Online Temporary Residential Facility Reservation System Software Design Specification

23.08.2020
Version 1.1

Kaif Muhammed
Team Codemo
EEI4189
Department of Electrical and Computer Engineering
Bachelor of Software Engineering

Table of Contents

1. Introduction

The Open University of Sri Lanka is an educational institute which serves a larger student base now. Also it provides a temporary residential facility (hostel) to their students.

The hostel premises can serve 400 students at time. Because of the Open and distance learning mythology The Open University does not provide the students with permanent accommodation facilities.

Because of the manual work at the hostel premises the hostel staff and the students had faced many difficulties to allocate a room or to get a room at the hostel. Because of this we have approached an Online Temporary Residential Facility Booking System for the University Staff and the students.

Goals are to use service coordination and technology integration to:

Ease the workload of the Hostel Staff and hassle free room booking system to the students.

Simply Get There Overview

The Online Hostel Booking System enables management of the hostel rooms, bookings for the University Hostel administration and enables to book a room for the students. Because of the web base system this could be accessed at anywhere who has access to the system and get the live details from the system.

SGT Summary:

- Discovery of available rooms for the students
- Book available rooms
- Management of room booking for the administrative users
- Unique to the Open University Hostel
- Responsive design for use on computers, tablets, and smartphones

1.1. Document Outline

Below is the outline of the each section described in this document.

- Chapter 1 – Document Description
- Chapter 2 - System Overview
- Chapter 3 – Design Considerations
- Chapter 4 – Architectural Strategies
- Chapter 5 – System Architecture (Use Cases from SRS)
- Chapter 6 – Policies and Tactics
- Chapter 7 - Design Documents
- Chapter 8 – Glossary

1.2. Document Description

1.2.1 Introduction

The project is to create an Online Temporary Residential Facility Reservation (OTRFR) system at the Open University of Sri Lanka. The System should be able to reserve rooms to the authenticated users, keep track of reservations, and be accessed only by the authorized personnel.

This Design Document presents the designs used or intended to be used in implementing the project. The design described, follows the requirements specified in the Software Requirements Specification document prepared for the project.

1.2.1.1 Purpose

The purpose of this document is to present a detailed description of the designs of the OTRFR System, created for the Open University of Sri Lanka. Firstly, this document is intended for the programming group in Team, to use the designs as guidelines to implement the project. Equally, this document is also for the team's instructor, Ms. Jeeva Padmini, as it fulfils one of the requirements of the project. Lastly, this document could be used for designers who try to upgrade or modify the present design of the OTRFR system.

1.2.1.2 Scope

This document gives a detailed description of the software architecture of the OTRFR system. It specifies the structure and design of some of the modules discussed in the SRS. It also displays some of the use cases that had transformed into sequential and activity diagrams. The class diagrams show how the programming team would implement the specific module

1.2.1.3 Audience

The intended audience for the OTRFR System is the project Supervisor, project team, and the future development team. The audience or users for this system design document include the following:

• Codemo Team

• Future application development team

1.2.1.4 Document Conventions

No document conventions are being used at this time.

1.2.1.5 References

The user of this SDD may need the following documents for reference:

IEEE Standard 1016-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, 1998.

Codemo 10.2019, Software Requirement Specification (Online Temporary Residential Facility Reservation)

1.2.1.6 Definitions, Acronyms, and Abbreviations

- DBMS – Database Management System. A programmable interface which provides a common layer of abstraction between a physical database and a user or external program.
- HTML – Hypertext Markup Language. Set of markup symbols or codes intended for display on a World Wide Web browser page. The markup instructs a web browser on how to display words and images for a web page.
- PHP – Hypertext Preprocessor. An extensible scripting language, suited for web-based development, typically embedded in HTML.

1.2.1.7 Overview

This document is written according to the standards for the Software Design Documentation explained in “IEEE Recommended Practice for Software Design Documentation”.

Section 1.0 introduces the project. Section 2.0 provides an abstract view of the system architecture, including the components, structure and relationships, and user interfaces. Section 3.0 describes each of these components in more detail, including design and architectural decisions. Section 4.0 explores the relationships to other products. Section 5.0 discusses design decisions, and the reasoning behind these decisions. Section 6.0 is reserved for policies and tactics. It also discusses design patterns that can be applied. Section 7.0 has detailed diagrams

2. System Overview

Module	Description
User Login Form	The starting page which asks for login credentials.
Registration Form	Administrator creates the registration forms for the students etc. This is also the form where the users are added
My Profile Page	This pages provide the information of the logged in account to the user.
Booking Page	The pages that provide the system users to allocate room, view booking history in the system
User Accounts Page	This page provide the information of the currently available user accounts, user account create requests in the system to the Administrator
Report Page	This page provide the information to the Administrator for the summery of the booking details
Room Details Page	This page gives the information of the rooms and beds available at the hostel to the Administrator.
Rules and Regulation Page	The pages that provide the information to the Administrator and voters for the functionalities of the pages

3. Design Considerations

This section describes many of the issues which need to be addressed or resolved before attempting to devise a complete design solution.

3.1. Assumptions and Dependencies

As a matter of client use, this web application is operating in dependent but as a matter of web hosting server, there should be server with the operating system such Microsoft Windows professional, Windows Vista Business, Windows Vista Ultimate, Windows which includes Internet Information Services that support HTML5 and JavaScript

The web application will work on all operating systems as long as those systems are equipped with a web browser program that is compatible with html5

Even Though the computer skills of most end- users of OTRFR are identified as intermediate, it cannot be assumed that all users are at that level. Therefore the aim is to make this system as easy to understand as possible. This means that the core functionality of the system will be made simple enough for users with poor computer and internet knowledge

3.2. General Constraints

End-user environment

Requirement: A web page based system

Impact: this requirement is by far the one with the biggest influence on the system design. It imposes the web browser limitations on the user interface, server-client considerations such as amount of information that is going to be passed between the two, data encryption etc.

Network communications

Network should be up all the time as part of the functionality is to be able to use this application on the network or internet.

Security requirements

Requirement: A login page is necessary. This will be handled by JavaScript configurations for user roles. Each user has a login username and password.

3.3. Goals and Guidelines

Apply The KISS principle ("Keep it simple and straightforward!").

The eight requirements that identify for a good design which are well structured, simple, efficient, adequate, flexible, practical, implementable and standardized are the guidelines to create this design.

Emphasis on speed versus memory use.

Working, looking, or "feeling" like an existing online application.

The goal of this project is to deliver the product completed on time. Use all the recommended models in the design document during coding. At the end we will demonstrate a prototype of the application.

3.4. Development Methods

The waterfall approach was identified as the most desirable among the group members and therefore adopted for the system design.

- Analyze the project requirements from the stakeholders
- Design the database by sketching the ER diagram
- UML Designs (Use case diagram)
- Analyzing the classes that will be needed for implementation
- Design the class diagram
- Development of the prototype

4. Architectural Strategies

The architecture and design has been influenced by the following design decisions and strategies:

- The overall system is designed upon a three-tier approach. This is an established and well-understood architecture and presents no problems to the team.
- Authentication will be limited to password checking on initial login and a session ID subsequently. This is considered sufficient to the low risk nature of the data.
- Details of sessions, User accounts and the data inputted by the users will reside in the DBMS. Access to this data will be easier and more secure than creating files on file system.
- Communication with the users will be by the HTML protocol as this is well-supported by the selected browsers and PHP.
- PHP and MySQL were selected because they had the necessary capabilities to provide the needed services to the user and their GNU licenses will reduce product cost.
- The team decided to not use PHP's built in session functions but to provide our own for increased flexibility.
- The system is meant to be modifiable:
    - By the use of new pre-defined requirements as well as the end-customer.
    - By having the system adapt to additional fields added by university to the database schema
    - By changes in functionality through code modification and replacement by University.

5. System Architecture

The architecture used for this web application is the Three-Tire-Architecture. The three-tire layered architecture has three different tires being Presentation (Client Tier), Application Logic, and the Technical Services. The presentation tier (client tier) is basically the most upper level of the structure which is used to present the information on the client machine. The Application Logic is the tier where all the PHP and JavaScript codes are being held and stored. And the Data tier is the tier for holding the database stored procedures and triggers.

