these concepts into practical representations, with both low and high-fidelity prototypes providing useful insights.

We gave the readers an understanding of the workflow of the prototype and the architecture designed to implement the technical specifications. We

discussed more about the technical aspects of the prototype such as selection tech stack for front-end, back-end and the database, by providing justification for picking that tech stack. We also listed the other libraries which played a crucial part in the development of the prototype.

Finally, we discuss the importance of error handling and with an example showed how error handling using a try-catch block has been implemented in the prototype development code.

4. Prototype and Functionality

4.1. Prototype execution

4.1.1 User Module

4.1.1.1 Notice Board

As shown in the below Figure 4.1 This feature keeps the user informed about various administrative developments.

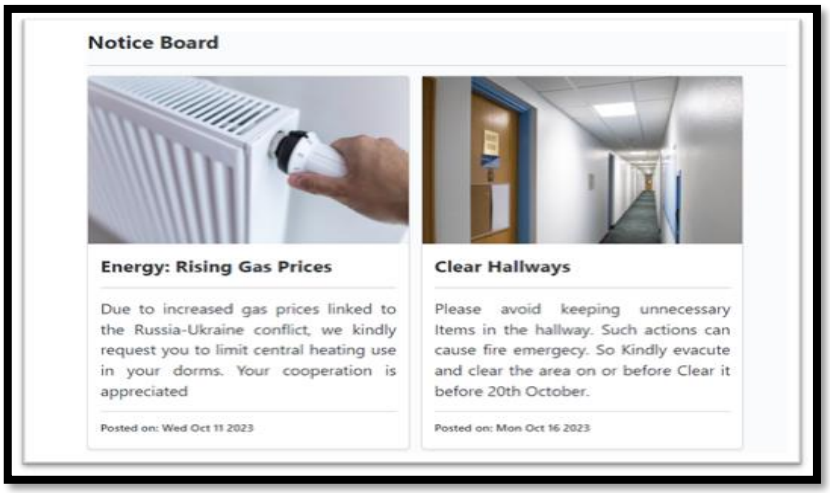


Figure 4.1 User notice board (Source: Author)

4.1.1.2 Functionality Board

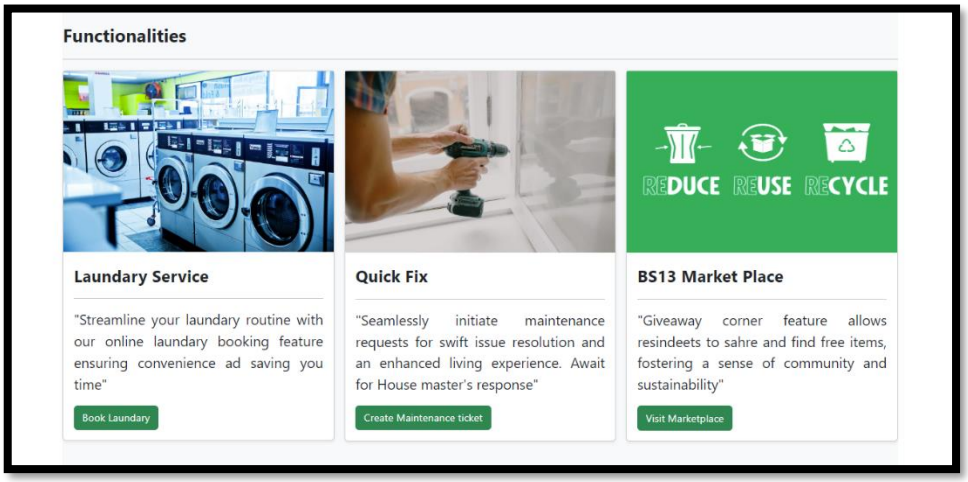


Figure 4.2 User functionality Board (Source: Author)

In the above figure 4.2 is a screenshot of the suer functionality Board the user can visit any feature depending on these needs from doing Laundry, raise maintenance request or visit marketplace.

4.1.1.3 Maintenance tickets

The user can create a ticket to raise a service request in Figure 4.3 below.

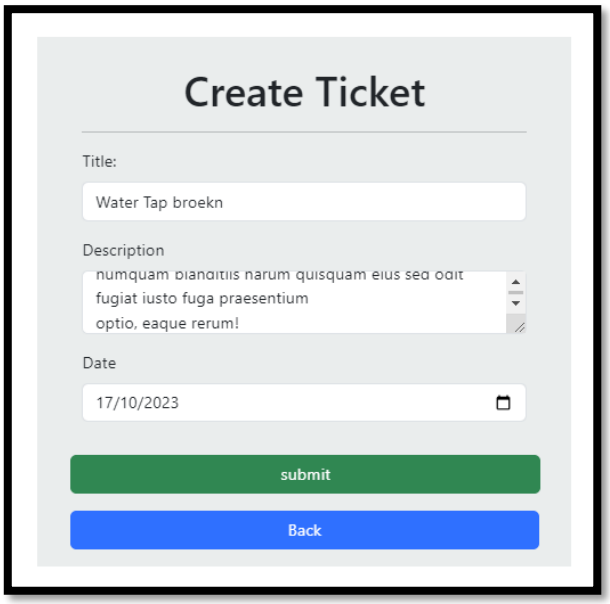


Figure 4.3 Create Notification (Source: Author)

As shown in the Figure 4.4 The user can view the tickets which were fixed by the admin and waiting for your response such tickets end up in author action. If satisfied close the ticket by clicking on “close”, if not satisfied with the service re-open the ticket by clicking on “open.”

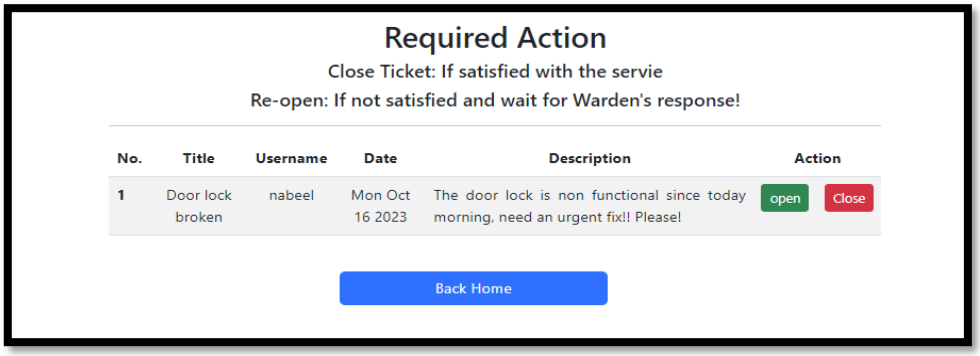


Figure 4.4 Tickets with pending Author action (Source: Author)

As per Figure 4.5 The user can view the history of previously created tickets, and which were fixed by the admin and closed by the user himself.

