[INTRODUCTION 2](#_Toc136198899)

[1.1 OBJECTIVE: 2](#_Toc136198900)

[1.2 SCOPE: 3](#_Toc136198901)

[SYSTEM ANALYSIS 4](#_Toc136198903)

[2.1 FEASIBILITY STUDY: 4](#_Toc136198904)

[2.1.1 Economic Feasibility: 4](#_Toc136198905)

[2.1.2 Technical Feasibility: 4](#_Toc136198906)

[2.1.3 Resources Feasibility: 5](#_Toc136198907)

[2.1.4 Behavioral Feasibility: 5](#_Toc136198908)

[2.1.5 Operational Feasibility: 5](#_Toc136198909)

[2.2 FUNCTIONAL REQUIREMENTS: 5](#_Toc136198910)

[2.3 NON-FUNCTIONAL REQUIREMENTS: 6](#_Toc136198911)

[2.4 REQUIREMENTS SPECIFICATION: 7](#_Toc136198918)

[2.4.1 SOFTWARE REQUIREMENTS: 9](#_Toc136198919)

[2.4.2 HARDWARE REQUIREMENTS: 9](#_Toc136198920)

[2.5 MODULES: 9](#_Toc136198921)

[SYSTEM DESIGN 11](#_Toc136198923)

[3.1 ARCHITECTURE: 11](#_Toc136198924)

[3.2 USE CASE DIAGRAM: 11](#_Toc136198925)

[3.3 SEQUENCE DIAGRAMS: 13](#_Toc136198926)

[3.4 DATABASE DESIGN: 14](#_Toc136198927)

[IMPLEMENTATION 15](#_Toc136198928)

[4.1 FRONT END TECHNOLOGIES 15](#_Toc136198929)

[4.2 BACKEND TECHNOLOGIES 19](#_Toc136198930)

[TESTING 23](#_Toc136198931)

[5.1. PUPPETEER: 23](#_Toc136198932)

[5.2. NUNIT TESTING: 24](#_Toc136198933)

[SCREENS 25](#_Toc136198934)

[USER MANUAL 31](#_Toc136198935)

[CONCLUSION 34](#_Toc136198936)

**CAMERA SERVICE CENTER APPOINTMENT SYSTEM**

**INTRODUCTION**

The Camera Service Center Appointment System helps customers fix their appointment for their camera service. Nowadays, to repair a camera, we need to search for the camera service centers and choose the one that is nearest to us. In order to know the available services, it takes a lot of time and effort. So, in order to overcome these, an online camera service center appointment system is designed. It involves the process of scheduling appointments for camera repairs, maintenance, and related services. It allows users to select the desired service and choose a convenient date and time. It reduces waiting time and provides bills according to the service provided.

The primary purpose of the Camera Service Center Appointment System is to provide a structured framework for customers to request and reserve a time slot for camera-related services, such as repairs, maintenance, or general troubleshooting. By implementing an appointment system, the service center can manage customer flow, allocate resources effectively, and ensure that each customer receives dedicated attention from the technical staff.

**1.1 OBJECTIVE:**

The objective of online camera service centers is to provide a convenient and accessible service for customers. These online service centers typically offer customers the ability to book repair or maintenance services for their cameras and related equipment through a user-friendly web interface. Through these centers, customers can track their repair status, access helpful resources and FAQs, and communicate with customer service representatives. Ultimately, the objective of camera service for booking online centers is to provide a seamless and efficient customer experience, delivering high-quality camera service while minimizing the hassle and inconvenience associated with traditional service centers.

Our camera service appointment system is designed to streamline the process of booking camera repair and maintenance services. We recognize that time is valuable, and waiting in long queues or experiencing prolonged turnaround times can be frustrating. With our online appointment system, you can easily schedule a service appointment at your convenience, saving you time.

**1.2 SCOPE:**

The scope of this project is to help the customers for booking their appointments for the camera service center appointment system is to

**Online appointment booking:** Customers can schedule appointments through a web portal or mobile application.

**Search by place or center name**: We can search the place or center name and it displays the nearest centers to us.

**Slots availability:** If the entire slots booked for the five days, it shows the available slots for the next 5 days

**Consultation and Service:** Once checked-in, customers are directed to the appropriate technician or service representative who will assess their camera and discuss the issues or services required.

**Customer Review:** The customer can review or give feedback about the service that the center provided.

**Reporting and analytics:** The system provides comprehensive reports and analytics on appointment statistics and performance metrics.

The working of the project is to be as follows:

In this, the user can describe the nature of the problem of a product. It displays the list of service centers and their availability. Based on the availability you can select the particular service center and then enter the camera details. Once the service is over the bill has to be generated automatically based on the service provided. It Shows availability of the service center for the next 5 days. It shows the service centers based on the place, district and also display the service center from closest to your location to far away from your location. if the entire slots are filled, then the system should show the next available dates. it maintains a database for customer information, Admin information, Appointment information and Bill information.

**SYSTEM ANALYSIS**

**2.1 FEASIBILITY STUDY:**

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. The key considerations involved in the feasibility analysis are:

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* RESOURCES FEASIBILITY
* BEHAVIOURAL FEASIBILITY
* OPERATIONAL FEASIBILITY

**2.1.1 Economic Feasibility**:

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. The project “Camera service center appointment system” is economically feasible as the technologies used for building the system are free and open sources and also the expenditures are also much less to this project. So, this project gives scope to economic feasibility.

**2.1.2 Technical Feasibility**:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. The project “Camera service center appointment system” is technically feasible as the system developed has high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system has a modest requirement, as only minimal or null changes are required for implementing this system.

**2.1.3 Resources Feasibility**:

Resources that are required for Camera Service Center Appointment System includes:

* + - Programming devices (Laptops)
    - Hosting space
    - Programming tools (freely available)
    - Programming individuals

**2.1.4 Behavioral Feasibility:**

   The system is quite easy to use and learn due to its simple and attractive interface.

**2.1.5 Operational Feasibility:**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the user slowly depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**2.2 FUNCTIONAL REQUIREMENTS:**

Functional requirements are very important system requirements in the system design process. These requirements are the technical specifications, system design parameters and guidelines, data manipulation, data processing, and calculation modules etc., of the proposed system. Functional requirements are in contrast to Non-Functional requirements which are descriptive of the parameters of system performance, quality attributes, reliability and security, cost, constraints in design/implementation, etc.

* Customer registration and profile management.
* Admin display list of services and their availability.
* Perform search operation based on the place districts.
* Appointment booking and scheduling
* Service status tracking and updates.
* Service history tracking for customers.
* Reporting and analytics for administrators.
* Generation of bill and payment option.
* Feedback/reviews for better improvement of application.
* Email integration for intimating new personalized offers to customers
* Multi-factor authentication for the sign-in process.

**2.3 NON-FUNCTIONAL REQUIREMENTS:**

Non-functional requirements tend to be stated in terms of constraints on the results of tasks which are given as functional requirements (e.g.., constraints on the speed or efficiency of a given task), a task based functional requirements statement is a useful skeleton upon which to construct a complete requirements statement. That is the approach taken in this work. It can be helpful to think of non-functional requirements as adverbially related to tasks or functional requirements. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are “constraints”, “quality attributes”, “quality goals”, “quality of service requirements” and “non-behavioral requirements”.

**Usability**: Usability is the degree to which a software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use. “Camera service center appointment system” is able to cope with any number of users who can use the system in different platforms and it provides efficiency and better accuracy.

**Reliability:** This quality attribute specifies how likely the system or its element would run without a failure for a given period of time under predefined conditions. Traditionally, it’s expressed as probability percentage. Reliability is the probability and percentage of the software performing without failure for a specific number of uses or amount of time. The project “Camera service center appointment system” has the high probability of out-casting the results under the predefined conditions.

**Maintainability:** Maintainability is defined as the probability that a system or system element can be repaired in a defined environment within a specified period of time. Increased maintainability implies shorter repair times. “Camera service center appointment system”, is a system that is capable of failures or any repairs within specified time in any environment.

**Scalability**:  The system should be able to accommodate larger volumes of users (whether of users, throughput, data) over time.

**Security**: The system should have to protect user data, including personal information, appointment details.

* + Secure Connection for transmission of any data
  + App Platform -Username Password-Based Credentials
  + Sensitive data has to be categorized and stored in a secure manner

**Performance**: The system should be able to handle a high volume of concurrent users and process appointment requests efficiently.

* + Peak Load Performance (during Festival days, National holidays etc.)
  + Admin application (response time must be less than 2 seconds)
  + Appointment Application (response time must be less than 2 seconds)
  + Admin Application (response time must be less than 2 seconds)

**Availability**: The system should be highly available and accessible to users (99.99%)

**Failover**: Automatically switch to a reliable backup system.

**Logging & Auditing:** The system should support logging(app/web/DB) & auditing at all levels

**Monitoring:** Should be able to monitor via as-is enterprise monitoring tools

**Cloud:** The Solution should be made cloud-ready and should have a minimum impact when   moving away to Cloud infrastructure

**Browser Compatible:**  Compatible with all latest browsers.

**2.4 REQUIREMENTS SPECIFICATION**:

Software requirement specification (SRS):

A requirements specification for a software system – is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development life cycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* Business requirements describe in business terms what must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization.