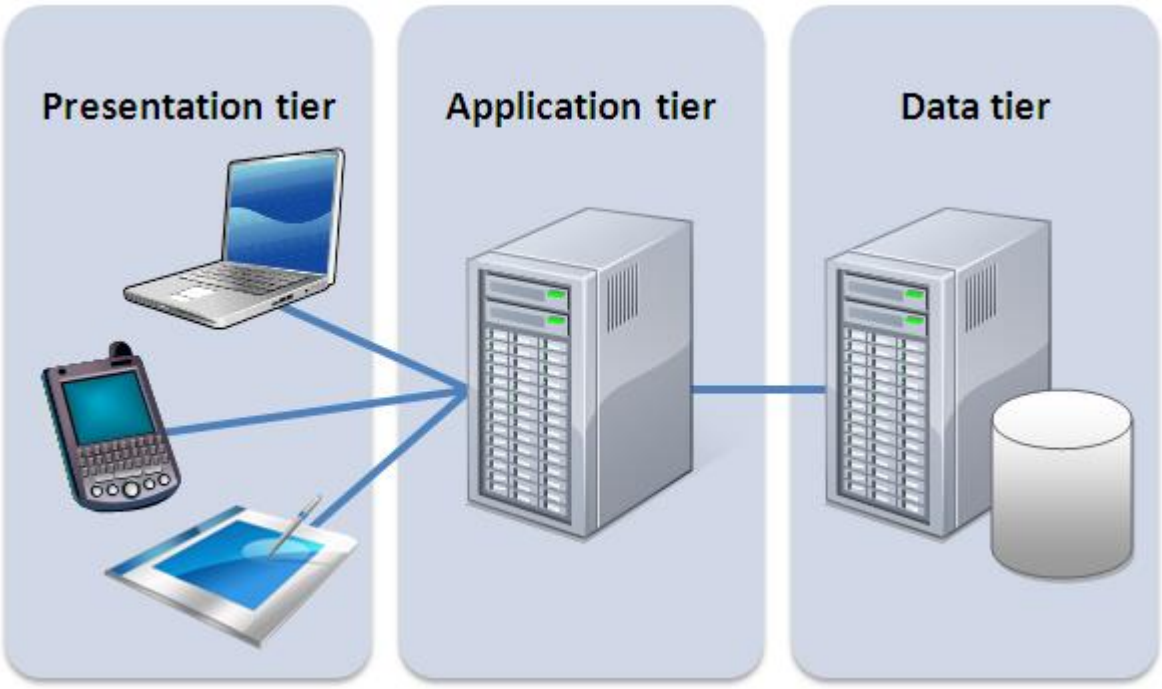


Figure: Three-Tier Architecture

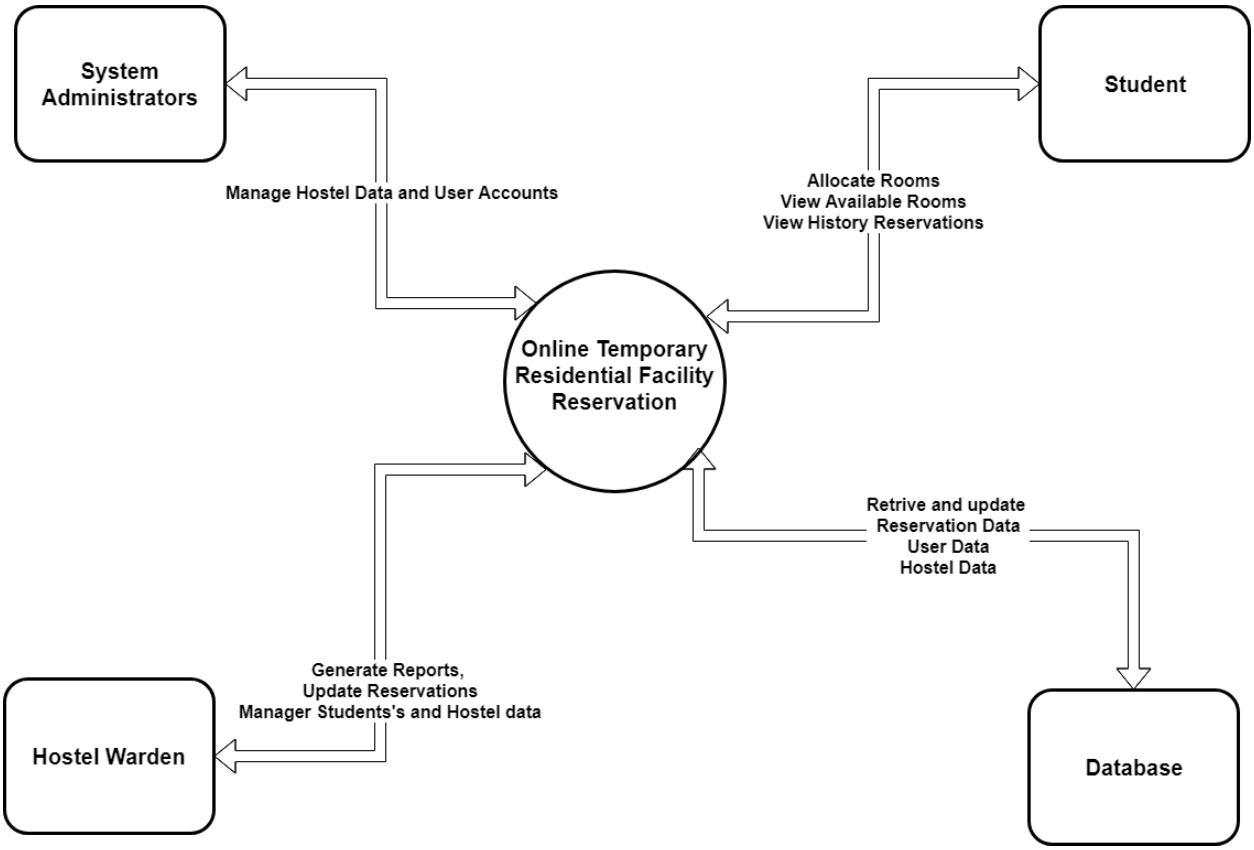


Figure: Context Diagram for OTRFR

5.1. Subsystem Architecture

In this section high-level overview of how the functionality and responsibilities of the system were partitioned and then assigned to subsystems or components are provided. Detail about the individual components themselves will be discussed in the detailed de-sign part of this document

Use Case 1 Specification:

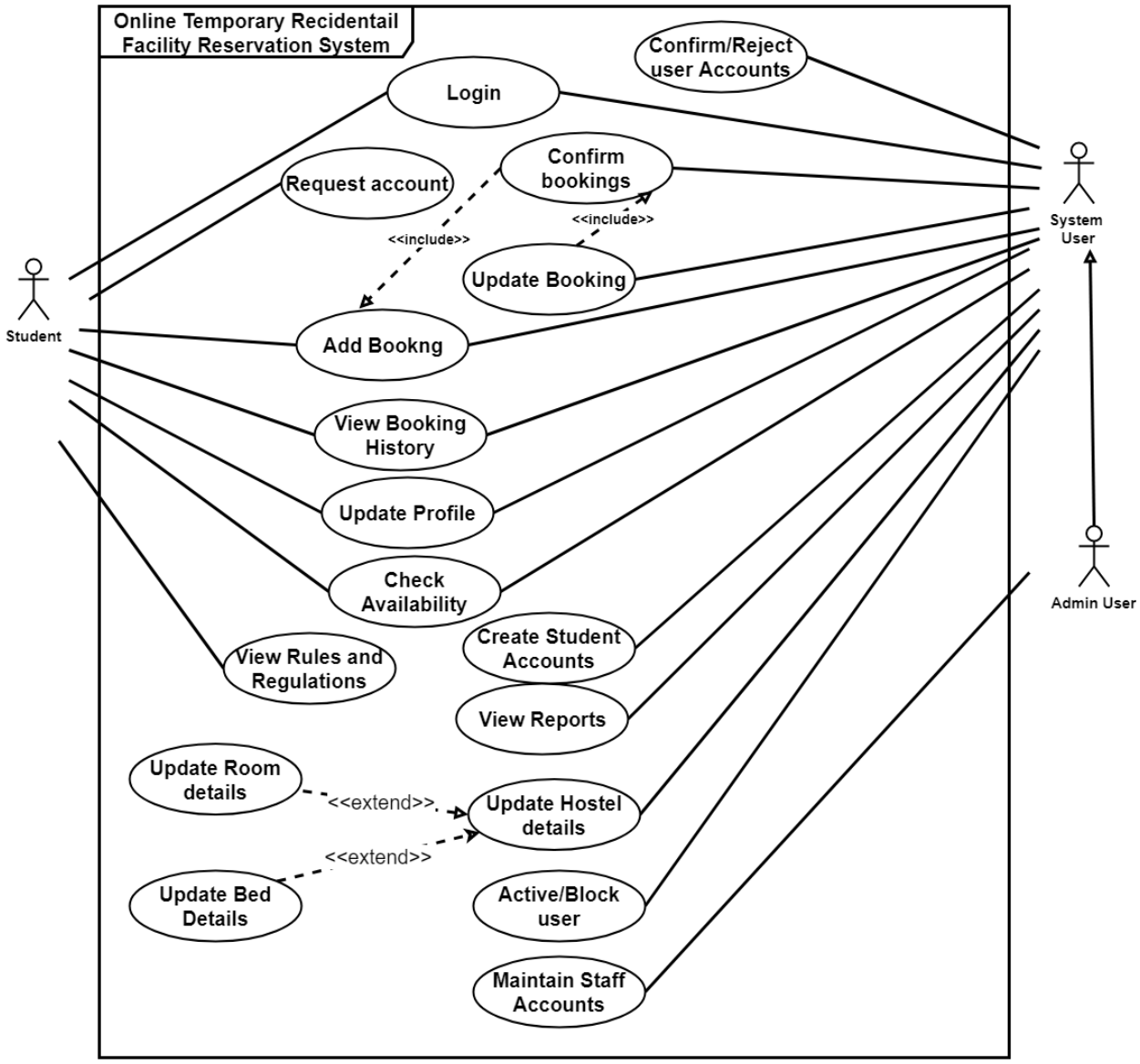


Figure: Use Case Diagram

Use case Id	1
Use case Name	Logging
Primary Actors	Admin User, System user, Student
Use case Description	This Use case describes how to access the OTRFR system
Pre-Conditions	1. Remember the username and password
Post Condition	Redirected to the Home Page of the System
Basic Flow	Navigate to the login page Enter the username and password Click on submit button If user entered wrong credentials Then an error message will be shown and request to try again Else user will be directed to the Home page
Alternative Flow	Navigate to the login page Click on forgot password link Enter new password Click on submit button
Exceptional Flow	if the internet connection is not stable or server is under maintenance user will not be able to login to the system

Use case Id	2
Use case Name	Request Account
Primary Actors	Student
Use case Description	User can request an account to access the system
Pre-Conditions	Should not have an account Should be an Open University Student (Registered)
Post Condition	a user account will be created
Basic Flow	Navigate to the Registration page Fill the Form Click on Register Button
Alternative Flow	N/A
Exceptional Flow	N/A

Use case Id	3
Use case Name	Confirm accounts
Primary Actors	Admin User, System User
Use case Description	User can add new students to access the system
Pre-Conditions	Should request an account
Post Condition	An active user account
Basic Flow	Navigate to the User Account menu Click on Request Sub menu View the Request of students Click on Confirm to accept the request Click on Reject to Deny the request
Alternative Flow	N/A
Exceptional Flow	N/A

Use case Id	4
Use case Name	Add Bookings
Primary Actors	Admin User, System user, Student
Use case Description	User can request to allocate a room
Pre-Conditions	Login to the system Should not have a booking on the same date Should not exceed the maximum staying days (continues or total)
Post Condition	Add a successful booking request
Basic Flow	Navigate to the Booking menu Click on New booking menu Select the staying date Select a floor Select a room no Add the reason Click on request
Alternative Flow	N/A
Exceptional Flow	N/A

Use case Id	5
Use case Name	Confirm booking
Primary Actors	Admin user, System User
Use case Description	User can accept or deny the requests of hostel reservations
Pre-Conditions	Booking should be added
Post Condition	Allow or deny the request
Basic Flow	Navigate to the Booking menu Click on Booking requests View the booking request Click approve to allow the reservation Click reject to deny the reservation
Alternative Flow	N/A
Exceptional Flow	N/A

Use case Id	6
Use case Name	Update Booking
Primary Actors	Admin user, System User
Use case Description	User can change the details of the confirmed Reservation
Pre-Conditions	Booking should be accepted
Post Condition	Change the necessary details in the booking
Basic Flow	Navigate to the Booking Menu Click on update booking Change the necessary details (date change, extend date etc.) Click on update
Alternative Flow	N/A
Exceptional Flow	N/A

Use case Id	7
Use case Name	View Booking History
Primary Actors	Student, System User, Admin User
Use case Description	User can check the previous booking details
Pre-Conditions	Should have allocate rooms previously
Post Condition	View the previously booked rooms and details
Basic Flow	Navigate to the Booking Menu Click on Hostel Bookings
Alternative Flow	N/A
Exceptional Flow	N/A