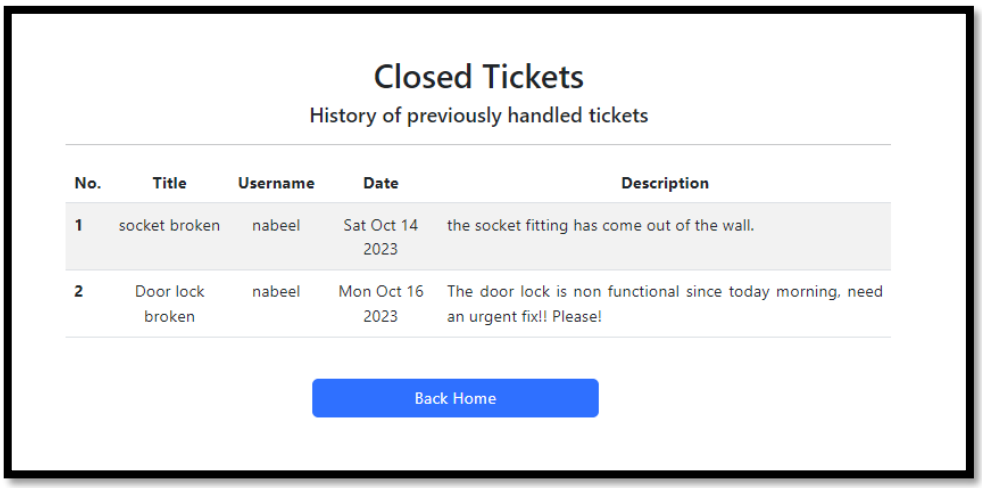


Figure 4.5 Ticket History (Source: Author)

4.1.1.4 Blogs

As shown in the figure 4.6 the user can read different blogs posted by the BS13 community to be informed on various diverse and important topics such as saving yourself during fire emergency, Waste segregation and much more.

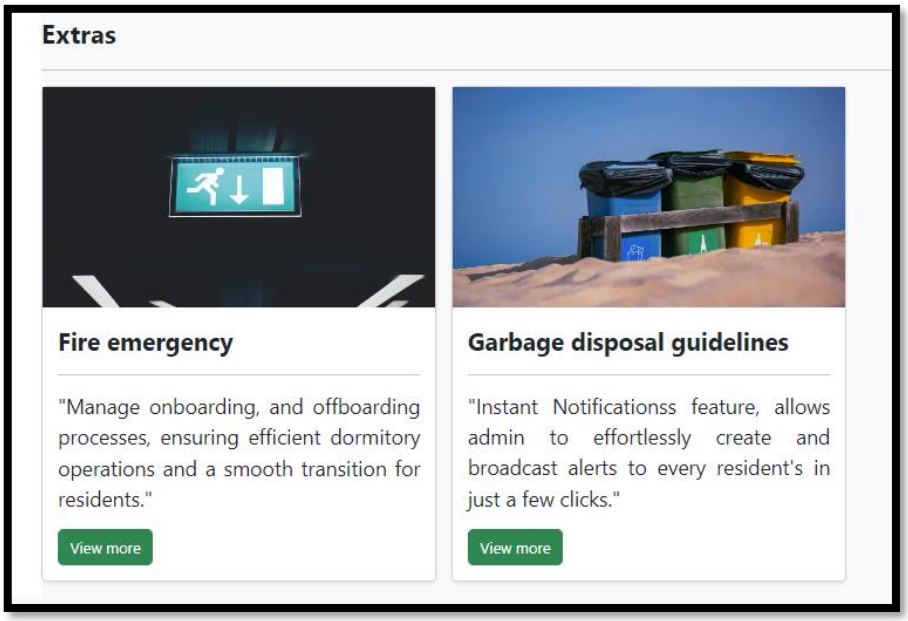


Figure 4.6 Blogs (Source: Author)

4.1.1.5 Marketplace

Users can both add new products and browse the complete product list. They can also switch to a detailed view to study extra details and exhibit interest by providing comments. As shown in Figure 4.7, you can zoom in on a specific product that is for sale.

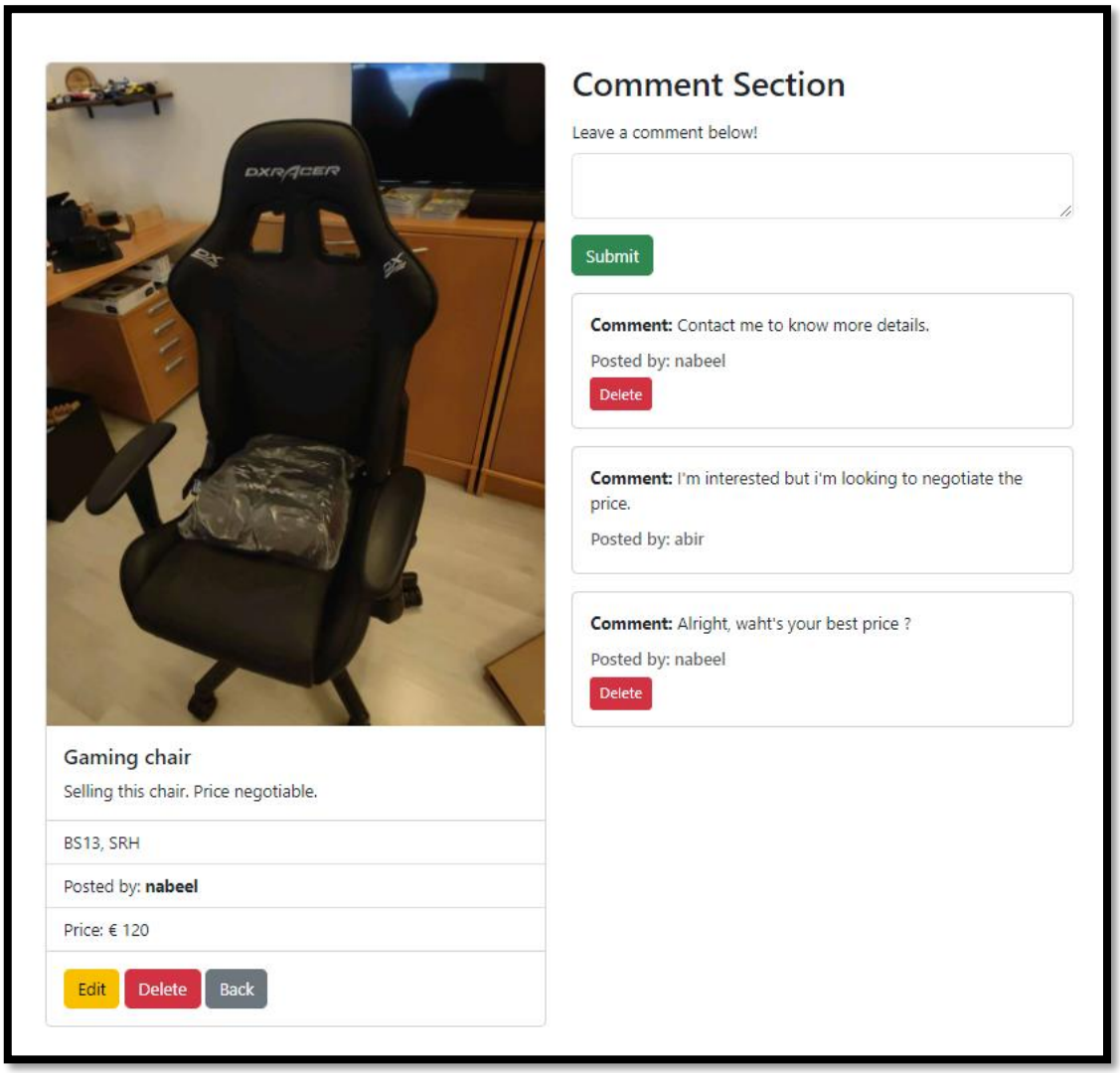


Figure 4.7 Detailed view of a product with comment section (Source: Author)

4.1.1.6 Laundry Booking

Figure 4.8 Laundry Booking Page (Source: Author)

In the above figure 4.8, The user is directed to the laundry booking page where the user can select the date to book his preferred laundry slot. The user can select a slot according to his preferred timing. The page also shows booked slots by the other users which are marked in red, the available slots are marked in green.