For instance, process requirements could specify specific methodologies that must be followed, and constraints that the organization must obey. Product and process requirements are closely linked. Process requirements often specify the activities that will be performed to satisfy a product requirement. For example, a maximum development cost requirement (a process requirement) may be imposed to help achieve a maximum sales price requirement (a product requirement); a requirement that the product be maintainable (a Product requirement) often is addressed by imposing requirements to follow a particular development style. In systems engineering, a requirement can be a description of what a system must do, referred to as a Functional Requirement. This type of requirement specifies something that the delivered system must be able to do. Another type of requirement specifies something about the system itself, and how well it performs its functions. Such requirements are often called Non-functional requirements, or 'performance requirements' or 'quality of service requirements. Examples of such requirements include usability, availability, reliability, supportability, testability and maintainability. A collection of requirements defines the characteristics or features of the desired system. A 'good' list of requirements as far as possible avoids saying how the system should implement the requirements, leaving such decisions to the system designer. Specifying how the system should be implemented is called "implementation bias" or "solution engineering". However, implementation constraints on the solution may validly be expressed by the future owner, for example for required interfaces to external systems; for interoperability with other systems; and for commonality (e.g., of user interfaces) with other owned products.

**2.4.1 SOFTWARE REQUIREMENTS:**

       Programming Languages: C#

       Front End: React JS, Bootstrap

       Server Side: ASP .Net Core Web API

       Database: MSSQL

       Tools: vs code, Visual Studio 2022, MSSQL Server Management Studio

Swagger UI, Postman

**2.4.2 HARDWARE REQUIREMENTS:**

Processor: i5/i7

        RAM: 4GB

        Hard Disk: 512GB

**2.5 MODULES:**

1.User Management:

* Registration: Allows users to create an account and provide necessary details.
* Authentication: Verifies user credentials during login.
* User can select nearer service center and can book appointment of their required date and time

2.Appointment Management:

* Booking: Allows users to schedule appointments by selecting preferred date and time slots.
* Cancellation: The user can cancel their booking.
* Confirmation: It display confirmation message to users once the appointment is successfully booked.

3. Service Centers List:

* This module provides list of Service centers closest to your location
* It also includes details like Address and pricing, timings.

4.Calendar and Scheduling:

* Calendar Integration: Integrates with a calendar system to manage and display available time slots.

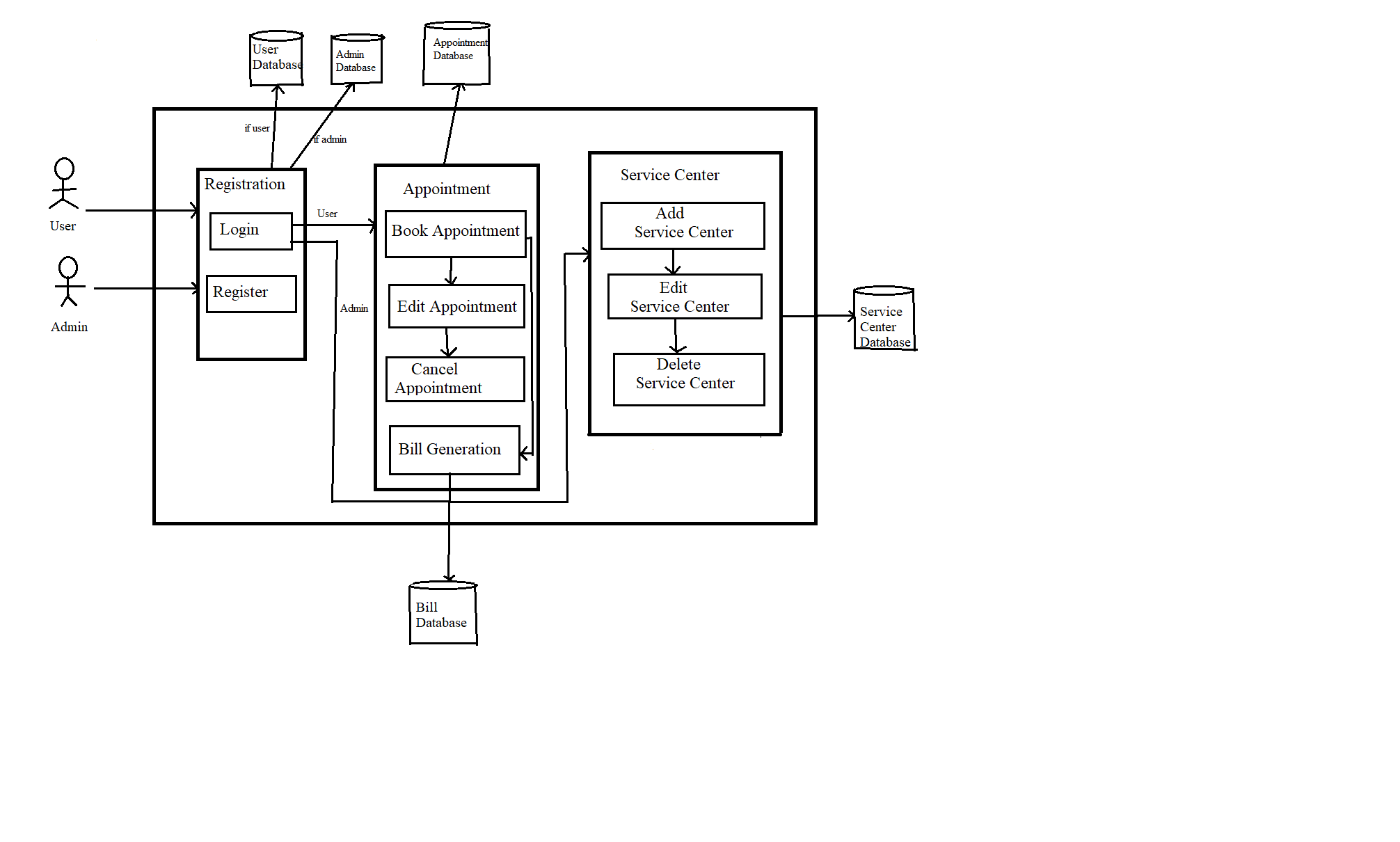
5.Admin Panel:

* Dashboard: Provides an overview of appointment statistics, user data, and other relevant information.
* Service Center Management: Admin can add, edit, delete service centers.
* Reporting: Generates reports and analytics related to appointment history, revenue, and other performance metrics.