Use case Id	8
Use case Name	Update Profile
Primary Actors	Student, Admin user, System User
Use case Description	User can view the details of his/her details and can edit some of the available details
Pre-Conditions	N/A
Post Condition	View the user account details
Basic Flow	Navigate to the My profile menu View account details Click on edit to edit details Enter new details Click on update
Alternative Flow	N/A
Exceptional Flow	N/A

Use case Id	9
Use case Name	Check Availability
Primary Actors	Admin user, System User, Student
Use case Description	User can view the availability for a particular date
Pre-Conditions	N/A
Post Condition	View the available rooms
Basic Flow	Navigate to the Booking Menu Click on check availability Enter valid date Click on search View available bookings
Alternative Flow	N/A
Exceptional Flow	N/A

Use case Id	10
Use case Name	View Rules and Regulations
Primary Actors	Student
Use case Description	User can view the rules and regulations of the Hostel Reservations
Pre-Conditions	N/A
Post Condition	View the Hostel Rules and Regulations
Basic Flow	Navigate to the Rules & Regulation menu View the Rules and Regulation Page
Alternative Flow	N/A
Exceptional Flow	N/A

Use case Id	11
Use case Name	Create Student Account
Primary Actors	Admin User, System User
Use case Description	User can create new accounts for the students
Pre-Conditions	Student should not already have an account in the system
Post Condition	Create an active account
Basic Flow	Navigate to User Accounts Click on Create Account Enter relevant details Click on Submit
Alternative Flow	N/A
Exceptional Flow	N/A

Use case Id	12
Use case Name	View Reports
Primary Actors	Admin user, System User
Use case Description	User can generate reports
Pre-Conditions	Record should be available in the database
Post Condition	View/generate relevant reports
Basic Flow	Navigate to the Report Menu Select/fill necessary Fields Click on submit View the report
Alternative Flow	N/A
Exceptional Flow	N/A

Use case Id	13
Use case Name	Update Hostel Details
Primary Actors	Admin User
Use case Description	User can update the room/bed availability etc.
Pre-Conditions	Rooms/Beds must be created to update details
Post Condition	Update the details of the Rooms and Beds
Basic Flow	Navigate to Room details Search the room else add a new room Add new bed or update existing ones Click on save
Alternative Flow	N/A
Exceptional Flow	N/A

Use case Id	14
Use case Name	Change user access
Primary Actors	Admin, System user
Use case Description	Users can block the access to the system of the student/staff account
Pre-Conditions	Student/staff user must have an account
Post Condition	Active or Block Access to the system
Basic Flow	Navigate to the User Accounts Click on Account details Enter the id/name of the student or staff Click on search Select the user by clicking on view Select the appropriate radio button (Active yes/no) Click on update
Alternative Flow	N/A
Exceptional Flow	N/A

Use case Id	15
Use case Name	Maintain Staff Account
Primary Actors	Admin User
Use case Description	Admin user can update the system users account
Pre-Conditions	User should have the admin privileges of the system
Post Condition	Update other system user details
Basic Flow	Navigate to the User Accounts Click on Account details Enter the id/name of the staff Click on search Select the user by clicking on view Update necessary Fields Click on update
Alternative Flow	N/A
Exceptional Flow	N/A

6. Policies and Tactics

In this section we discuss the different policies/tactics that would not significantly affect the overall organization of the system, but which might have a significant effect on the implementations and interface of certain aspects of our program.

1. Specific "Products" Used

Since our software is online based for easier accessibility, JavaScript with PHP programming language will be used. To handle the database, newest MySql Server (current is 5.7) will be used.

2. Plans for testing the software

Since our software is mostly going to be user orientated, which means that ease of use for the user is one of our main priorities, there is going to be a lot of testing in this department. The heuristic evaluation will be used in order to make sure that the user will have an easy time navigating around. Another testing plan would be having a couple of staff members from the University and evaluate our program. By doing this, we can see what our main faults are and hence change the program according to what they think is good or bad.

3. Plans for ensuring requirements traceability

SRS document was provided and all the requirements specified in the document have been applied in the design. Also use cases are created to make sure that all the functionality will be defined in the functions according to the requirements.

4. Interfaces for end-users, software, hardware, and communications

OTRFR system will require end-users to have internet access, user login credentials, PC and a browser.

7. Detailed System Design

7.1. Classification

Class diagrams are drawn for the classes used in this project. Operations and attributes are defined for each class.

7.2. Definition

The specific purpose and semantic meaning of the component are below. This black box model is drawn by referring to the requirements specification document. All the requirements are drawn in this diagram to make it clear for the developer. For additional functionalities main level is divided into sub levels.

7.3. Responsibilities

The primary responsibilities and/or behavior of the forms are:

User Login Form: This is the login form. Login credentials will be provided by the Administrator. According to the login credentials users will be connected either to the Administrator pages, sub-admin pages or student pages.

Election Form: This form will be accessible only by the Administrators. They will be able to create the Elections through this form.

Registration Form: This form also can be accessed only by the Administrators. Administrator can add new users using this form.

Booking Form: Users will be able to allocate rooms using this forms

7.4. Constraints

There won't be any constraints on completing this project. It will be completed on time.

7.5. Uses/Interactions

The interactions between the classes are defined in the class diagrams drawn below

7.6. Resources

This is a server, client tool. Most of the functions will be running from the server and managed through the server.

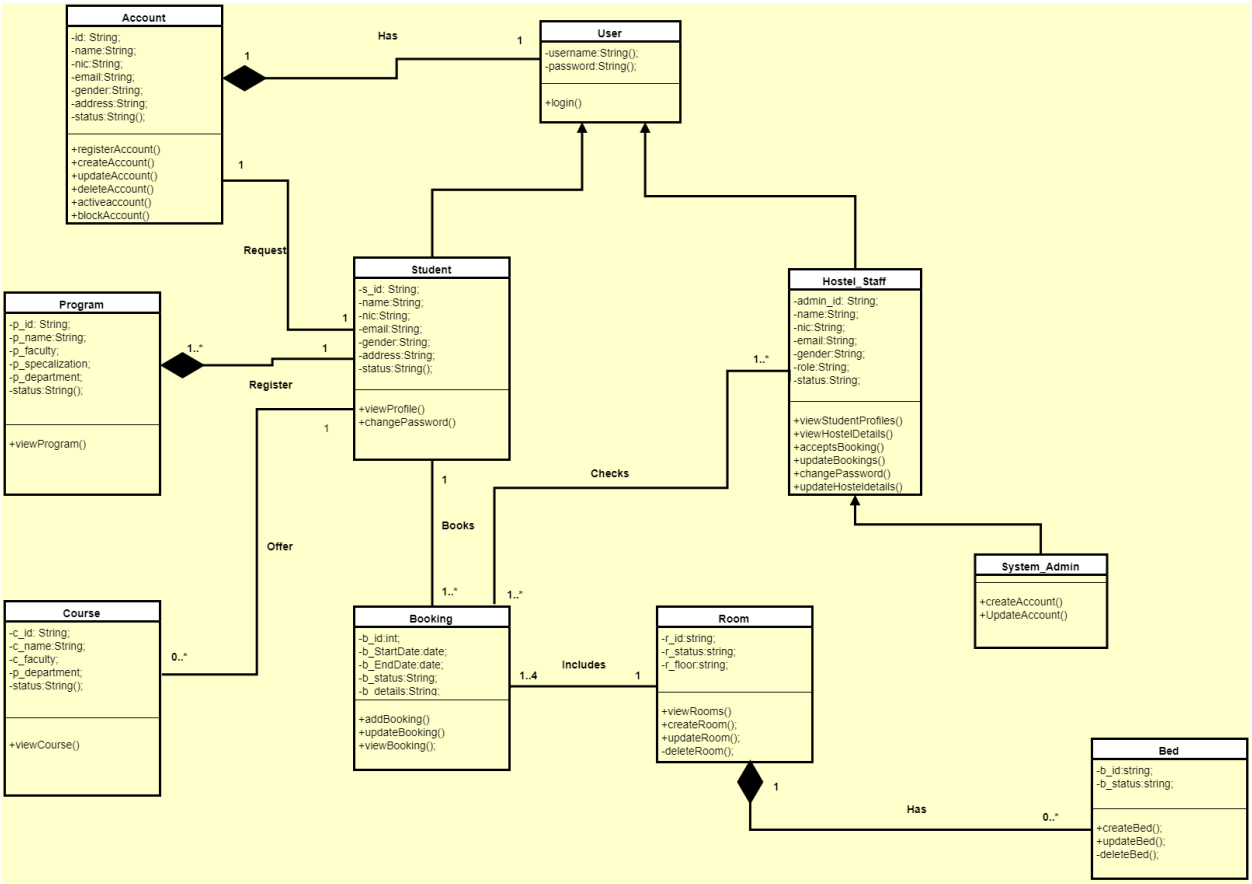
7.7. Processing

Handling of exceptional conditions should be done in each module. All the scenarios that can cause errors need to be handled and not cause applications to crash. Error detection and recovery will be done. To be able to separate error-handling code from the regular code, we will add exception errors in the code. For example, use the following to print the stack trace.

```
catch (Exception e) {

//A (too) general exception handler. Output goes to lblInfoexceptions to display. Additional messages can be added by the developer if needed.

   ...

}
```