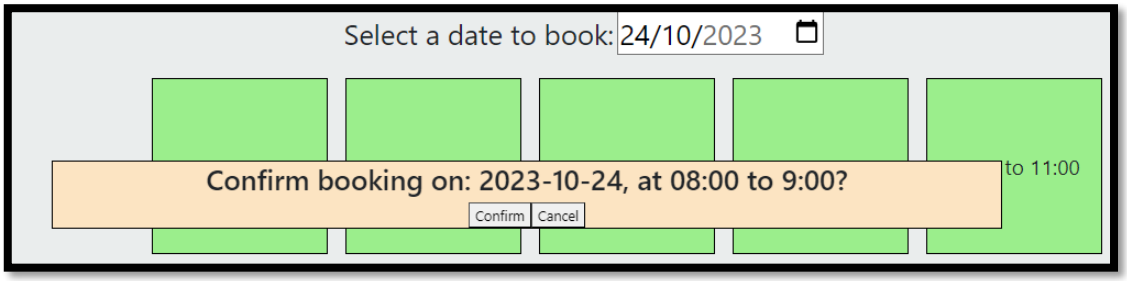


Figure 4.9 Laundry Confirmation Dialogue Box (Source: Author)

Referring to figure 4.9, Once after selecting the preferred date the user can finally confirm the slot by clicking confirm in the dialogue box and check the booking details before confirming the slot.

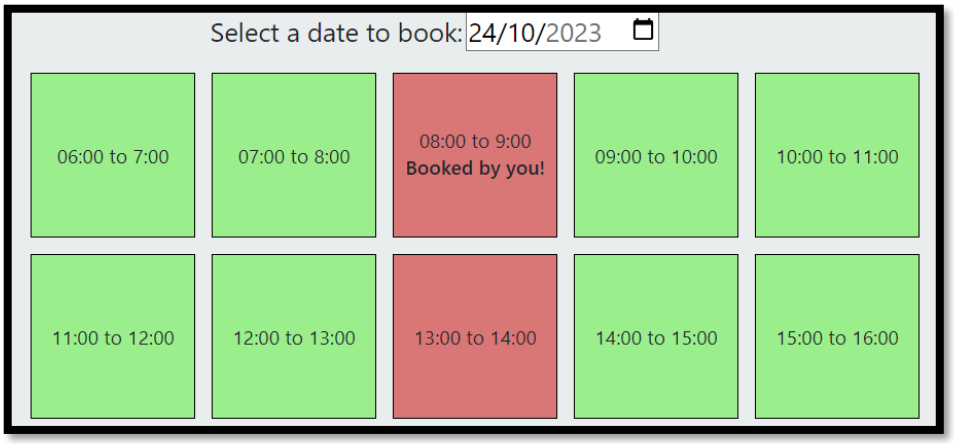


Figure 4.10 Booking Confirmed (Source: Author)

In the above figure 4.10, the user can see the slots that are confirmed and booked. The slots which are booked by the current user are highlighted in red with caption “Booked by you”. The slots which are booked by other users are simply highlighted in red.

4.1.2 Admin module

4.1.2.1 Admin Panel

The admin when logs in is welcomed by the admin panel where they can pick between different features to perform the admin tasks, referring to the below Figure 4.11.

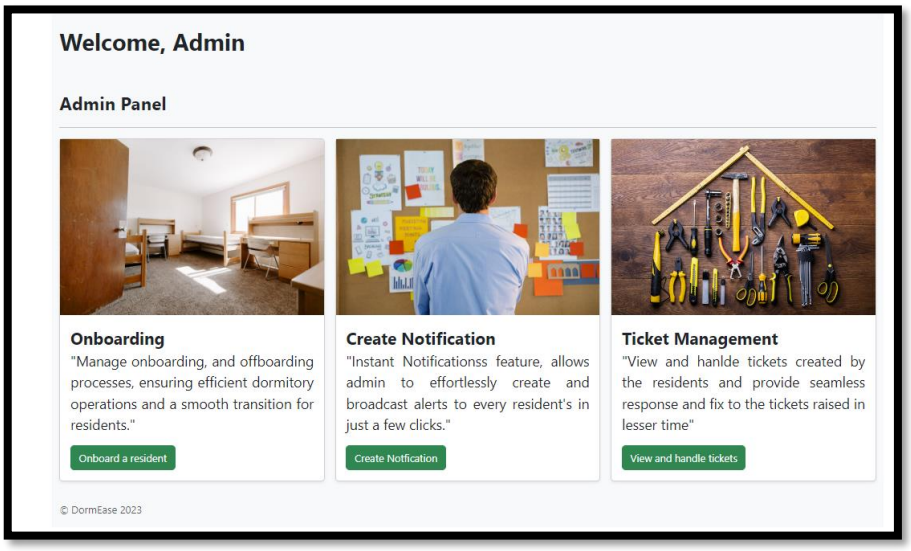


Figure 4.11 Admin panel (Source: Author)

4.1.2.2 On-boarding

The admin can onboard the new residents by filling in the register form to register the student as the new resident of BS13 dorm, by filling in the detailed registration form.

4.1.2.3 Manage maintenance tickets.

Referring to section 4.1.1.3, The admin can handle ethe tickets from his end and after fixing and providing the service the admin can push to ticket to author action section and wait for user's response, as shown in the Figure 4.12.

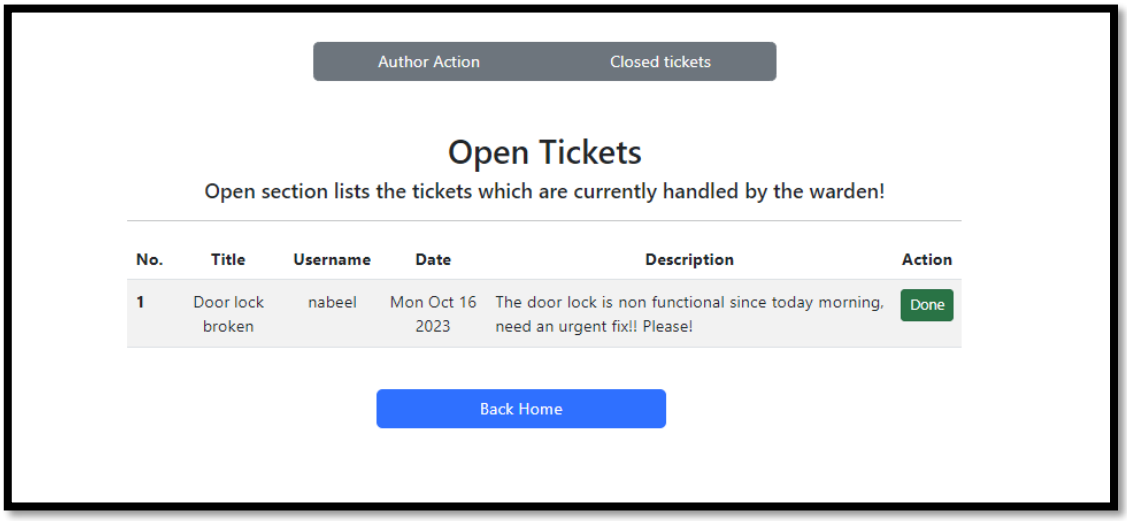


Figure 4.12 Tickets created by the users. (Source: Author)

4.1.4.4 Broadcast Notifications

This feature enables the admin to create new notification updates and broadcast them to all the users in just a few clicks. First the admin must fill in a detailed form to create a new notification. The user can also view the list of notifications previously created and delete them if the notification has gone out of date, as shown in the below Figure 4.13.