# SYSTEM DESIGN

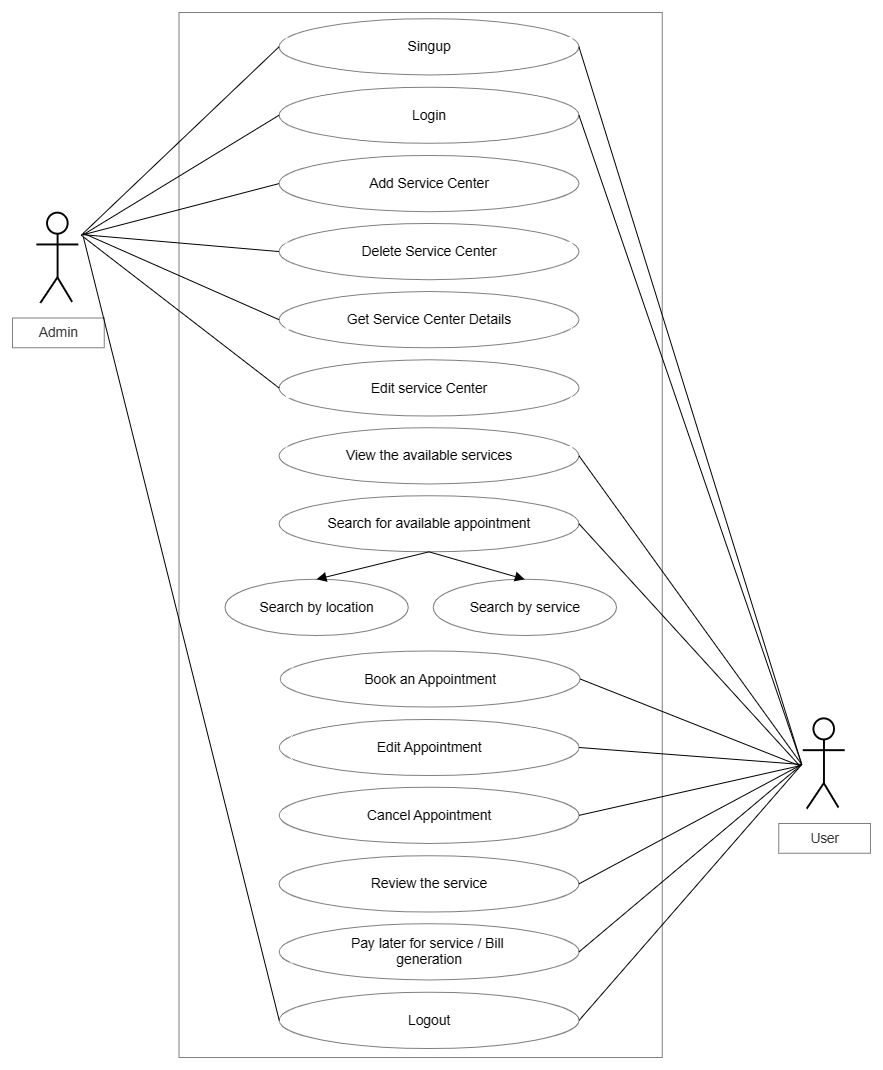
## ****3.1 ARCHITECTURE:****

**The architecture of Camera Service Center Appointment System is as follows:**



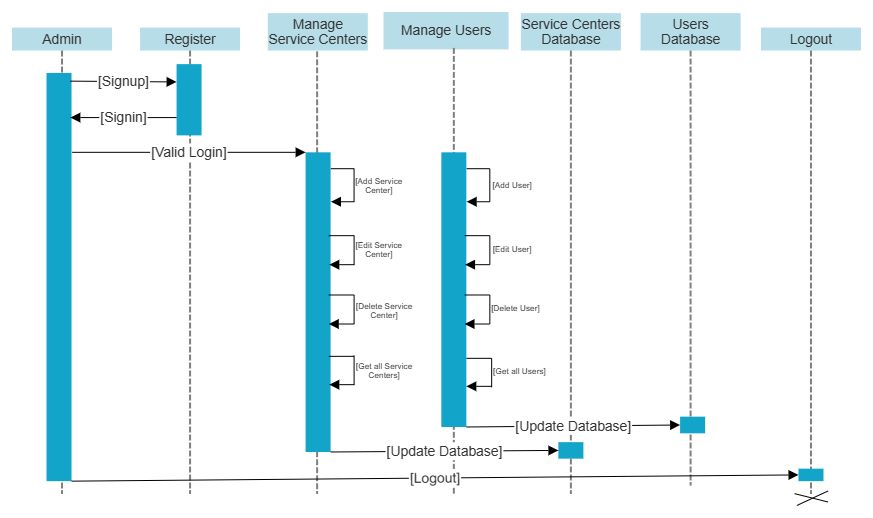
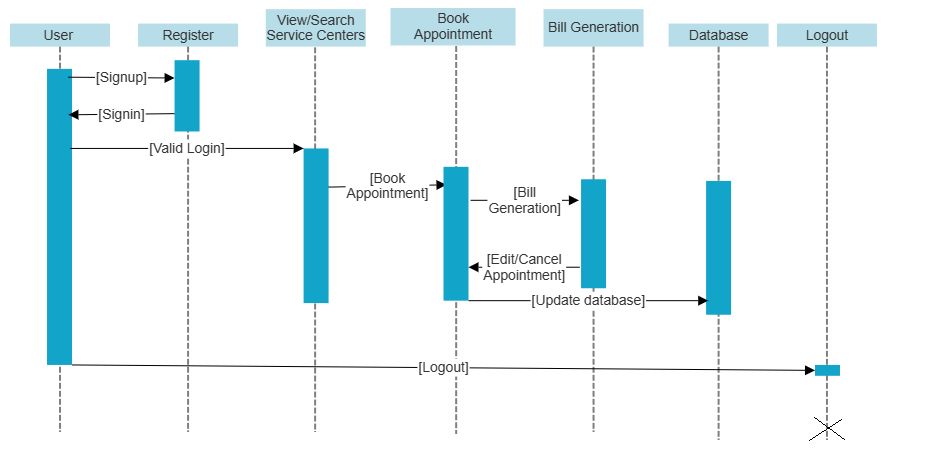
## 3.2 USE CASE DIAGRAM:

**Use Case Diagram** captures the system’s functionality and requirements by using actors and use cases. Use Cases model the services, tasks, function that a system needs to perform. Use cases represent high-level functionalities and how a user will handle the system.



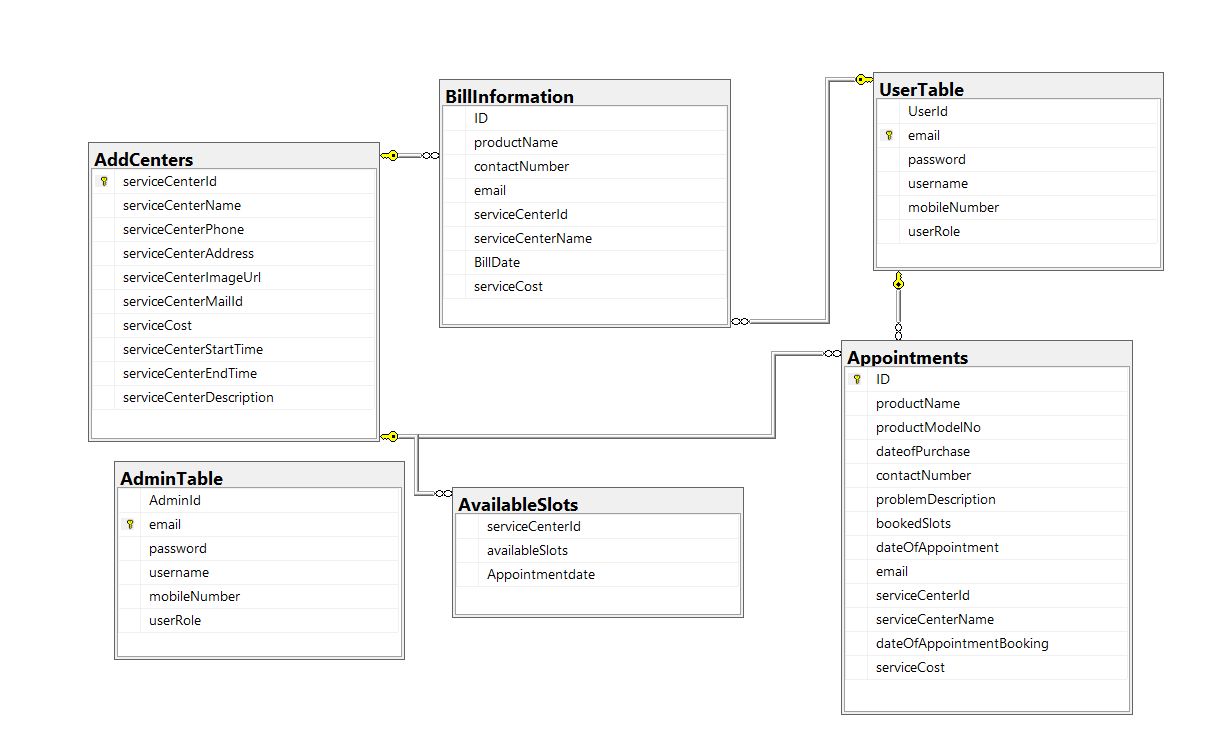
## 3.3 SEQUENCE DIAGRAMS:

Sequence diagrams are a popular dynamic modeling solution in UML because they specifically focus on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.



## 3.4 DATABASE DESIGN:

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. A database management system manages the data accordingly.



**IMPLEMENTATION**

## 4.1 FRONT END TECHNOLOGIES

**REACT JS**

ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based front-end library which is responsible only for the view layer of the application. It was initially developed and maintained by Facebook and later used in its products like WhatsApp & Instagram.

**Why do we use ReactJS?**

The main objective of ReactJS is to develop User Interfaces (UI) that improves the speed of the apps. It uses virtual Document Object Module (JavaScript object), which improves the performance of the app. The JavaScript virtual DOM is faster than the regular DOM. We can use ReactJS on the client and server-side as well as with other frameworks.

**Installation**

To install React.js and set up a development environment, follow these steps:

**Step-1:  Set up Node.js**

* React.js requires Node.js, so start by downloading and installing Node.js from the official website ([https://nodejs.org](https://nodejs.org/))
* Choose the LTS (Long-Term Support) version for stability.

**Step-2:  Verify Node.js installation**

* Open a command prompt or terminal and run the following command to verify that Node.js is installed correctly:

**node -v**

**npm -v**

* You should see the versions of Node.js and npm (Node Package Manager) displayed if the installation was successful.

**Step-3:  Create a React project**

* Run the following command to create a new React project using Create React App:

**npx create-react-app my-react-app**

* Replace "my-react-app" with the desired name for your project.

**Step-4:  Navigate to the project directory**

* Use the following command to change to the project directory:

**cd my-react-app**

* Replace "my-react-app" with your project's name.

**Step-5:**  **Start the development server**

* Run the following command to start the development server:

**npm start**

* This command will start the server and open your React application in a web browser at [http://localhost:3000](http://localhost:3000/).

**Step-6:  Explore and customize**

* Your React project is now set up, and you can start exploring and customizing it.
* Open the project directory in your preferred code editor to modify the source files located in the "src" folder.
* As you make changes to the code, the development server will automatically reload the application in the browser.

**BOOTSTRAP**

Bootstrap is a free and open-source front-end web framework that helps developers to create responsive and mobile-first web applications. It includes HTML, CSS, and JavaScript templates to create web pages with a consistent layout and design. With Bootstrap, developers can easily build responsive and attractive web pages, without having to start from scratch. It is an essential tool for any modern web developer.

**Why do we use Bootstrap?**

It is very easy to use. Anybody having basic knowledge of HTML and CSS can use Bootstrap. It facilitates users to develop a responsive website. It is compatible on most browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.