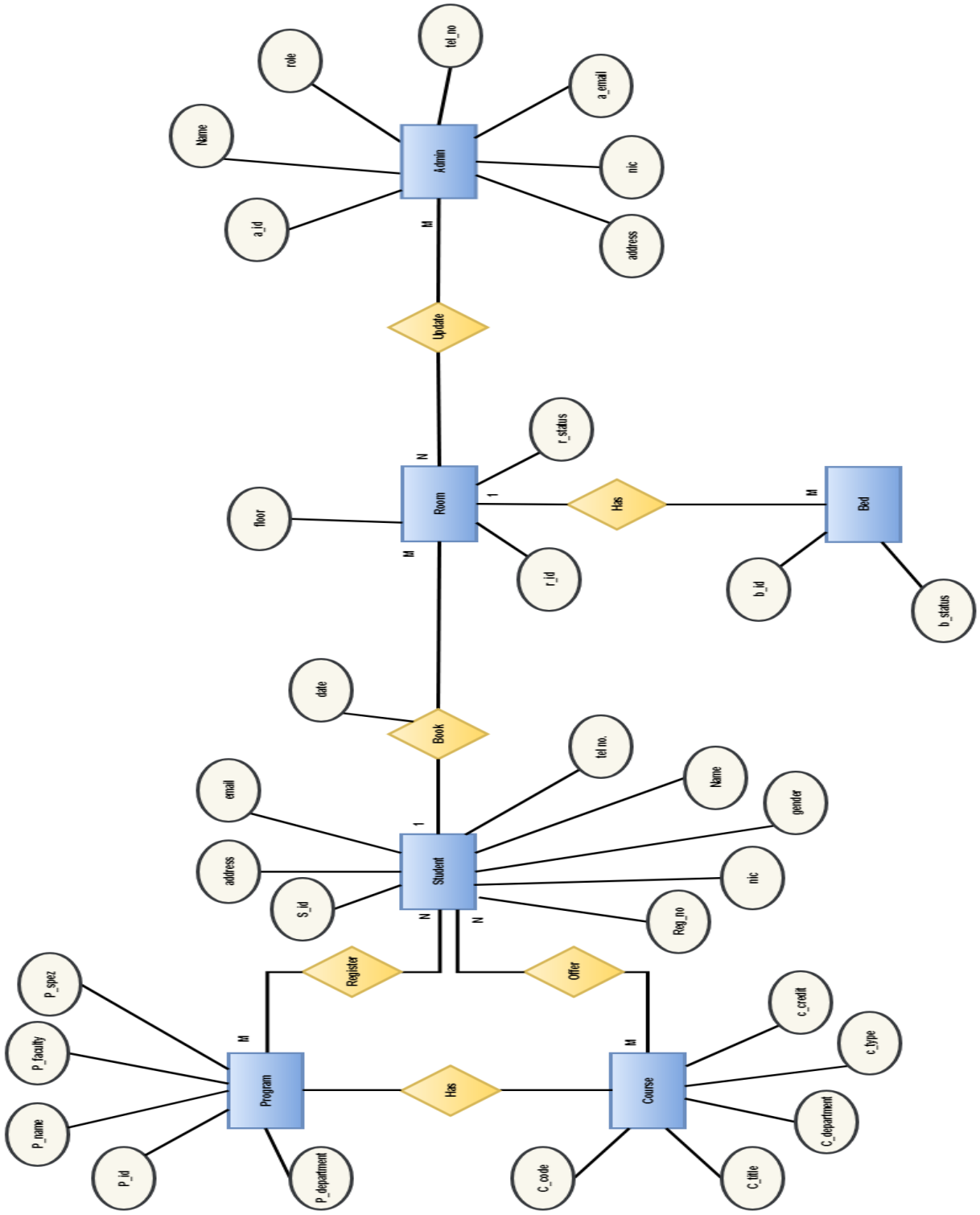
7.8. Detailed Subsystem Design

The class diagram defines a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities.