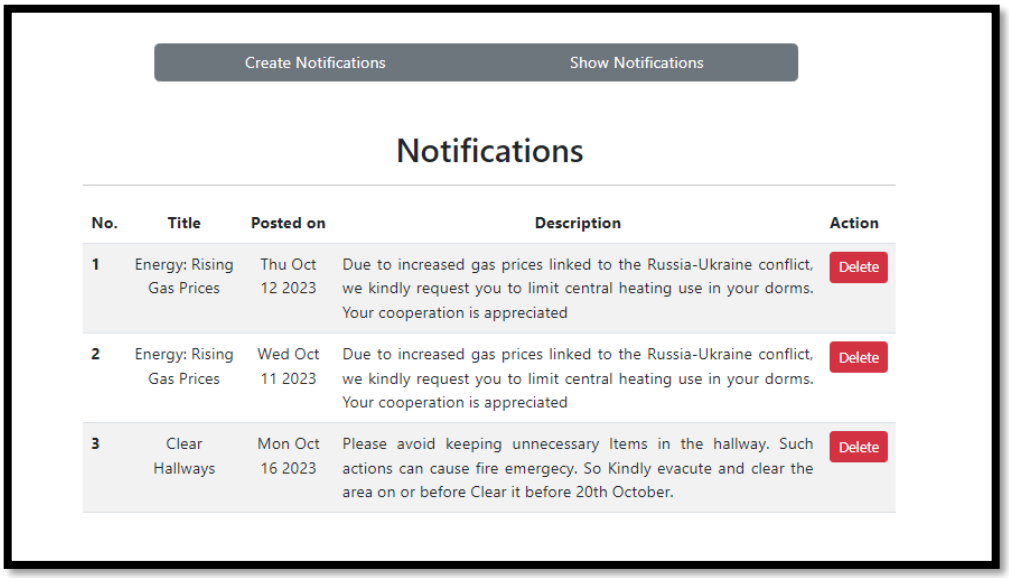


Figure 4.13 All notifications listed at admin side. (Source: Author)

4.2 Critical thinking (Pros and cons)

After the prototype was created, it was critical to evaluate its strengths and limitations to identify areas where the application might be enhanced for better efficiency. Since the development process can always be improved, this assessment was essential.

Pros of DormEase Applications are as follows:

1. **Convenience:** DormEase provides residents with a convenient platform to manage various aspects of dormitory life, including laundry booking, maintenance requests, and communication.
2. **Efficiency:** The system streamlines administrative tasks, such as onboarding new residents and managing maintenance tickets, making it more efficient for administrators.
3. **Real-time Updates:** Users receive real-time notifications, keeping them informed about the latest news and updates within the dormitory.
4. **User-Friendly:** The user interface is designed to be visually appealing and easy to navigate, enhancing the overall user experience.

5. **Data Validation:** DormEase employs data validation and error handling, reducing the risk of incorrect or malicious data entering the system.
6. **Responsive Design:** The system is responsive, ensuring it works well on various devices and screen sizes.

Cons of DormEase Applications are as follows:

1. **Initial Learning Curve:** Users and administrators may need some time to learn and adapt to the system, especially if they are not tech-savvy.
2. **Limited User Base:** The system is designed specifically for a single dormitory (BS13), limiting its usability to residents of that dorm.
3. **Maintenance:** Like any software system, DormEase requires ongoing maintenance to ensure it runs smoothly and securely.
4. **Security Concerns:** While the system enforces security measures, there is always a risk of security breaches in any online platform.
5. **Reliance on Technology:** In the event of technical issues or downtime, users and administrators may face disruptions in their daily activities.
6. **Potential Bugs:** Like any software, DormEase may have occasional bugs or issues that need to be addressed through updates and maintenance.
7. **Privacy Concerns:** Users may have concerns about the privacy and security of their personal data stored in the system.

4.3 Summary

This section explained how the features included in the prototype DormEase application functioned. We provided a detailed explanation of the prototype's real-world functionality with the help of screenshots. Since it is essential for identifying areas that need improvement, we also examined its advantages and disadvantages. DormEase's benefits are readily apparent in their simplicity, as it offers users an accessible platform to handle activities related to living in dorms. By automating administrative tasks and providing real-time information, it enhances the overall experience.

The drawbacks of DormEase include a learning curve, a small user base, maintenance requirements, and the potential for security issues. Bugs or technical issues could make using the app unpleasant, and the protection of personal data could raise privacy issues.

In conclusion, DormEase offers a lot of benefits for managing dorms, but it also has drawbacks that need to be fixed, emphasizing the need for ongoing development and user support.

5. Summary and Future Scope

5.1 Future Scope

In this section we will look at the endless possibilities of making the application more user-centric by providing more useful features and by improving the areas which require attention, as there is always room for improvement.

1. **Expansion to Other Dormitories:** To increase its user base and improve dorm management, DormEase can extend its services to additional dorms within the university or to other universities.
2. **Integration with University Systems:** For academic and administrative reasons, such as course registration or fee payments, integration with university systems is necessary to establish a smooth campus environment.
3. **Smart Laundry:** Making the Laundry systems completely automated by integrating smart devices to the current laundry machines using IOT technology. Hence automating the whole laundry process by eliminating the involvement of admin.
4. **Smart Building Integration:** Utilizing smart building technologies to automate security, energy conservation, and temperature management results in a more technologically sophisticated and ecologically conscious living environment for residents.
5. **AI-Powered Assistance:** Using AI to assist support, such as responding to frequently asked questions, scheduling maintenance, or making suggestions for events and activities.
6. **Visual analytics board:** We can provide the admin with a visual board representing the resident's data from current number of residents residing, time eft on the contract, new on boarding, tickets raised and much more. This will help the admin to view the data and improve in the areas which need attention.

5.2 Summary

Finally, to summarize, with the help of all the research, development of the prototype and feedback from the user's considered and conducted in our master's thesis we answer the research questions which were framed in section.

Research Question 1: How to digitalize the booking of laundry system to reduce the time and effort required by the residents using current conventional methods?

We developed a digital platform that enables residents to schedule washers remotely, digitizing the process and eliminating the need for in-person visits for coin transactions. The entire process has been significantly streamlined by this solution as the residents don't have to wait in lines during the busy hours waiting for their turn. Hence, saving time and effort.