**What Bootstrap package contains**

**Scaffolding:** Bootstrap provides a basic structure with Grid System, link styles, and background.

**CSS:** Bootstrap comes with the feature of global CSS settings, fundamental HTML elements style and an advanced grid system.

**Components:** Bootstrap contains a lot of reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more.

**JavaScript Plugins:** Bootstrap also contains a lot of custom jQuery plugins. You can easily include them all, or one by one.

**Customize:** Bootstrap components are customizable and you can customize Bootstrap's components, LESS variables, and jQuery plugins to get your own style.

**Installation**

Installing Bootstrap is quite easy and straightforward. Here are the steps you need to follow:

* Go to the official Bootstrap website (<https://getbootstrap.com/>) and click on the "**Download**" button on the homepage.
* You will be directed to a page where you can download the latest version of Bootstrap. You can choose to download either the compiled CSS and JS files, or the source code.
* Once the files have been downloaded, extract them to a folder on your computer.
* Next, you need to link the Bootstrap CSS and JS files to your HTML document. You can do this by adding the following code to the "head" section of your HTML document:

**<link rel="stylesheet" href="path/to/bootstrap.css">**

**<script src="path/to/bootstrap.js"></script>**

* Make sure to replace "path/to/" with the actual path to the CSS and JS files on your computer.
* Finally, you can start using Bootstrap classes and components in your HTML document to create responsive web pages and web applications.

For our project we need some packages like: -

* **npm i react-router-dom:** It is a popular package used in React.js web applications to handle routing/navigation. It allows developers to render different components based on the URL/Route that a user navigates to, without having to reload the page. This package provides a declarative way to handle client-side routing and is commonly used for creating Single Page Applications (SPAs).
* **npm install axios:**  It is a popular JavaScript library used for making HTTP requests from a web browser or Node.js. The npm install axios command is used to install the axios package in your project so that you can use its functions to make HTTP requests to APIs and servers. By using axios, you can easily communicate with web services and retrieve data from them, and then use that data to update your application's state or render it on your UI.
* **npm install react-bootstrap bootstrap:** You might ask why bootstrap is installed in addition to react-bootstrap. The reason is that react-bootstrap doesn't depend on a specific version of Bootstrap and does not include any CSS on its own, but a stylesheet is required to use react-bootstrap components.
* **npm install react-toastify:** This command is used to install a package that provides a simple and customizable notification system for React applications. This package allows you to easily show pop-up notifications to users in response to different actions or events in your application.
* **npm install react-bootstrap-icons:** This command is used to install a package that provides a collection of Bootstrap Icons as React components. This package allows you to easily add Bootstrap icons to your React projects by importing the icon components and using them in your code.

With react-bootstrap-icons, you can save time and effort by not having to search and download individual icon files. Instead, you can access the full collection of Bootstrap Icons as React components and use them with ease.

* **npm install --save or -S:**  When the following command is used with npm install, this will save all your installed core packages into the dependency section in the package.

## 4.2 BACKEND TECHNOLOGIES

**MS SQL**

MS SQL, also known as Microsoft SQL Server, is a relational database management system (RDBMS) developed by Microsoft. It is designed to store and manage structured data, allowing users to interact with databases through SQL (Structured Query Language) queries.

**Installation**

To install Microsoft SQL Server (MS SQL), you can follow these steps:

**Step-1:**  **Download SQL Server**

* Visit the official Microsoft SQL Server Downloads page (<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>).
* Choose the edition of SQL Server that suits your requirements (e.g., SQL Server Express, Developer Edition, Standard Edition, etc.).
* Click the "Download" button to initiate the download.

**Step-2:  Run the SQL Server Installer**

* Locate the downloaded SQL Server installer file and run it.
* The installer will guide you through the installation process.

**Step-3:  Select Installation Type**

* Choose the installation type based on your needs (e.g., New SQL Server stand-alone installation or Adding features to an existing installation).
* Click "Next" to proceed.

**Step-4:  Accept License Terms**

* Read and accept the license terms.
* Click "Next" to continue.

**Step-5:  Choose Features**

* Select the features you want to install. Typically, you would install the Database Engine Services, SQL Server Replication, Full-Text and Semantic Extractions for Search, and Management Tools.
* Click "Next" to proceed.

**Step-6: Specify Instance Configuration**

* Choose whether you want to install a default instance or a named instance.
* Provide a unique name for the instance if you select the named instance option.
* Click "Next" to continue.

**Step-7: Configure Server**

* Set up server authentication mode (Windows Authentication or Mixed Mode).
* Specify SQL Server administrators by adding the appropriate user accounts.
* Click "Next" to proceed.

**Step-8: Choose Database Engine Configuration**

* Select the appropriate authentication mode (Windows or Mixed Mode).
* Specify SQL Server administrators for the database engine.
* Click "Next" to continue.

**Step-9: Specify Installation Location**

* Choose the installation directory for SQL Server.
* Click "Next" to proceed.

**Step-10: Ready to Install**

* Review the installation summary.
* Click "Install" to start the installation process.

**Step-11: Wait for the Installation to Complete**

* The installer will perform the installation process, which may take some time.
* Once the installation is finished, click "Next" and then "Close" to exit the installer.

**ASP.NET**

Core Web API is a framework for building HTTP-based services using the ASP.NET Core framework. It allows you to create lightweight and efficient APIs that can be consumed by various clients, such as web applications, mobile apps, or other services.

**Installation**

To install ASP.NET Core Web API, follow these steps:

**Step-1: Install the .NET Core SDK**

* Visit the official .NET website (https://dotnet.microsoft.com/download) to download the .NET Core SDK.
* Choose the appropriate version for your operating system (Windows, macOS, or Linux).
* Follow the installation instructions provided by the installer.

**Step-2: Verify the .NET Core SDK installation**

* Open a command prompt or terminal.
* Run the following command to verify that the .NET Core SDK is installed correctly:

**dotnet --version**

* You should see the version number displayed if the installation was successful.

**Step-3: Create a new ASP.NET Core Web API project**

* Open a command prompt or terminal.
* Navigate to the directory where you want to create your project.
* Run the following command to create a new ASP.NET Core Web API project:

**dotnet new webapi -n MyWebApi**

* Replace "MyWebApi" with the desired name for your project.

**Step–4: Navigate to the project directory**

* Use the following command to change to the project directory:

**cd MyWebApi**

* Replace "MyWebApi" with your project's name.

**Step-5: Build and run the Web API project:**

* Run the following command to restore dependencies and build the project:

**dotnet build**

* Once the build is complete, run the following command to start the Web API:

**dotnet run**

* The Web API will be hosted on http://localhost:5000 (or a different port if specified).

**Step-6: Test the Web API:**

* Open a web browser or use a tool like Postman to send HTTP requests to the Web API endpoints.
* By default, the project template includes a sample "WeatherForecast" API endpoint that can be accessed at <http://localhost:5000/weatherforecast>.

You can now start building your API by adding controllers, defining routes, implementing business logic, and integrating with data sources as required.

**TESTING**

The purpose is to exercise the different parts of the module code to detect coding errors. The goal of this is to detect designing errors, while focusing the interconnection between modules.

It is to check whether the actual software product matches expected requirements and to ensure that software is error free. The purpose of software testing is to identify errors, gaps or missing requirements

The testing in our project involves these tools:

## 5.1. PUPPETEER:

Puppeteer is a Note.js library which provides a high-level API to control Chrome/Chromium over the DevTools protocol. Puppeteer runs in headless mode by default, but can also be configured to run in full(headful) chrome/chromium. Puppeteer is a Node library which provides a high-level API to control Chromium or Chrome over the DevTools Protocol.

The Puppeteer API is hierarchical and mirrors the browser structure. Puppeteer is a product for browser automation. When installed, it downloads a version of Chromium, which it then drives using puppeteer-core. Being an end-user product, puppeteer supports a bunch of convenient PUPPETEER \* env variables to tweak its behavior.

The only differences between puppeteer-core and puppeteer are:

* Puppeteer-core doesn't automatically download Chromium when installed.
* Puppeteer-core ignores all PUPPETEER\_\* env variables.

Installation:

**npm i puppeteer**

Uses of Puppeteer:

* Things that can do manually in the browser can be done using Puppeteer.
* Generate screenshots and PDFs of pages.
* Crawl a SPA (Single-Page Application) and generate pre-rendered content (server-side rendering)
* Automate form submission, UI test, keyboard input and others
* Create an automated testing environment using the latest JavaScript and browser features.
* Capture a timeline trace of the site to help diagnose performance issues.

## 5.2. NUNIT TESTING:

NUnit is a unit-testing framework for all .Net languages. Initially ported from JUnit, the current production release, version 3, has been completely rewritten with many new features and support for a wide range of .NET platforms.

Developers can write and execute automated tests for their code using the NUnit framework. NUnit is written in C# and can test code in any.NET language, including C#, VB.NET, and F#. The framework includes several assert methods for ensuring that the code under test is working properly. It also has a number of attributes that can be used to control test execution and set up and tear down the test environment. NUnit is a popular tool among developers to ensure that their code is high quality and free of defects.