8. Glossary

Hostel ER diagram



Original User Interface Prototype Pages

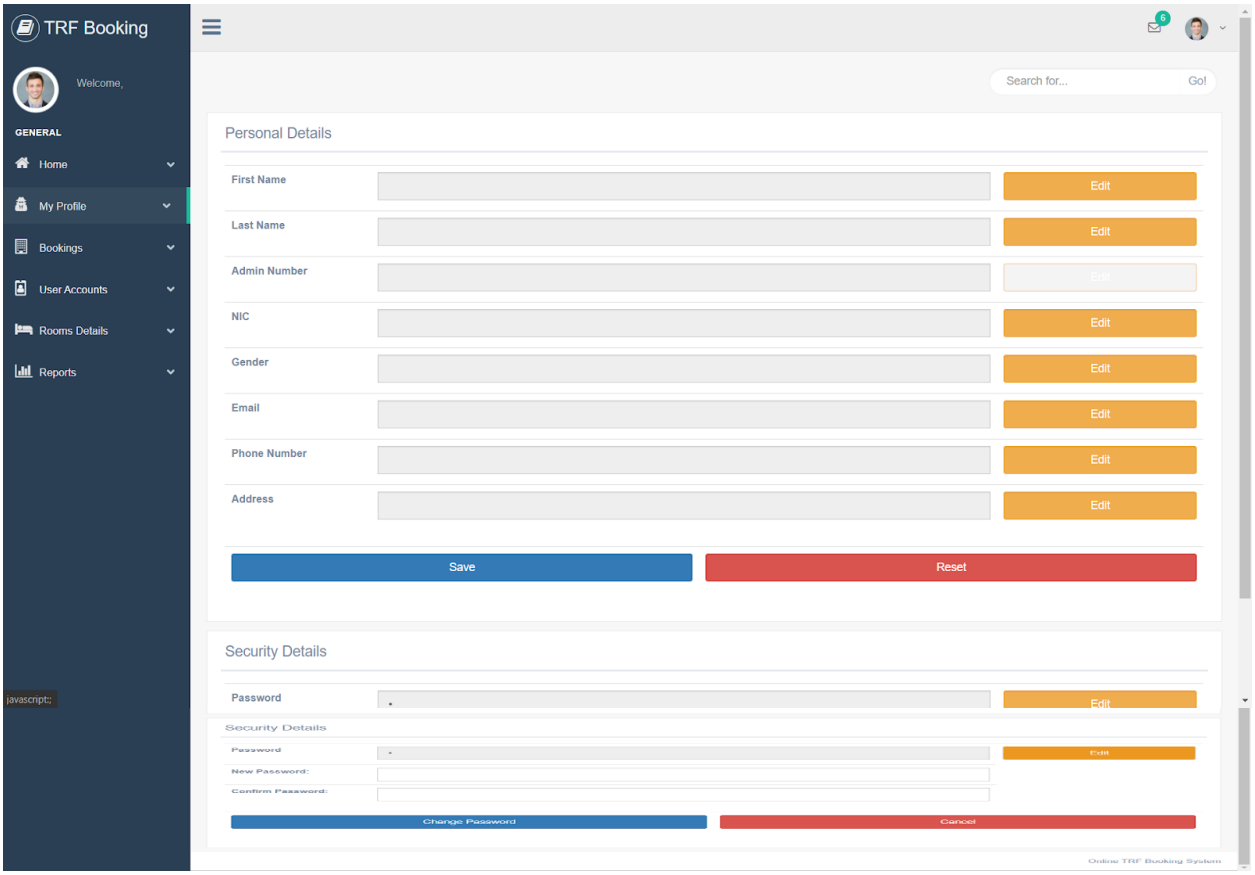


Figure:Personal details

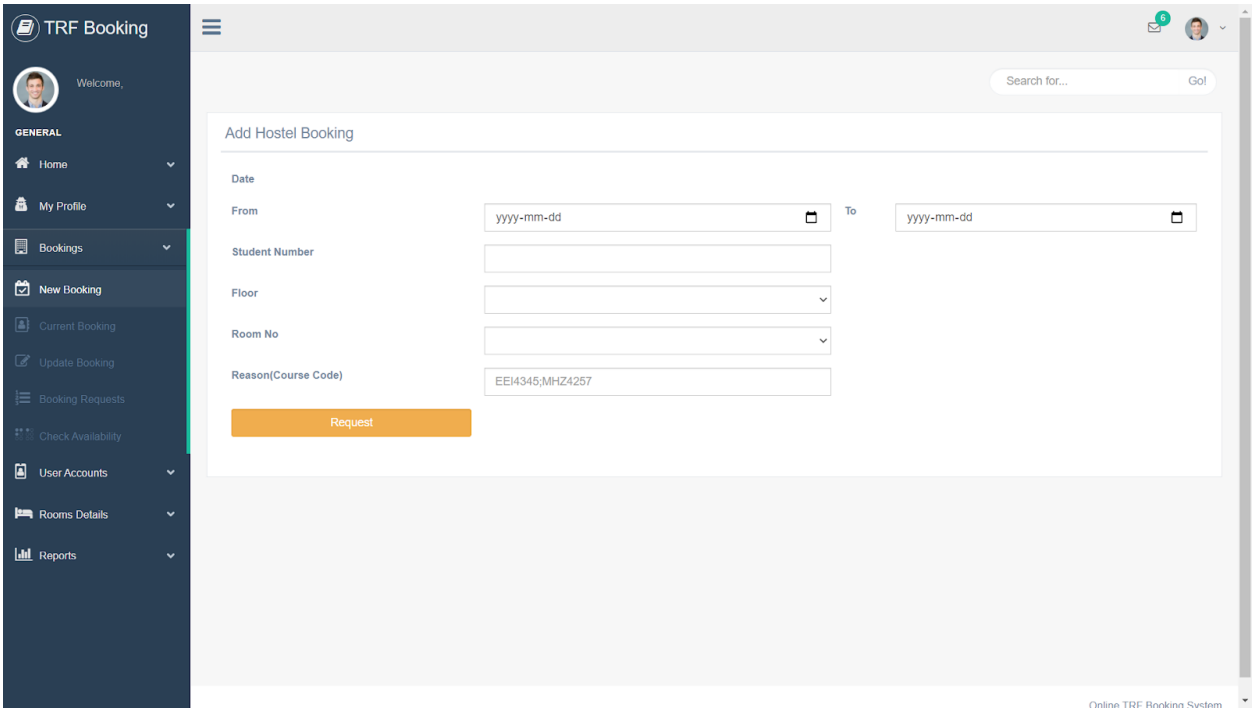


Figure: Add Booking

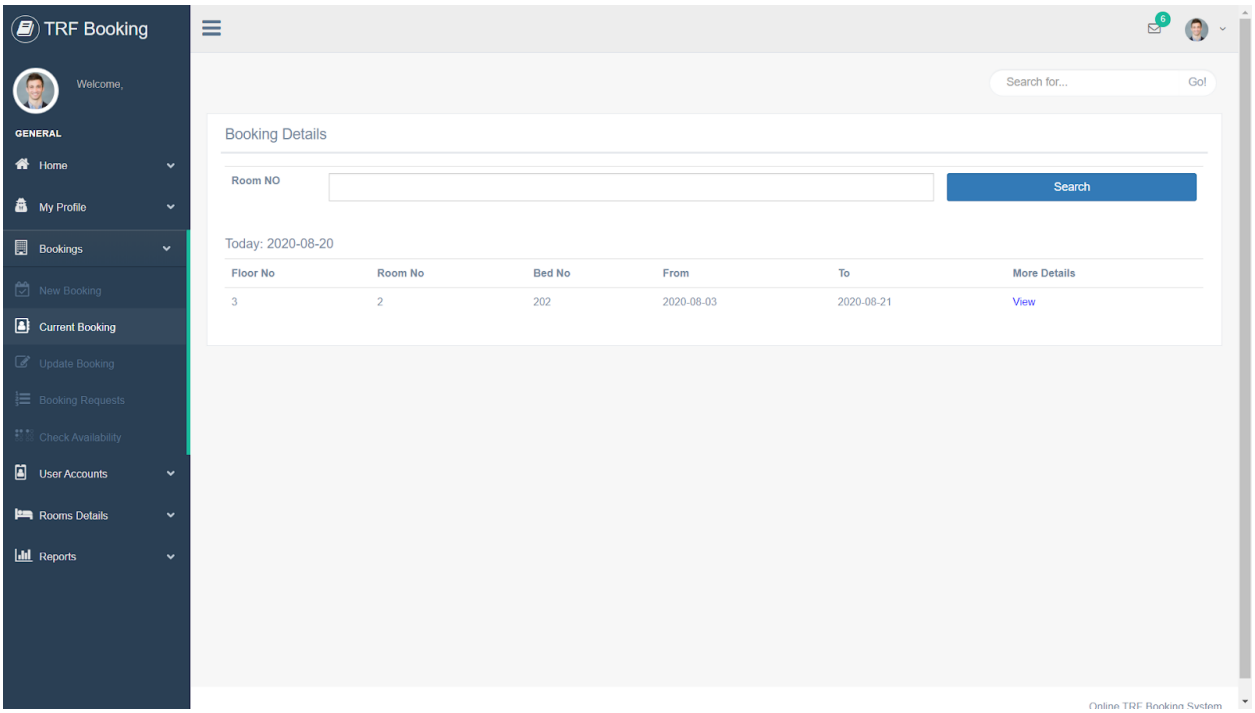


Figure: View Booking Details

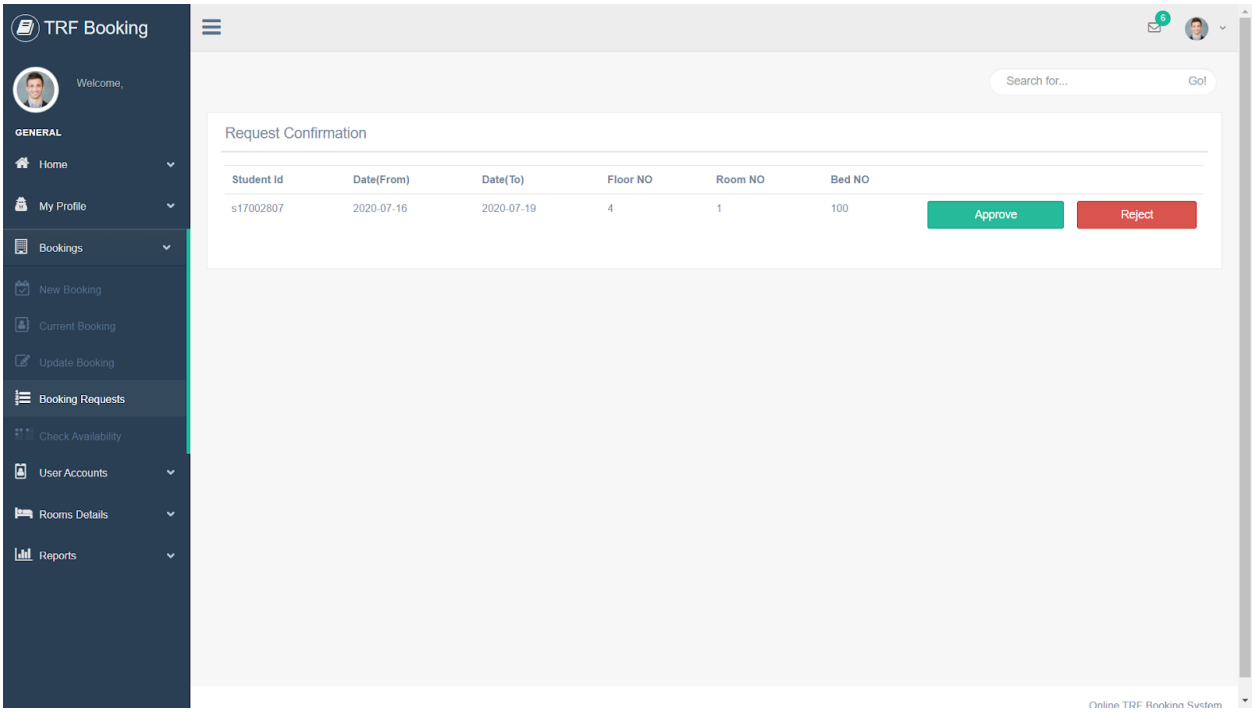


Figure: Booking Confirm

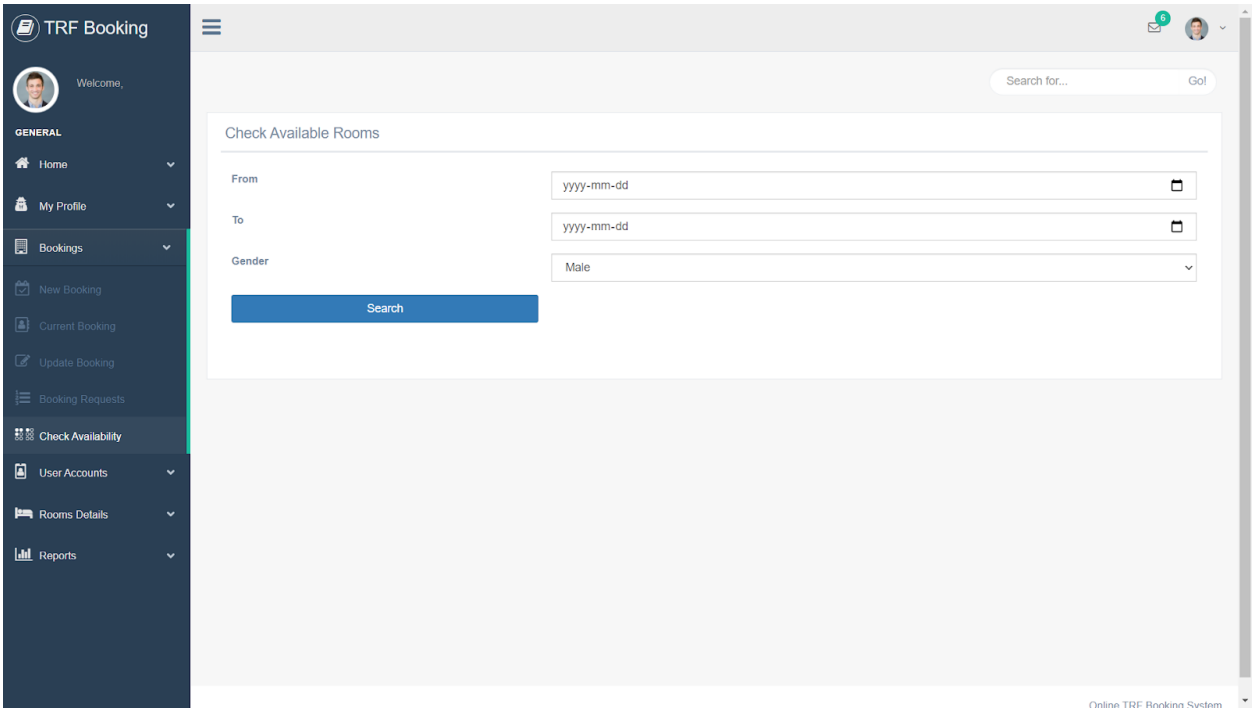


Figure: Check Availability of Rooms

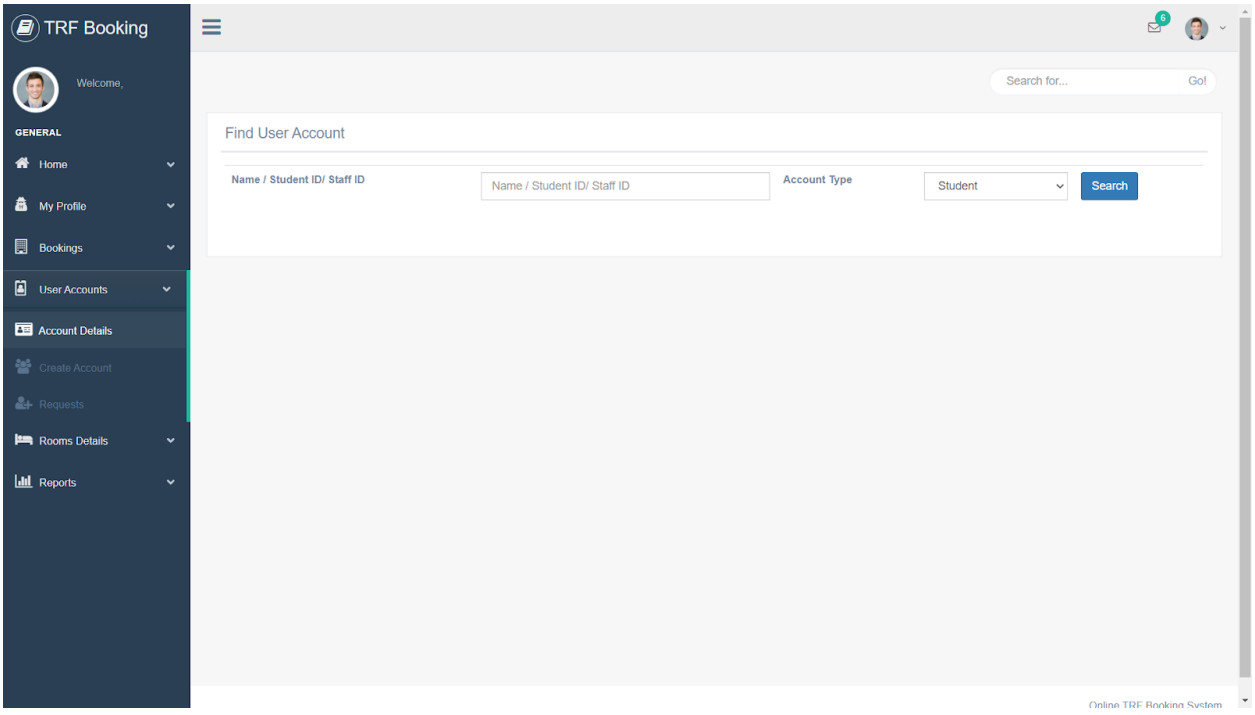


Figure: Find User Account

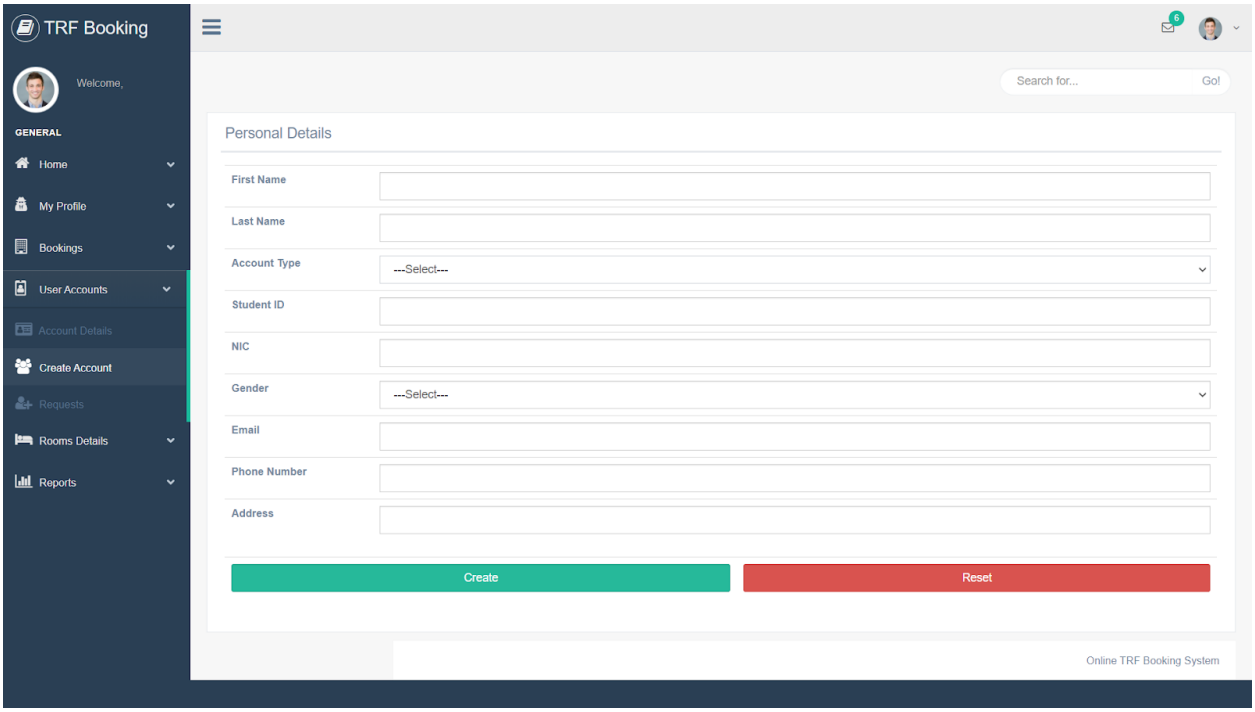


Figure: Create Account

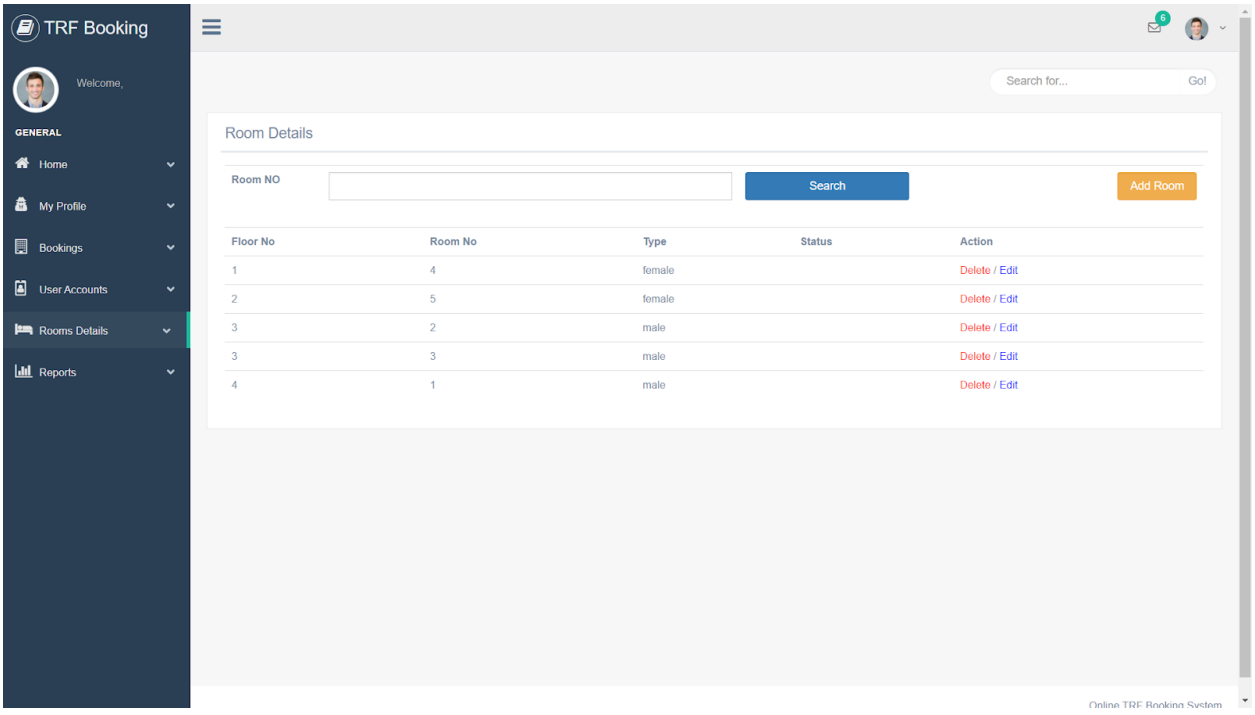


Figure: Room Details

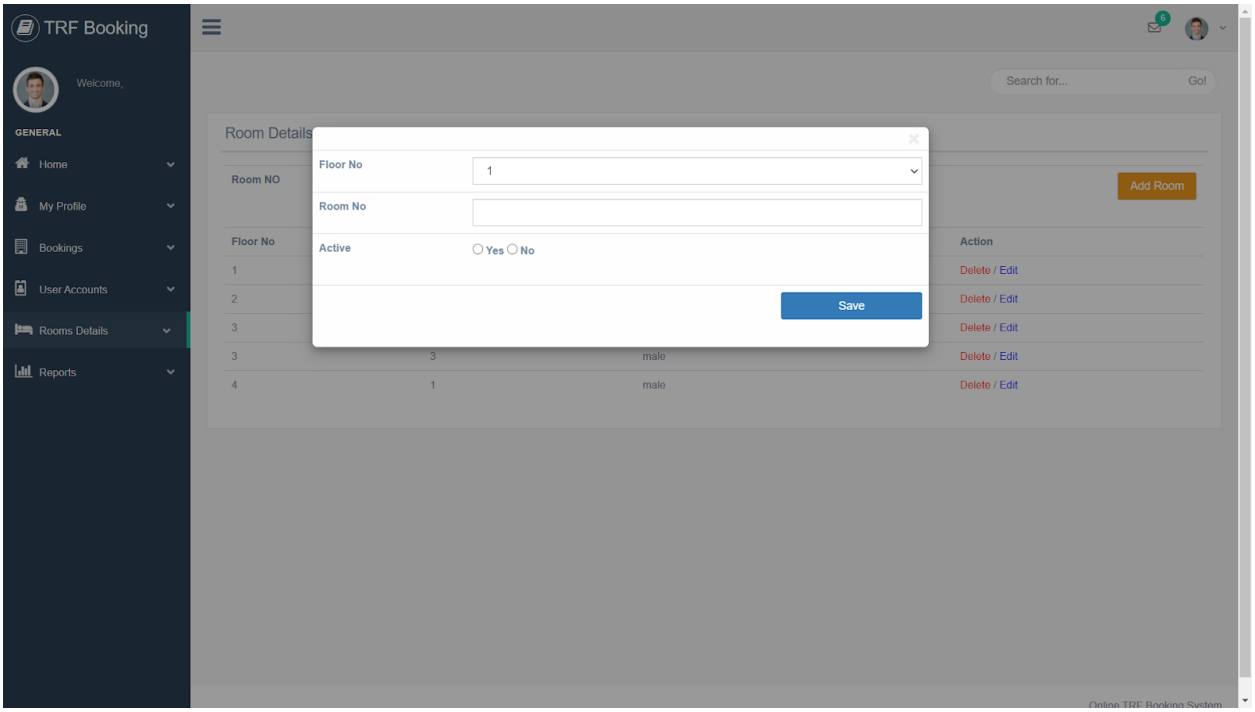


Figure: Add Room

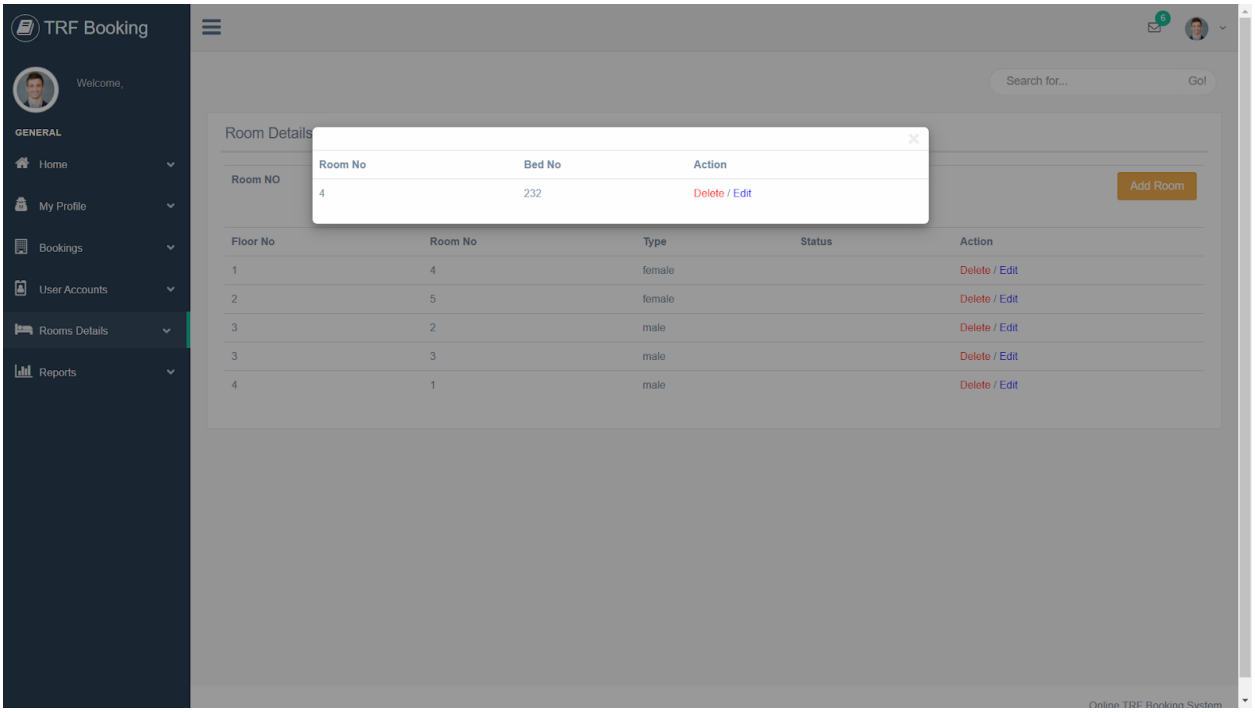


Figure: View Room Details

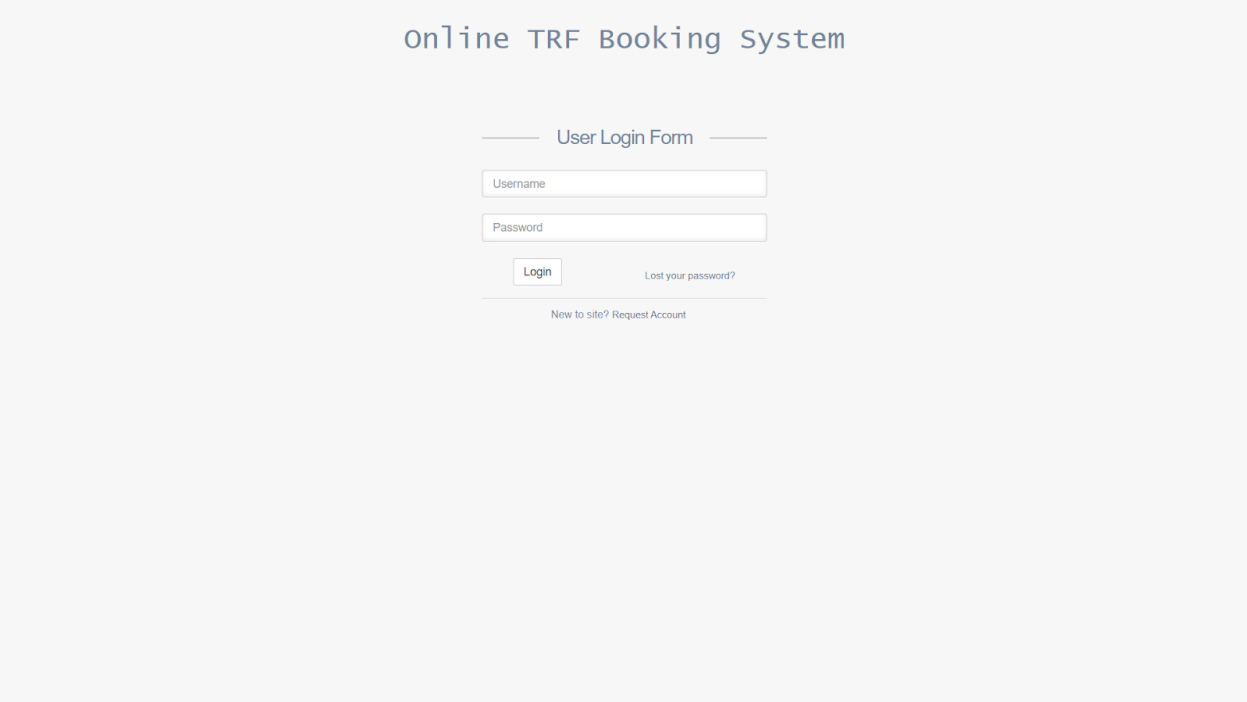


Figure: Login Page

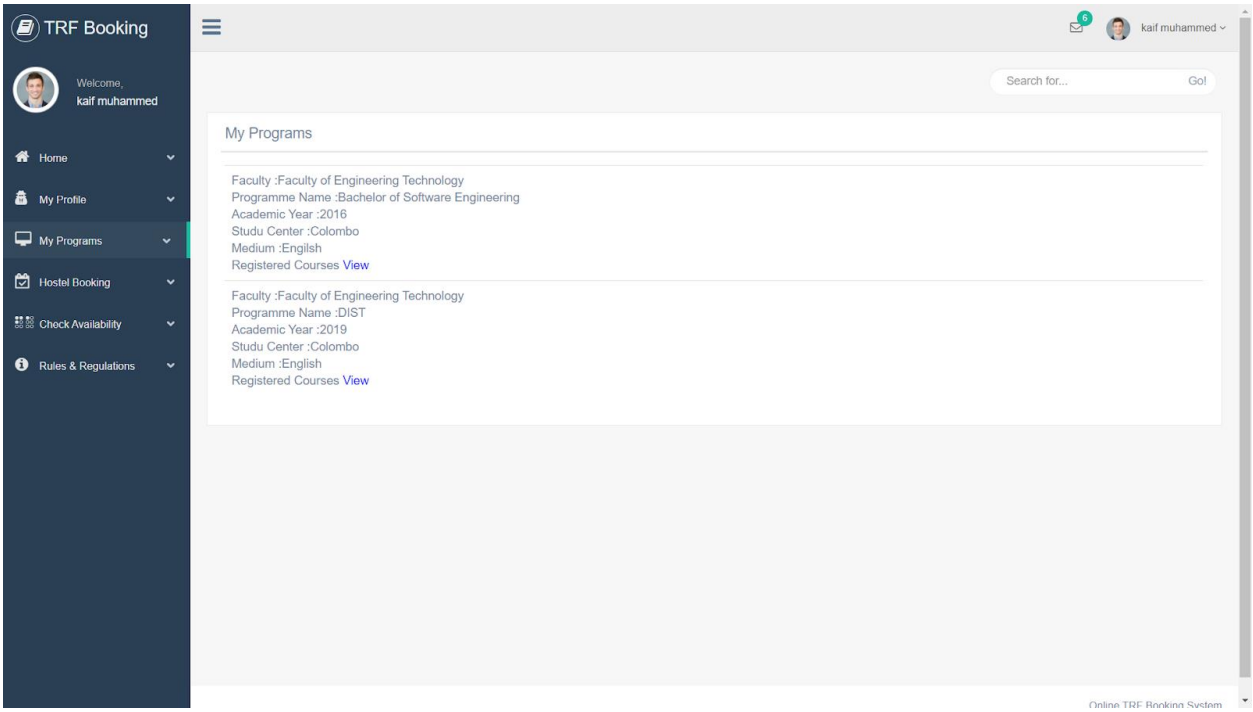


Figure: Student Programs

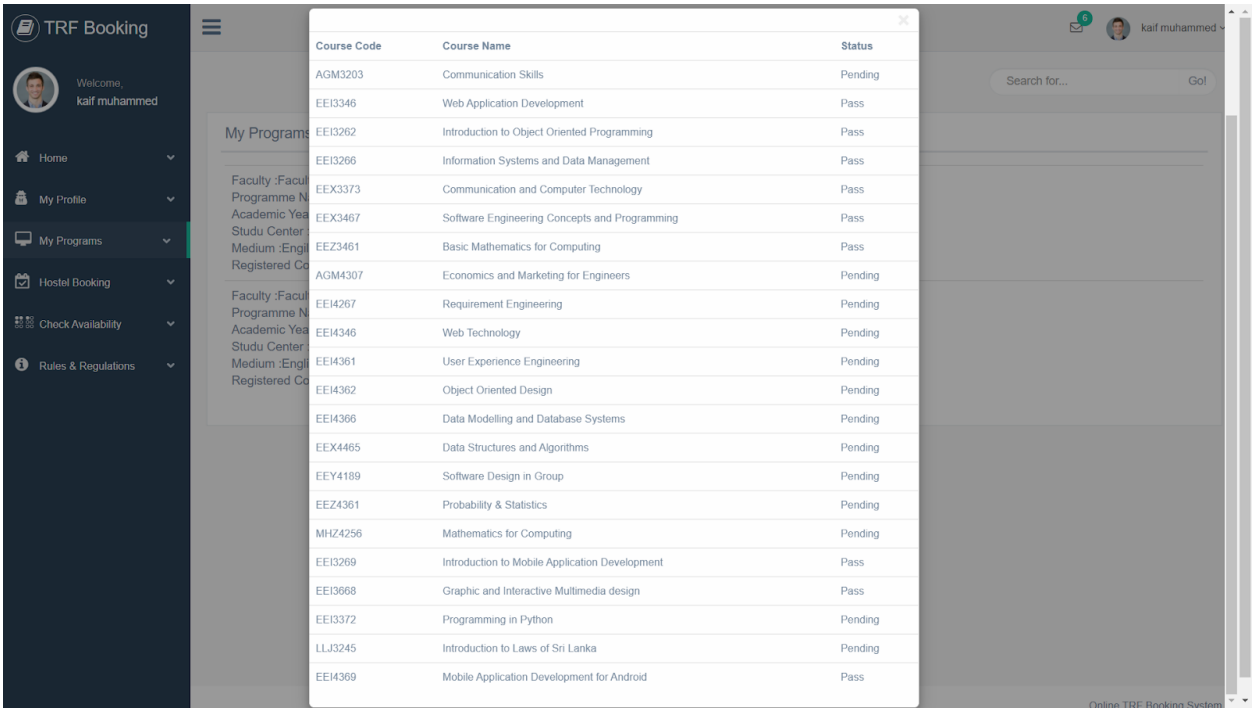