Research Question 2: What technical stack and architecture would be the right pick? How to make the frontend more user-centric and user interactive

For the front end, we chose a tech stack that included HTML, CSS, Bootstrap, JavaScript, and Node.js with Express.js for the back end. Front-end and back-end are intended to be connected through restful delivery. Data was stored and processed using Mongo DB and Mongoose. The selection gave the user-centric design a solid foundation. To make it more user interaction, the architecture concentrated on developing a user interface that is simple to user and user-friendly.

Research Question 3: What is the most optimal and efficient solution to digitalize the maintenance ticket handling?

We have set up a digital maintenance ticket system which enables residents to report issues without requiring in-person visits or emails. The rapid and efficient handling of maintenance requests is ensured by this system. Using

this system, the user can also share feedback on whether he is satisfied with the service.

Research Question 4: How can a digital solution promote sustainability and resource sharing by facilitating the donation of furniture and equipment by residents moving out of the dormitory?

To encourage resource sharing and sustainability, we included a feature that lets residents give, buy, or sell their furniture and equipment to other residents in the dorm. The resident can practice sustainability by reusing the products which are no longer needed by the other residents.

Research Question 5: Which approaches can be utilized to establish a unified notification system and messages, with the aim of improving communication between dormitory residents and admin?

To improve communication between residents and the house master we set up a centralized notification system that gathers and shows alerts, messages, and notifications from various sources. The admin-created notifications are shown instantly on each user's home page, doing away with the drab email notification design that is currently in use.

Research Question 6: How do user-centered design principles align with the development of an IT solution for enhancing the BS13 dormitory experience?

We clearly understood the needs of the residents, identifying their problems, coming up with creative proposals for solutions, and testing the designs for user feedback, we implemented the principles of user-centered design. This strategy made sure the prototype was very user focused.

Research Question 7: What challenges and considerations should be addressed in the development and deployment of the web application's technical stack, software architecture, and database for optimal performance?

Challenges included choosing the right technical stack, considering performance, and ensuring scalability. To achieve the performance requirements, we carefully considered the technologies, optimized the architecture, and selected MongoDB as the suitable database.

Research Question 8: What are the best practices for ensuring the success and usability of the proposed IT solution for BS13 dormitory residents?

We focused on automation, real-time information, and simplicity to ensure success and effectiveness. We realized that to continually improve and optimize the solution for the benefit of the residents, it was necessary to maintain development and user support.

In conclusion, a lot of thought went into each research in the first place, and the end product is the development of DormEase, an IT solution that significantly improves the standard of living for BS13 students living in dorms. The method offers occupants of dorms solutions to several problems while streamlining daily operations, encouraging sustainability, and creating a sense of community. Our goals have been achieved by combining modern architecture, efficient technology, and dedication to the satisfaction of the residents. In addition to its advantages, DormEase has the potential to expand further and improve living in dorms in the future.

Table of Figures

Publication bibliography

Al-Samarraie, Hosam; Hurmuzan, Shuhaila (2018): A review of brainstorming techniques in higher education. In *Thinking Skills and Creativity* 27, pp. 78–91. DOI: 10.1016/j.tsc.2017.12.002.

Ames, A. L. (2001): Users first! An introduction to usability and user-centered design and development for technical information and products. In : IEEE International Professional Communication Conference, 2001. IPCC 2001. Proceedings. IPCC 2001. Communication Dimensions. Proceedings IEEE International Professional Communication Conference. Sante Fe, NM, USA, 24-27 Oct. 2001: IEEE / Institute of Electrical and Electronics Engineers Incorporated, pp. 135–140, checked on 6/17/2023.

Archer, R. (2015): ExpressJS: Web App Development with Node.js Framework. Available online at https://www.semanticscholar.org/paper/ExpressJS%3A-Web-App-Development-with-Node.js-Archer/52682e91657acfbde9c8f9267e30a4b6bc10e1f4, checked on 7/9/2023.

Bagnall, Peter (2007): Using Personas Effectively, checked on 6/23/2023.

Barnum, Carol M.: Usability testing essentials. Ready, set…test! 2nd edition. Amsterdam: Morgan Kaufmann, an imprint of Elsevier. Available online at https://books.google.de/books?hl=en&lr=&id=L6_SDwAAQBAJ&oi=fnd&pg=PP1&dq=barnum+carol+ready+set+go+pdf&ots=kAjMU4C7t8&sig=1vksqCid-Hy3_Te_8QL5l34BV9s&redir_esc=y#v=onepage&q=barnum%20carol%20ready%20set%20go%20pdf&f=false, checked on 7/6/2023.

Belkhir, Abdelkarim; Abdellatif, Manel; Tighilt, Rafik; Moha, Naouel; Gueheneuc, Yann-Gael; Beaudry, Eric: An Observational Study on the State of REST API Uses in Android Mobile Applications. In : 2019 IEEE/ACM 6th International Conference, pp. 66–75.

Berglund, A.; Grimheden, M. (2011): The Importance of Prototyping for Education in Product Innovation Engineering. Available online at https://www.semanticscholar.org/paper/The-Importance-of-Prototyping-for-Education-in-Berglund-Grimheden/b03d23fafaf9c0f3bc3806f741d434d778f0026b, checked on 6/11/2023.

bin Uzayr, Sufyan; Cloud, Nicholas; Ambler, Tim (2019): Mongoose. In : JavaScript Frameworks for Modern Web Development: Apress, Berkeley, CA, pp. 309–375. Available

online at https://link.springer.com/chapter/10.1007/978-1-4842-4995-6_9, checked on 7/17/2023.

Braxton, John M. (2016): How College Affects Students: Volume 3. 21st Century Evidence That Higher Education Works by Matthew J. Mayhew et al. (review). In *Journal of College Student Development*. Available online at https://www.semanticscholar.org/paper/How-College-Affects-Students%3A-Volume-3.-21st-That-Braxton/6d57faa5d71eb5b2a063490d217eb4d1e0342827, checked on 6/28/2023.

Challapalli, Sai Sri Nandan; Kaushik, Prakarsh; Suman, Shashikant; Shivahare, Basu Dev; Bibhu, Vimal; Gupta, Amar Deep: Web Development and performance comparison of Web Development Technologies in Node.js and Python. In : 2021 International Conference on Technological, pp. 303–307, checked on 7/10/2023.

Chandra, Vishal (2015): Comparison between Various Software Development Methodologies. In *International Journal of Computer Applications* 131 (9), pp. 7–10. Available online at https://www.academia.edu/29931163/Comparison_between_Various_Software_Development_Methodologies, checked on 7/1/2023.

Chauhan, Anjali (2019): A review on various aspects of MongoDb databases. Available online at https://d1wqtxts1xzle7.cloudfront.net/60661268/a-review-on-various-aspects-of-mongodb-databases-IJERTV8IS05003120190921-10051-a8lu8n-libre.pdf?1569063684=&response-content-disposition=inline%3B+filename%3DIJERT_A_Review_on_Various_Aspects_of_Mon.pdf&Expires=1697503470&Signature=bXPmtHrHNpe8EtJzjqku4pTPN~bgf22-lZrb3HoUE~Pmn-Db2Vdk~-ZIz5HFqo6yR8r0QkjceBjXtIMqmr9h-cdH6RuEt-M9QPTxNiHebyLwmP~lE6HmYyDghetpEsCtq8WVg4Oxzfbq676TgGP3emwnVlmAUmpp-x0n6G0wYqG-UVx4VEZM2RjKPqoHouL9IzeyyBcmgWY9Q5owEJZRhNtTOk7yHO0JU1YAgqubkQ~HQUJ0YWo3nut2OZK0PHAZd9mKZgcxqEhou~xB6U-pvC6vnDAU6LJ7S8VF5TdgeJ90JvRRnQseP~UBB2LR7AnQpIyqI1OcdUTww4GVk986ew__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA, checked on 7/12/2023.