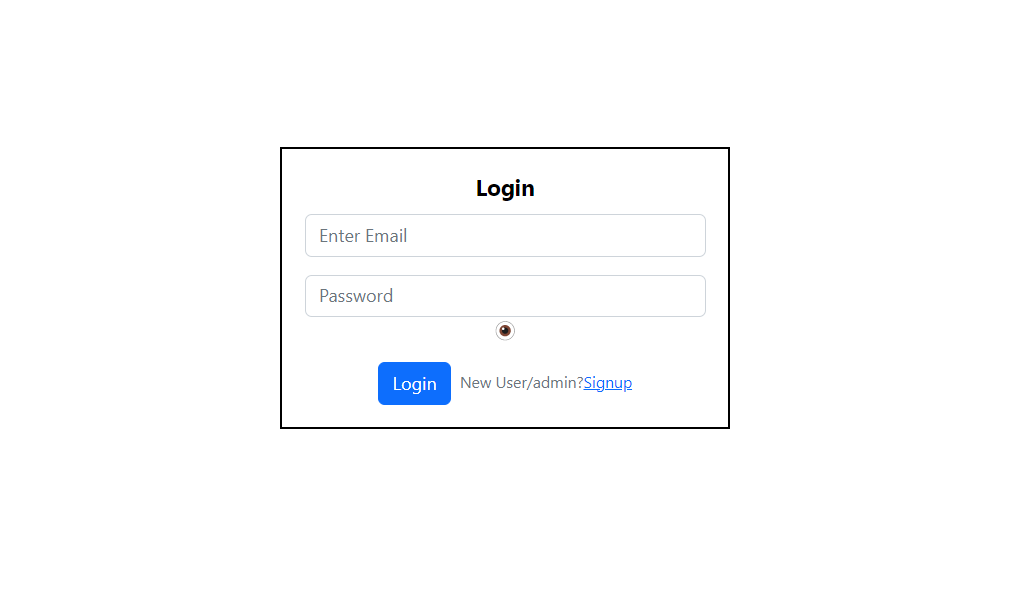
# SCREENS

**1.Login Page**

The screens are available here like:

At first, login page is visible. If the user/admin have an account, they can login otherwise they have to register in the website.

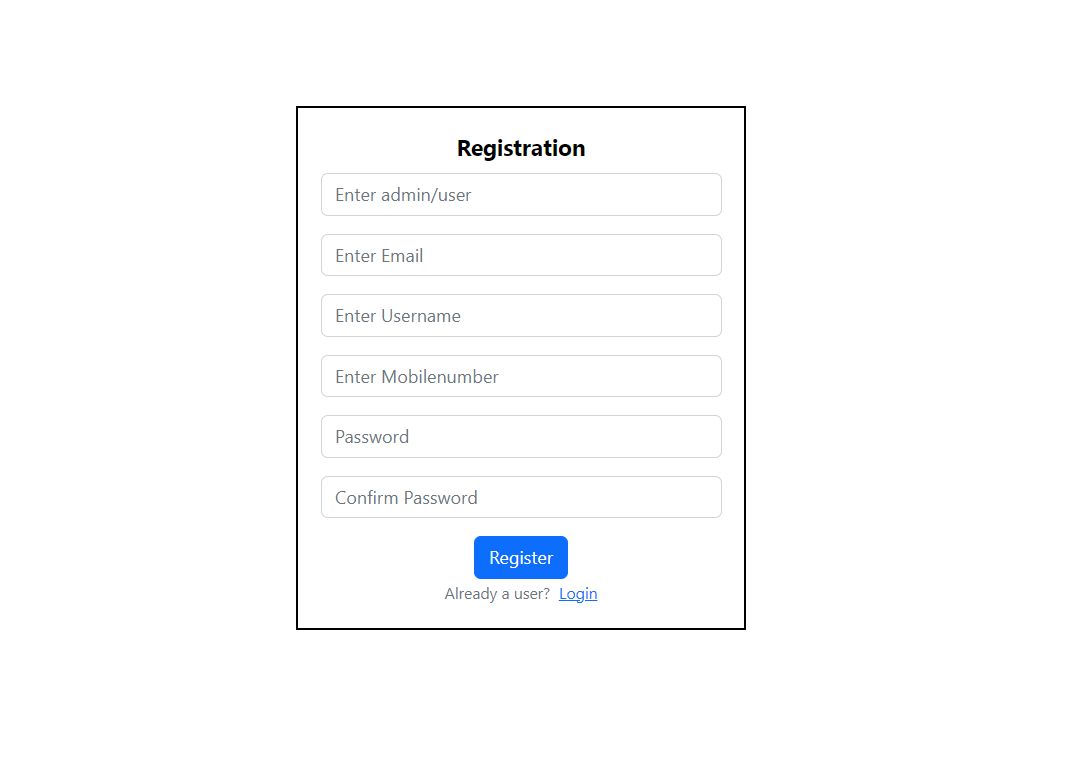
The following url for login page is: http://localhost:3000/s



**2. Register**

If the user wants to register, they have to enter the details and register here.

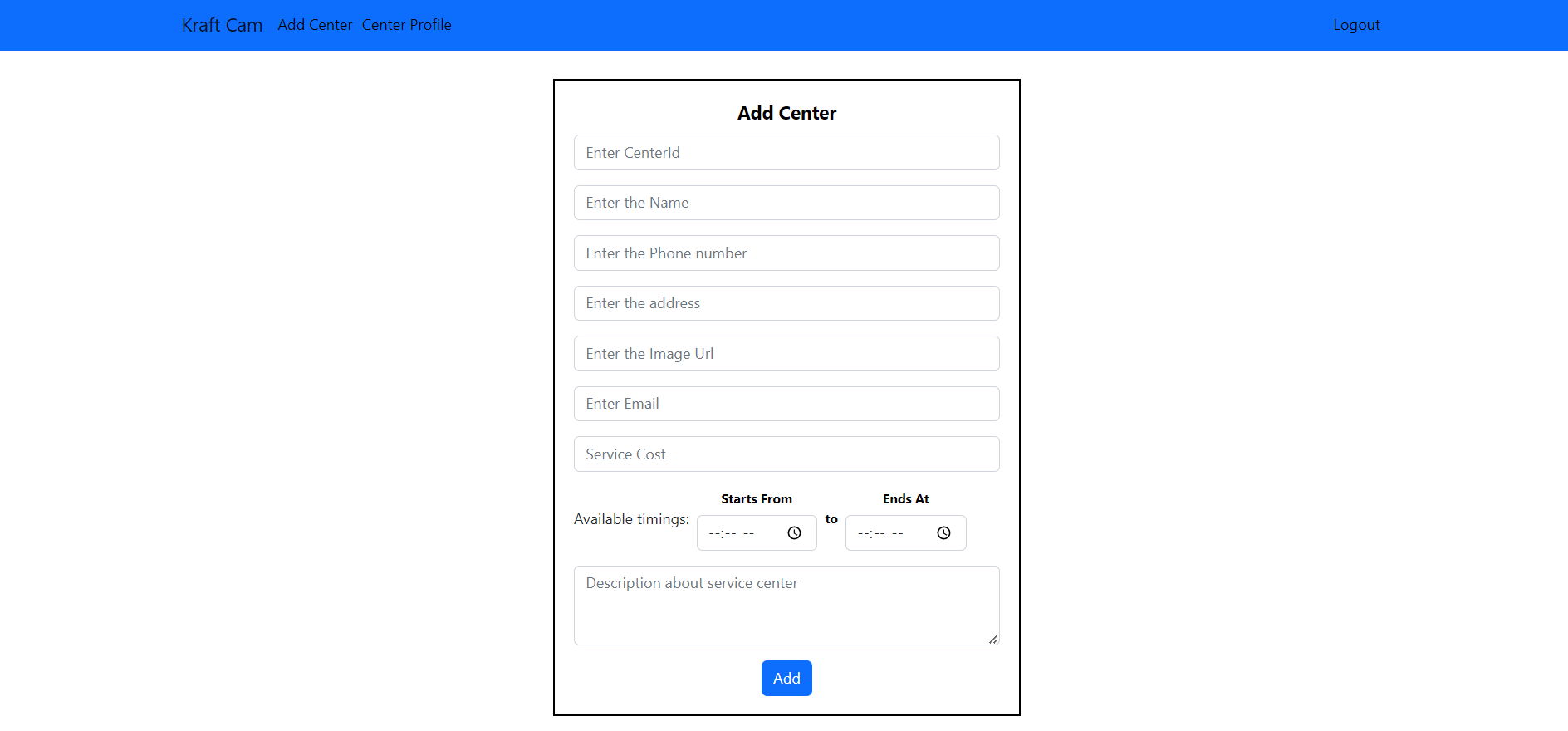
The following url for user will be: http://localhost:3000/signup