Figure: Student's Course List

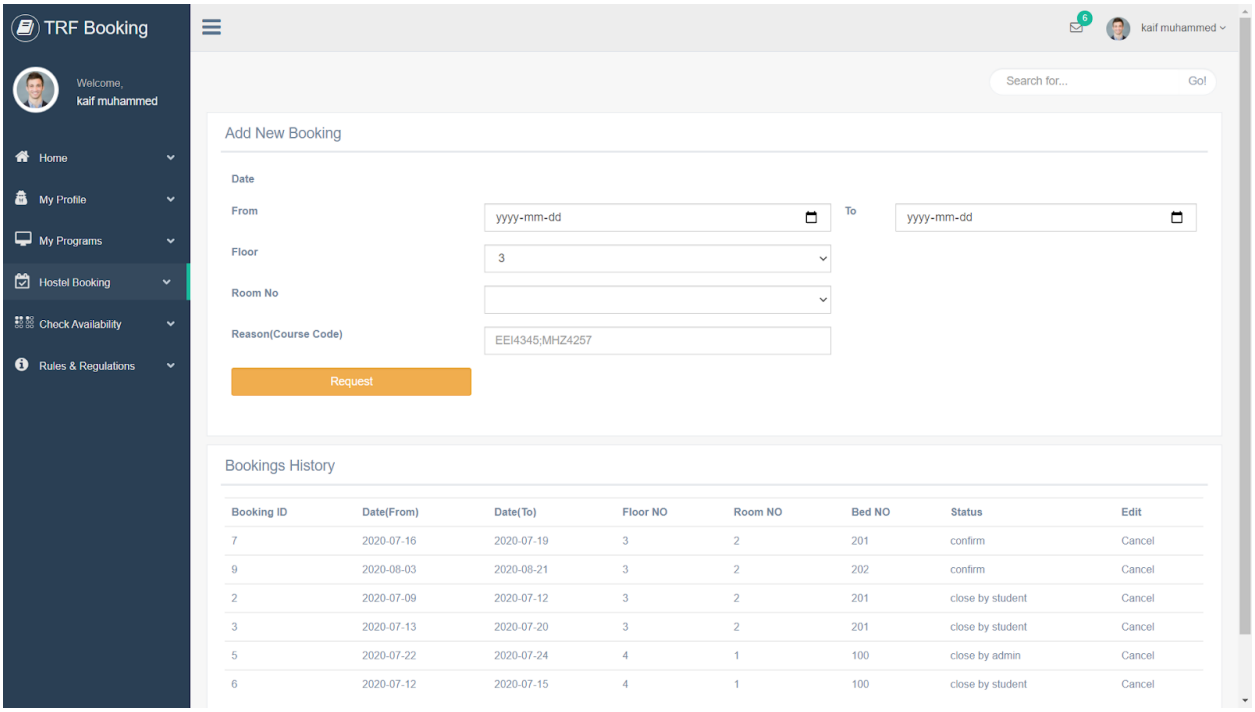


Figure: Add New Booking

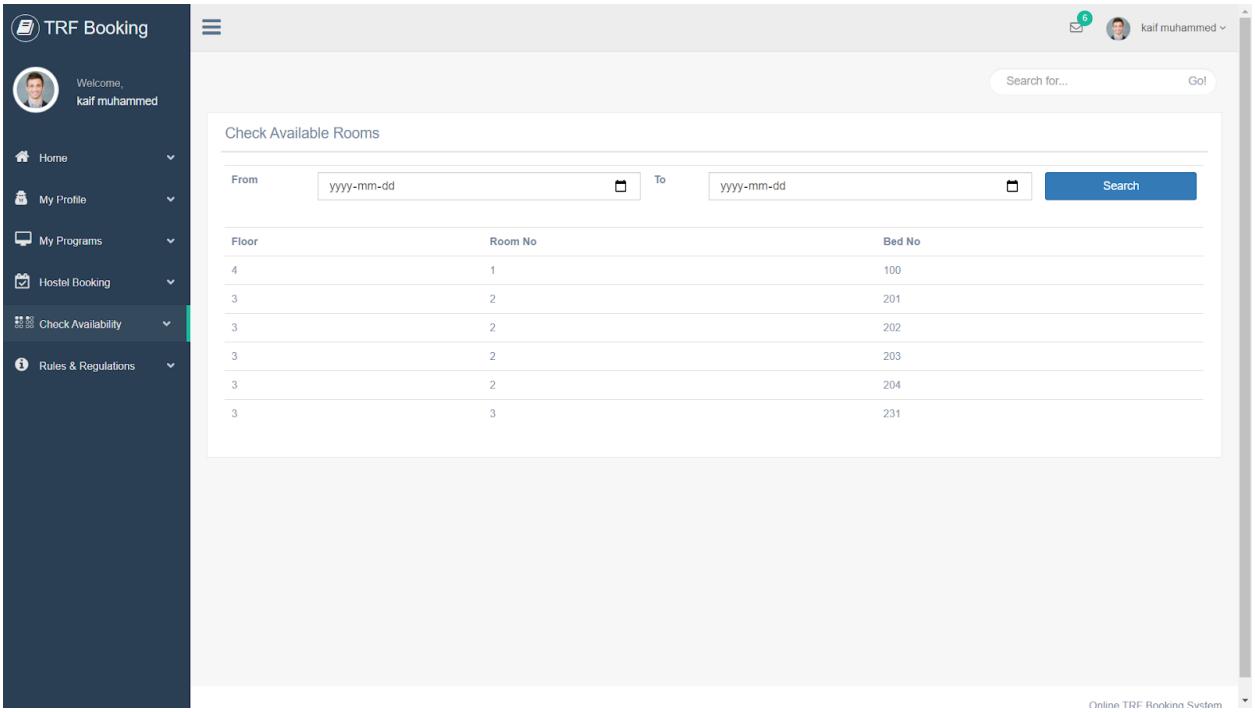


Figure: Check Room Availability

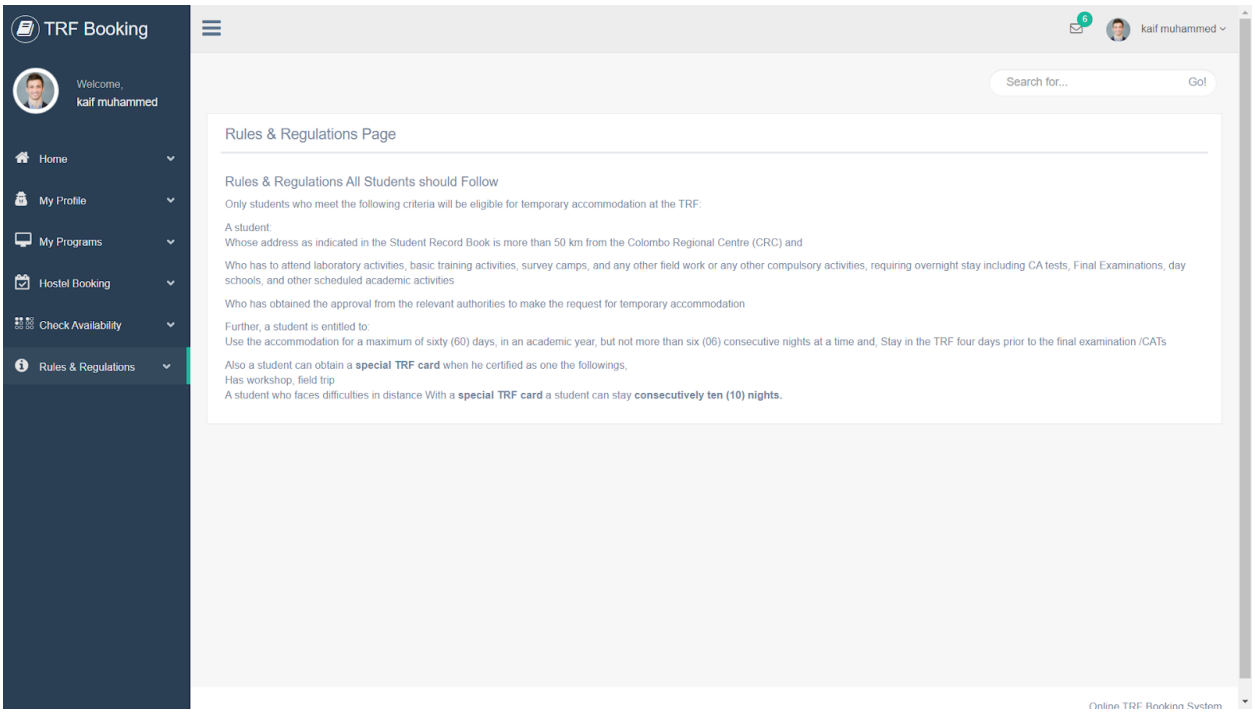


Figure: Rules And Regulation

Data design

4.1. Data description MySQL database and JDBC to communicate with the database that is installed locally on the server.

4.2. Data dictionary Table

1. Data Dictionary

	Field	Type	Null	Key	Default	Extra	
admin							
	aid	varchar	NO	PRI	NULL		
	fname	varchar	NO		NULL		
	lname	varchar	NO		NULL		
	nic	varchar	NO		NULL		
	email	varchar	NO		NULL		
	telNo	int	NO		NULL		
	address	varchar	NO		NULL		
	gender	varchar	NO		NULL		
booking							
	bookingid	int	NO	PRI	NULL	auto_increment	
	roomid	int	NO		NULL		
	dfrom	date	NO		NULL		
	dto	date	NO		NULL		
	reason	varchar	NO		NULL		
	status	varchar	NO		pending		
	sid	varchar	NO		NULL		
course							
	course_code	varchar	NO	PRI	NULL		