Christa, Sharon; Suma, V. (2017): Analysis of ticket severity: A case study. In IEEE Staff (Ed.): 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). 2017 International Conference on Innovative Mechanisms for Industry

Applications (ICIMIA). Bengaluru, India, 2/21/2017 - 2/23/2017. Piscataway: IEEE, pp. 763–765, checked on 6/29/2023.

Dam, Rikke; Siang, Teo Yu (2023): 5 Common Low-Fidelity Prototypes and Their Best Practices. Available online at https://www.interaction-design.org/literature/article/prototyping-learn-eight-common-methods-and-best-practices, checked on 7/2/2023.

Dam, Rikke; Siang, Toe (2021): What is Design Thinking and Why Is It So Popular. Available online at https://athena.ecs.csus.edu/~buckley/CSc170_F2018_files/What%20is%20Design%20Thinking%20and%20Why%20Is%20It%20So%20Popular.pdf, checked on 6/20/2023.

Fleischmann; Katja; Visini; Gemma; Daniel, Ryan (2012): WE WANT TO ADD TO THEIR LIVES, NOT TAKE AWAY…

Hensley, Reena (2022): What Is Front-End Development and How Does It Impact Your Website? Available online at https://www.sitecrafting.com/articles/what-is-front-end-development/, checked on 7/9/2023.

Hesen, Li; Ying, Luo (2019): Laundary Design of Public Self-service Laundry System in College Dormitory, checked on 6/27/2023.

John Kim (2017): Why Error Handling Is So Important. Available online at https://www.profocustechnology.com/general/error-handling-important/#:~:text=When%20we%20neglect%20to%20make,things%20that%20unexpectedly%20go%20wrong., checked on 7/7/2023.

Khan, Md. Akib Zabed (2018): 9 Dorm managemnt report, checked on 6/3/2023.

Khan, Wisal; Kumar, Teerath; Zhang, Cheng; Raj, Kislay; Roy, Arunabha M.; Luo, Bin (2023): SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review. In *BDCC* 7 (2), p. 97. DOI: 10.3390/bdcc7020097.

Kokelj, Andrej (2018): Web system for managing resources in laundry. Available online at https://www.semanticscholar.org/paper/Web-system-for-managing-resources-in-laundry-Kokelj/6121ba95ae3ee837c7520a1be8f1db87b5e528e5, checked on 6/29/2023.

Krause, Jörg (2017): Introduction to Node. In : Programming Web Applications with Node, Express and Pug: Apress, Berkeley, CA, pp. 15–46. Available online at https://link.springer.com/chapter/10.1007/978-1-4842-2511-0_3, checked on 7/21/2023.

Mgbemena, Chinedu Victor (2023): Introduction to EJS: A Guide to Building Dynamic Web Applications. Available online at https://dev.to/victrexx2002/introduction-to-ejs-a-guide-to-building-dynamic-web-applications-2737, checked on 7/15/2023.

Nik Marsyahariani Nik Daud; Nor Azila Awang Abu Bakar; Hazlifah Mohd Rusli (2010): 2010 International Symposium on Information Technology. ITSim '10] ; Kuala Lumpur, Malaysia, 15 - 17 June 2010, checked on 6/27/2023.

Ordonez, Maria Isabel; S. Hagy; F. Bard; L. Wahlgren; Beatrice Ringstrand (2017): Room for change: Impact of building-level innovations to facilitate product reuse among residents. Available online at https://www.semanticscholar.org/paper/Room-for-change%3A-Impact-of-building-level-to-reuse-Ordonez-Hagy/4b8743f1d5ea8445420b7d3b58e45ae60b2327ff, checked on 6/22/2023.

Paskhaver, Mary (2023): Principles of Question-Driven Development, checked on 9/18/2023.

Riesener, Michael; Doelle, Christian; Schloesser, Sebastian; Schuh, Guenther (2020): Prototype Design in Agile Product Development Processes for Technical Systems. In : 31th International Conference on Design Theory and Methodology. Presented at ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, August 18-21, 2019, Anaheim, California, USA. ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Anaheim, California, USA, 8/18/2019 - 8/21/2019. ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. New York, NY: The American Society of Mechanical Engineers (Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference 2019, volume 7), checked on 6/29/2023.

Sadiku, Matthew N. O.; Adedamola A. Omotoso; Sarhan M. Musa (2019): Design Thinking, checked on 7/11/2023.

Setthawong, Pisal; Triyason, Tuul; Osothsilp, Anan; Chinggungval, Titinai (2023): Cost-Effective IoT Extensions for Existing Public Coin Operated Washing Machine Towards Smarter Apartment Complexes. In *Curr. Appl. Sci. Technol.* 23 (2). DOI: 10.55003/cast.2022.02.23.001.

Unsplash.com: Free images

Zehra, Zarmeen (2023): What is a high-fidelity prototype? Available online at https://www.educative.io/answers/what-is-a-high-fidelity-prototype, checked on 7/4/2023.

Zola, Andrew (2023): What is Bootstrap? Available online at https://www.techtarget.com/whatis/definition/bootstrap, checked on 7/4/2023.

Appendices

A. Code Snippets

Below shows the schema of MongoDB model created for products which are posted in the marketplace.

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const ProductSchema = new Schema({
    title: String,
    image: String,
    price: Number,
    description: String,
    location: String,
    author: {
        type: Schema.Types.ObjectId,
        ref: 'User'
    },
    comments: [
        {
            type: Schema.Types.ObjectId,
            ref: 'Comment'
        }
    ]
});

module.exports = mongoose.model('Product', ProductSchema);
```

```
const commentSchema = new Schema({
   body: String,
   rating: Number,
   author: {
      type: Schema.Types.ObjectId,
      ref: 'User' // refer to USer model
  },
});
module.exports = mongoose.model("Comment", commentSchema);
```

The above code snippets represent MongoDB schema for comment which are referring to each product posted in the marketplace.