**3. Add Service Center**

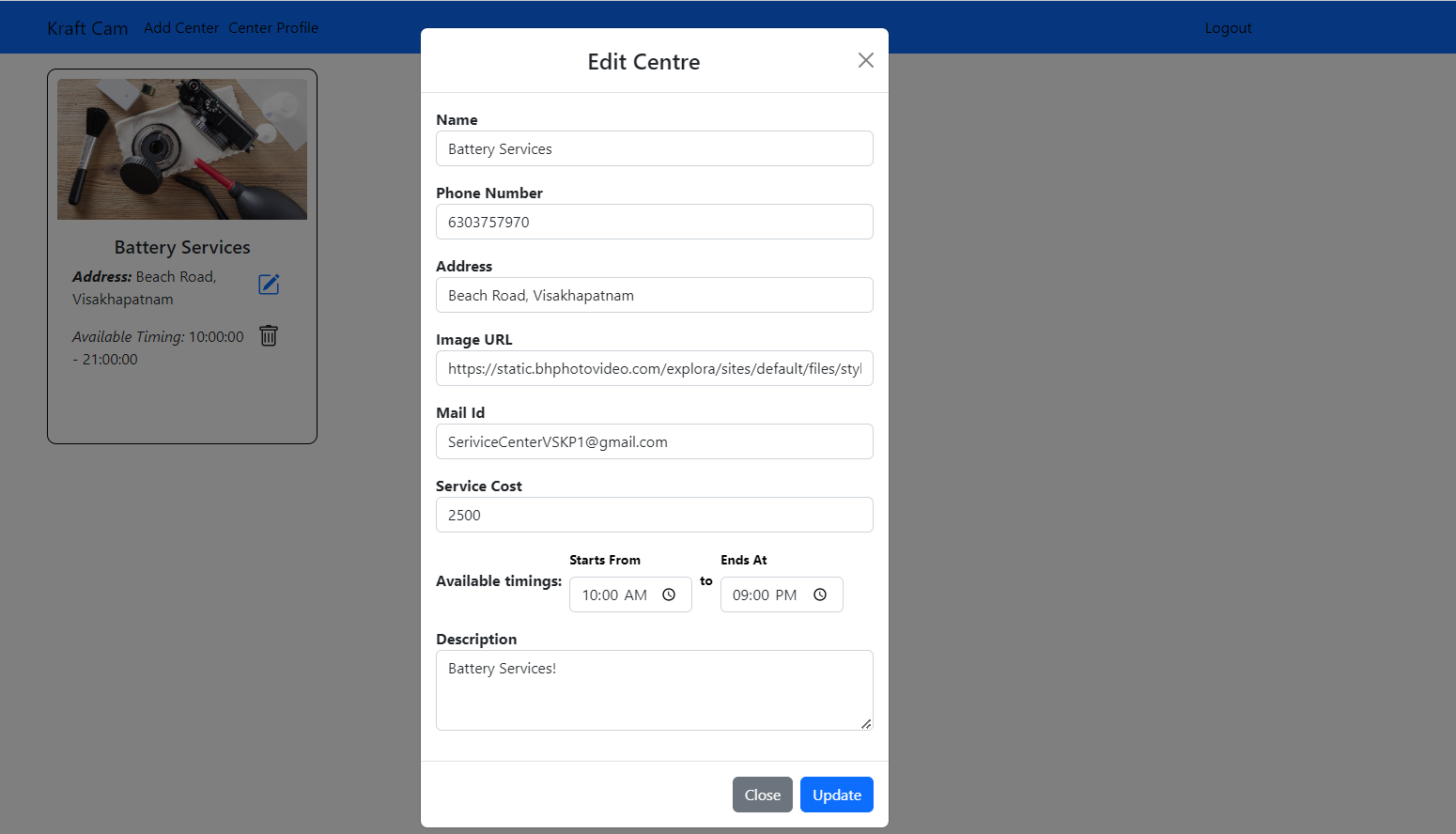
If admin is login into the website, admin have the access to add service centers in the website.

The following url for add service center is http://localhost:3000/admin/addServiceCenter



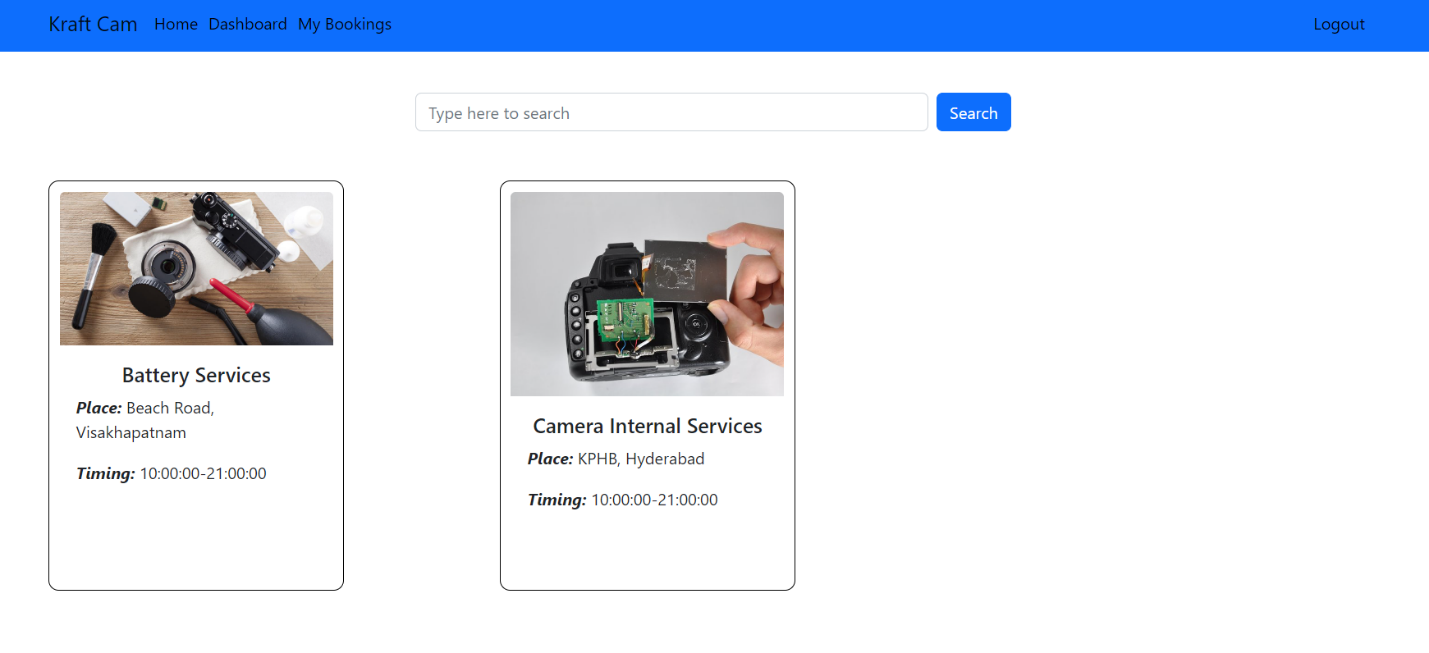
**Edit Service Center**

Admin can edit the service center details and update the details.



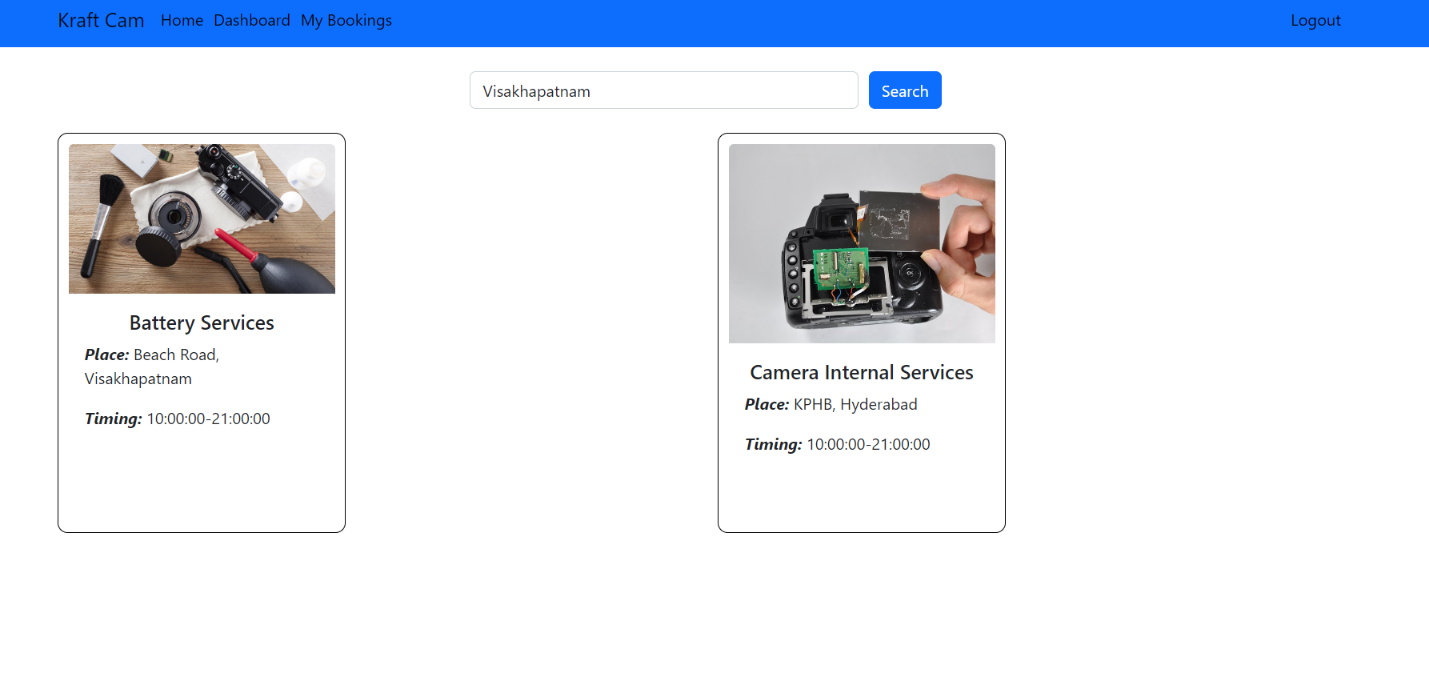
**User Dashboard:**

The dashboard for user will displays all the service centers that are available for repairing the camera issues.