	course_name	varchar	NO		NULL		
	credit	int	NO		NULL		
	Category	varchar	NO		NULL		
	level	int	NO		NULL		
dailylog							
	id	int	NO	PRI	NULL	auto_increment	
	Date	Date	NO		NULL		
	time	time	NO		NULL		
	roomid	int	NO		NULL		
	sid	varchar	NO		NULL		
	performedBy	varchar	NO		NULL		
login							
	id	varchar	NO	PRI	NULL		
	username	varchar	NO	UNI	NULL		
	password	varchar	NO		NULL		
	type	varchar	NO		NULL		

	active	varchar	NO		No		
programme							
	Programme_id	int	NO	PRI	NULL	auto_increment	
	Programme_name	varchar	NO		NULL		
	Specialisation	varchar	NO		NULL		
	curriculum	varchar	NO		NULL		
	faculty	varchar	NO		NULL		
programme_course							
	pc_id	int	NO	PRI	NULL	auto_increment	
	Programme_id	int	NO		NULL		
	course_code	varchar	NO		NULL		
	course_type	varchar	NO		NULL		
	taketoGpa	varchar	NO		NULL		
registration							
	aid	varchar	NO	PRI	NULL		
	fname	varchar	NO		NULL		
	lname	varchar	NO		NULL		
	nic	varchar	NO		NULL		