```
const UserSchema = new Schema({
   fname: {
      type: String,
      required: true,
      unique: true
     },
   lname: {
      type: String,
      required: true,
      unique: true
     },
   roomnumber: {
      type: Number,
      required: true,
      unique: true
     },
   email: {
    type: String,
    required: true,
    unique: true
   },
});

UserSchema.plugin(passportLocalMongoose);

module.exports = mongoose.model("User", UserSchema);
```

The above code snippets represent the schema for MongoDB model “Users”

```
const NotificationSchema = new Schema({

    title: String,
    image: String,
    description: String,
    details: String,
    date: {
        type: Date,
        // Set the default time to midnight (00:00:00)
        default: function () {
            return new Date().setHours(0, 0, 0, 0);
        }
    }
});
```

```
module.exports = mongoose.model('Notification', NotificationSchema);
```

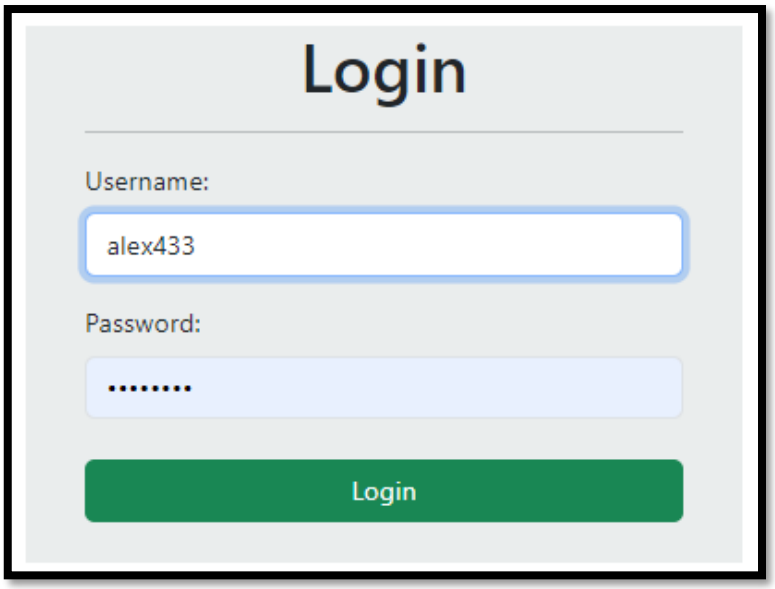
The above code snippets represent MongoDB schema for model Notifications.

The below code snippet is responsible for rendering all the products listed in the marketplace, to post new products in the marketplace, to view the products and edit the products, to show the comments related to each product. And finally, to delete the products. Basically, it shows the CRUD operations on Product model using RESTful routes.

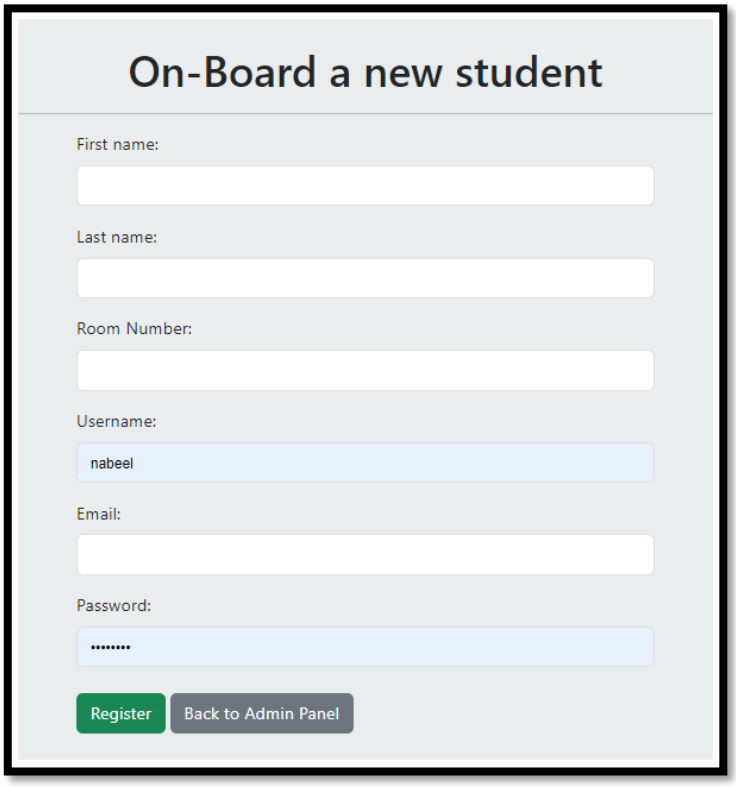
```
app.get('/products', isLoggedIn, async(req, res, next)=>{
    try{
      const products = await Product.find({});
      console.log(products);
    res.render('products/index.ejs', {products})
    }
    catch(e){
      next(e)
    }
})

app.get('/products/new', isLoggedIn, (req, res)=>{
    res.render('products/new.ejs')
})
app.post('/products', isLoggedIn, async(req, res, next)=>{

  try{
      console.log(req.body)
      const product = new Product(req.body)
      product.author = req.user._id;
      console.log(product);
      await product.save();
      req.flash('success', 'Successfully made a new product')
      res.redirect(`/products/${product._id}`)
    }

    catch(error){
      next(error)
    }
})

app.get('/products/:id/edit', isLoggedIn, async(req, res, next)=>{
  try{
    const {id} = req.params;
```

```
    const product = await Product.findById(id);
    if(!product){
      return next(new AppError('No product with id', 404));
    }
    res.render('products/edit.ejs', {product})
  }
  catch(e){ next(e); }})

app.put('/products/:id', async(req, res, next)=>{
  try{
    const {id} = req.params;
    const product = await Product.findByIdAndUpdate(id, req.body);
    console.log(req.body)
    res.redirect(`/products/${product._id}`)
  }
  catch(e){ next(e); }})

app.get('/products/:id', isLoggedIn, async(req, res, next)=>{
  try{
    const {id} = req.params;
    console.log(id)
    const product = await Product.findById(id).populate({
path:'comments',
    populate: {
      path: 'author'
    }
  }).populate('author');
    console.log(product)
    if(!product){
      throw new AppError('No product with id', 404);
    }
    res.render('products/show.ejs', {product})
  } catch (e) { next(e); }})