The url for user dashboard will be: http://localhost:3000/user/Dashboard

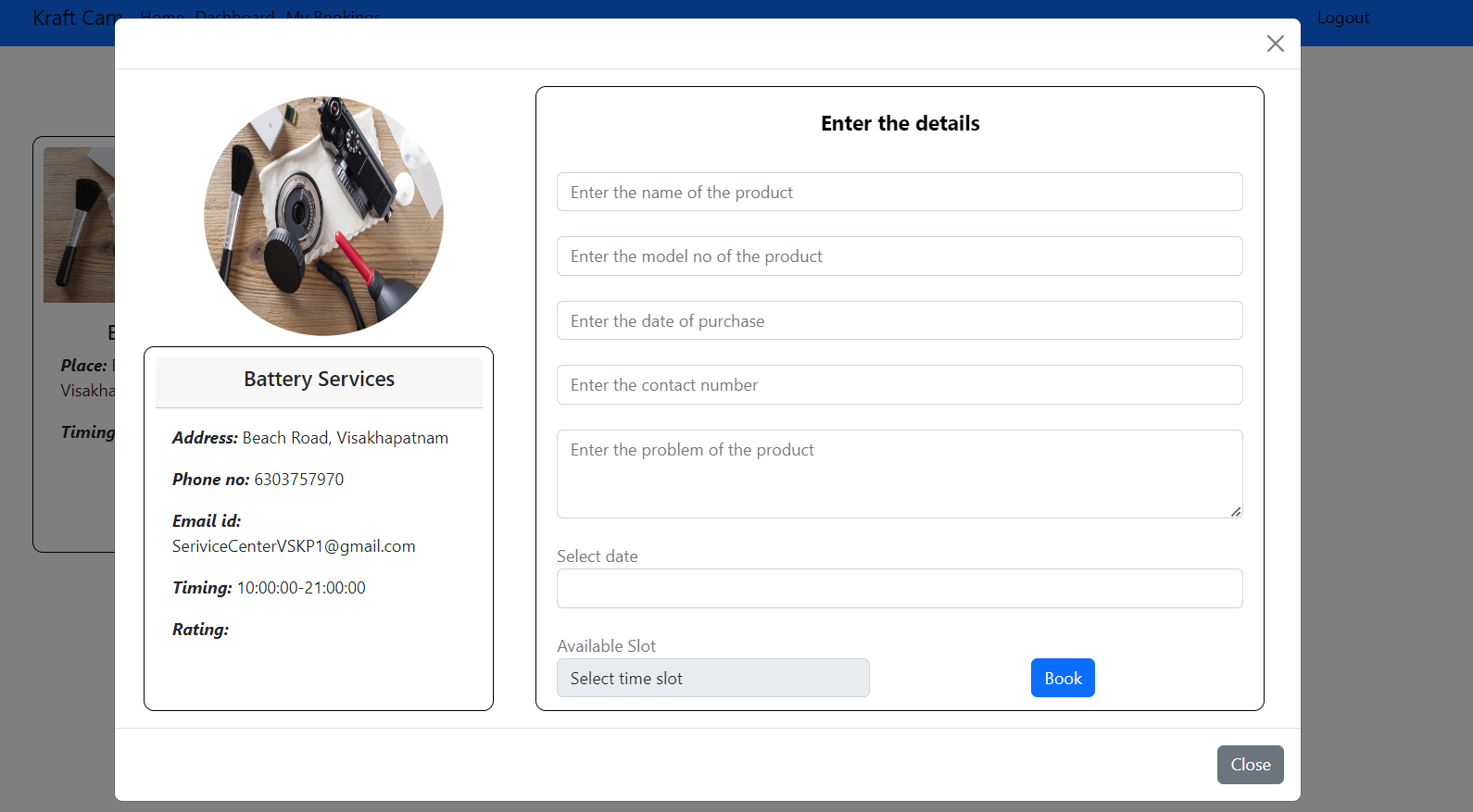
**Search by place and center name:**

If we search the service center name or its location, it displays the available centers nearer to us.



**Edit Appointment**

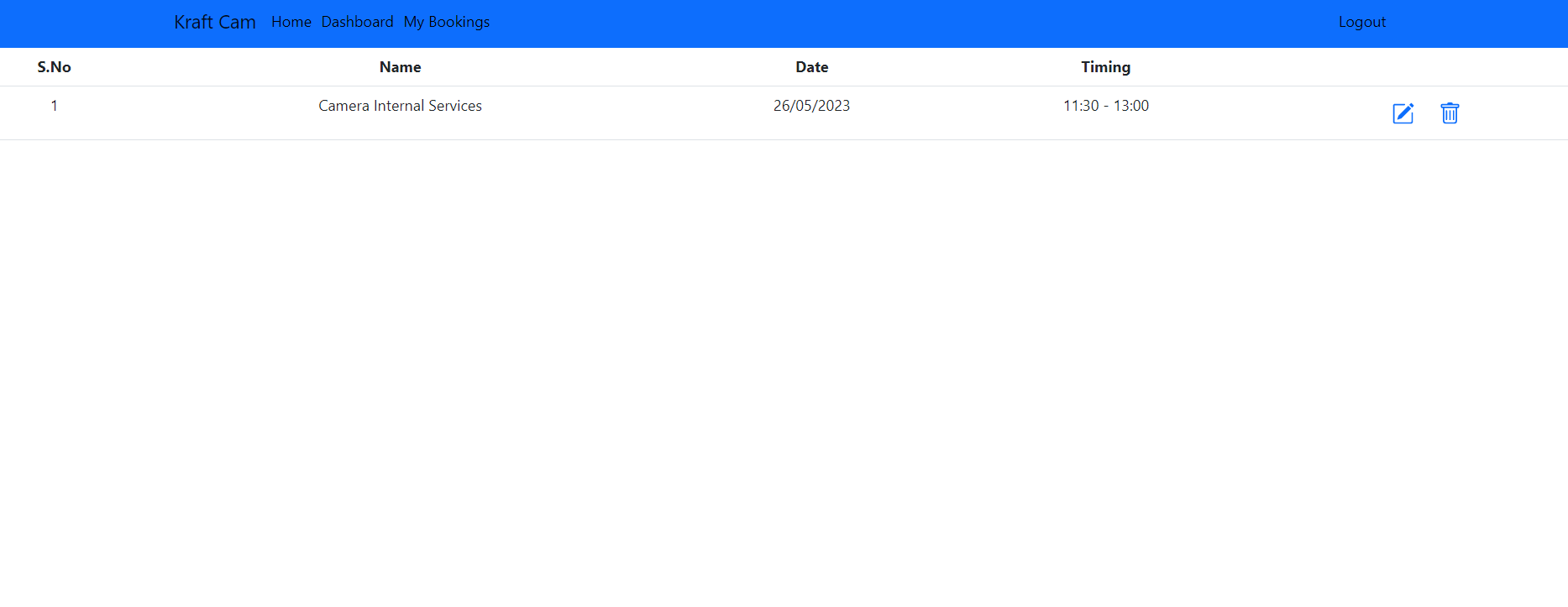
The user can book their appointment for the service. After entering the details of the camera repair, the user can book appointment.