	email	varchar	NO		NULL		
	telNo	int	NO		NULL		
	address	varchar	NO		NULL		
	gender	varchar	NO		NULL		
rooms							
	roomid	int	NO	PRI	NULL	auto_increment	
	roomNo	varchar	NO		NULL		
	floor	int	NO		NULL		
	bedNo	varchar	NO		NULL		
	comment	varchar	NO		NULL		
	type	varchar	NO		NULL		
student							
	aid	varchar	NO	PRI	NULL		
	fname	varchar	NO		NULL		
	lname	varchar	NO		NULL		
	nic	varchar	NO		NULL		
	email	varchar	NO		NULL		
	telNo	int	NO		NULL		
	address	varchar	NO		NULL		
	gender	varchar	NO		NULL		
student_course							
	id	int	NO	PRI	NULL	auto_increment	
	reg_no	int	NO		NULL		
	course_code	varchar	NO		NULL		
	status	varchar	NO		NULL		

student_programme							
	regno	int	NO	PRI	NULL		
	a_year	int	NO		NULL		
	s_center	varchar	NO		NULL		
	medium	varchar	NO		NULL		
	Sid	varchar	NO		NULL		
	Programme_id	int	NO		NULL		

8. Bibliography

A list of referenced and/or related publications.

Brad Appleton <brad@bradapp.net>

http://www.bradapp.net