app.delete('/products/:id', async(req, res)=>{
  try{
    const {id} = req.params;
    const product = await Product.findByIdAndDelete(id);
    res.redirect('/products');
  }
  catch(e){ next(e);}})
```

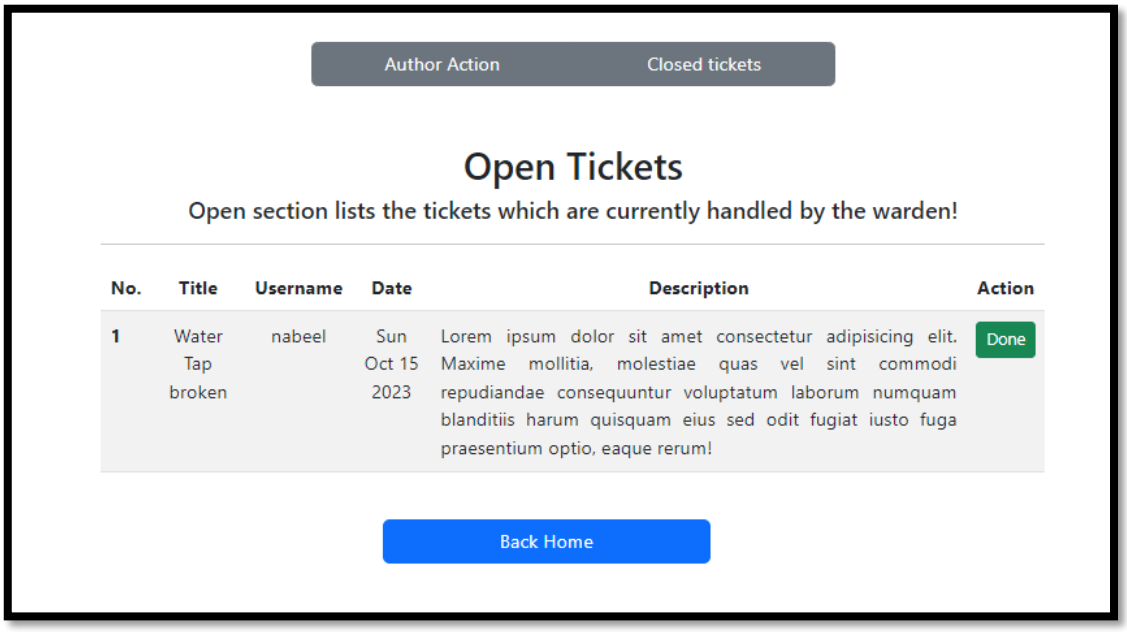
B. Screenshots

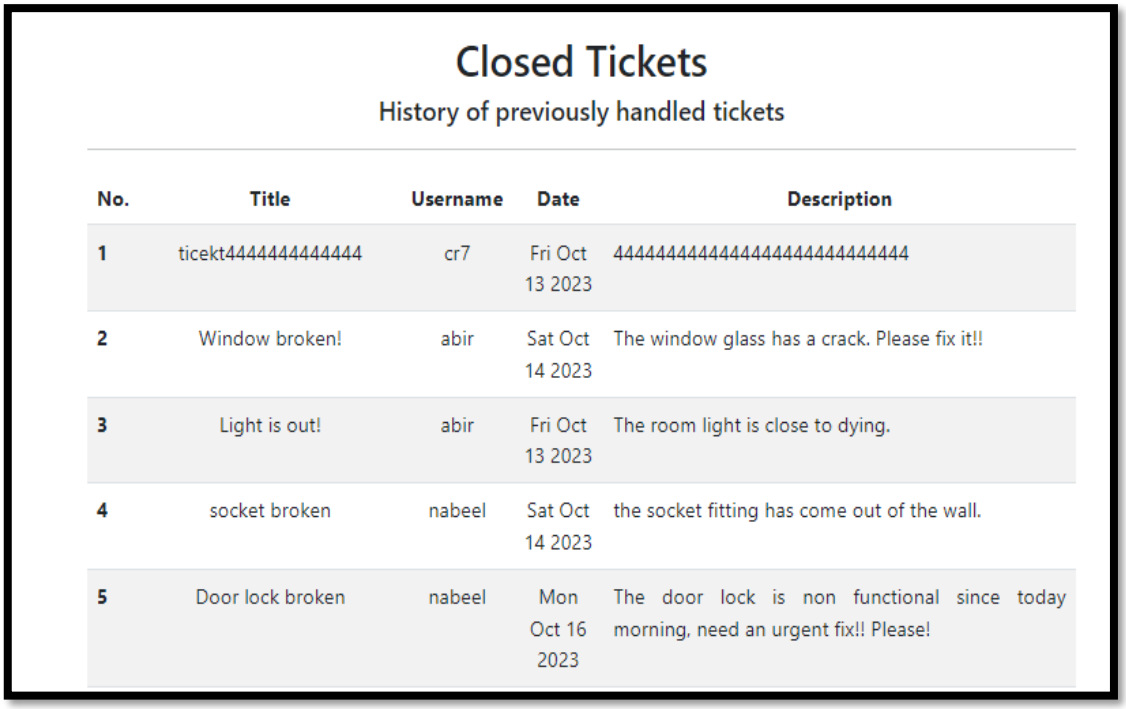


The unified login page shows login form for both users and administrators.

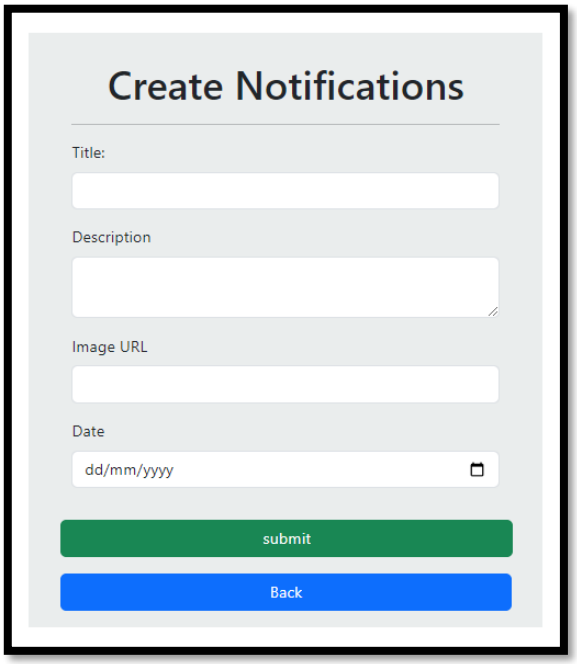


Admin side registration form for new users.

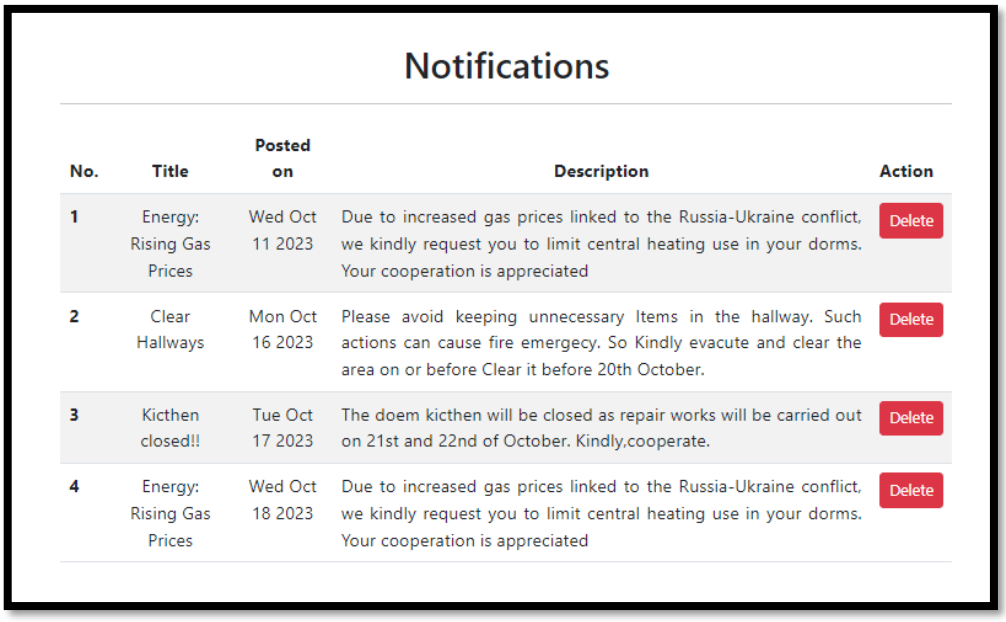




The above two screenshots show the “Open Tickets” and “Closed Tickets” on admin side. In the open tickets section shows the tickets created by the users. The admin can fix the ticket and click “done” to move the ticket into Autor action from where the user can re-open or close the ticket. The closed tickets section shows a history of all tickets previously created and closed tickets by the user.



The above screenshots show the form to create a new notification from the admin, which will be broadcasted to all the residents.



The above screenshot is the page at the admin end from where the admin can view all notifications along with its posted date. The admin also deletes the notification once it gets out of date.