**My Bookings:**

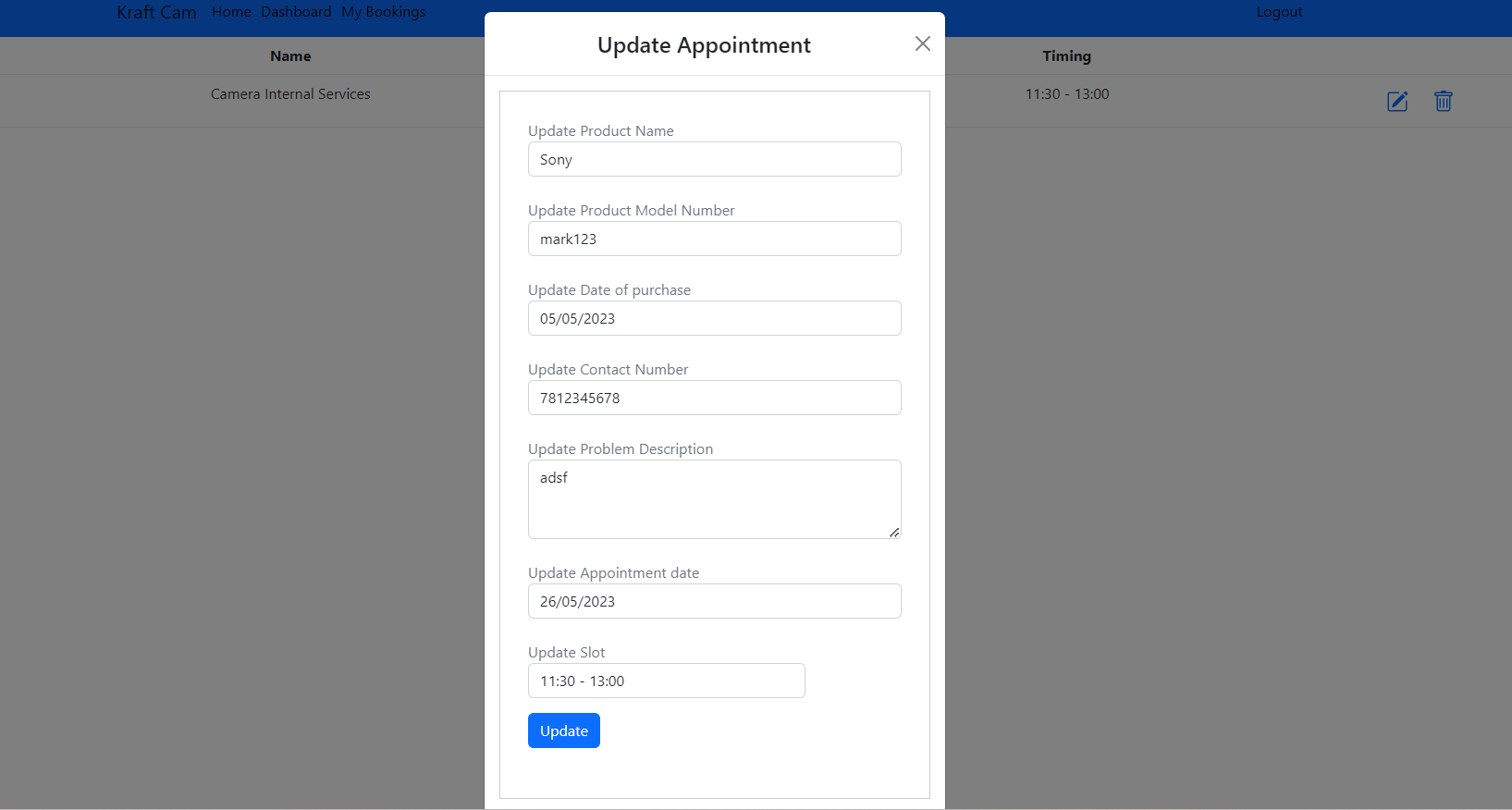
The user will be able to see the bookings in “My Bookings” section

The following url will be: http://localhost:3000/user/Appointment



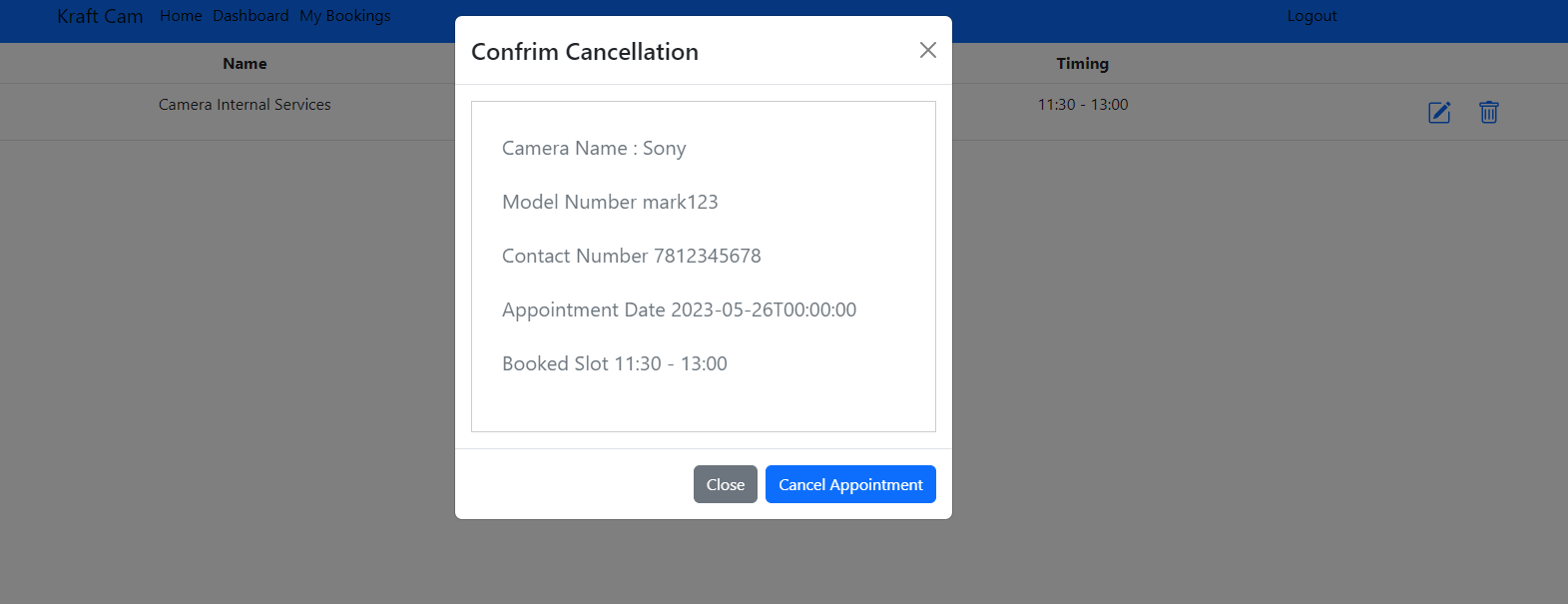
**Update Appointment:**

If the user want to update the appointment like appointment date and slot , they can book here in “My Bookings”



**Cancel Appointment:**

The user can cancel the appointment in the “My Bookings” section



# 

# USER MANUAL

1. Introduction

2. System Access

3. User Registration

4. Login

5. Dashboard

6. Schedule an Appointment

7. Appointment Management

8. Reschedule Appointment

9. Cancel Appointment

10. View Appointment History

1. Introduction

The Camera Service Center Appointment System is designed to streamline the process of scheduling appointments for camera repairs and services. This user manual will guide you through the various features and functions of the system to help you effectively utilize its capabilities.

2. System Access

To access the Camera Service Center Appointment System, follow these steps:

a. Open your preferred web browser.

b. Enter the URL or web address provided by the service center.

c. Press Enter or click "Go" to navigate to the system's login page.

d. Enter your username and password provided by the service center.

e. Click "Login" to access the system.

3. User Registration:

If you are a new user, you may need to register an account before accessing the system. Follow the provided instructions to complete the registration process. This typically involves providing your personal information, such as name, email address, and contact details.

4. Login:

Once registered, use your login credentials (username and password) to access the system. Enter the required information in the appropriate fields and click "Login" to proceed.

5. Dashboard

Upon successful login, you will be directed to the system's dashboard. The dashboard provides an overview of your appointments, service center details, and any pending tasks. Here you can find important notifications and announcements as well.

6. Schedule an Appointment:

To schedule a camera service appointment, follow these steps:

a. Locate and click on the "Book Appointment" or similar button on the dashboard.

b. Select the camera make and model from the provided options.

c. Choose the type of service required, such as repair, cleaning, or maintenance.

d. Specify the preferred date and time for the appointment.

e. Provide any additional notes or details regarding the camera issue.

f. Click "Submit" or "Confirm" to schedule the appointment.

7. Appointment Management

The Camera Service Center Appointment System allows you to manage your appointments efficiently. Here's how:

a. View Upcoming Appointments: Access the “My Bookings" section to see the list of upcoming appointments along with their details.

b. Reschedule Appointment: If you need to change the date or time of an existing appointment, locate the appointment in the list and click on the "Reschedule” . Follow the prompts to select a new date and time.

c. Cancel Appointment: In case you want to cancel an appointment, locate the relevant entry in the appointment list and click on the "Cancel" or similar option. Confirm the cancellation when prompted.

d. View Appointment History: To review your past appointments, navigate to the "My Bookings". Here you can find details of completed appointments, including service provided and any notes or recommendations from the service center.

8. Reschedule Appointment:

To reschedule an existing appointment, follow these steps:

a. Navigate to the "My Bookings" section.

b. Locate the appointment you wish to reschedule.

c. Click on the "Reschedule" option.

d. Select a new date and time for the appointment.

e. Click "Submit" or "Confirm" to finalize the rescheduling.

9. Cancel Appointment:

To cancel an existing appointment, follow these steps:

a. Access the "My Bookings" section.

b. Find the appointment you want to cancel.

c. Click on the "Cancel" option.

d. Confirm the cancellation when prompted.

10. View Appointment History:

To review your past appointments, navigate to the "My Bookings"

# 

# CONCLUSION

In conclusion, the camera service center appointment system provides numerous benefits and advantages for both customers and the service center itself.

Firstly, the appointment system allows customers to schedule their visits to the camera service center in advance, eliminating the need to wait in long queues or spend excessive time at the center. This saves customers valuable time and ensures a smoother and more convenient experience for them.

Moreover, the appointment system helps the service center manage their resources effectively. They can allocate the necessary personnel and equipment based on the scheduled appointments, preventing overcrowding and ensuring that each customer receives adequate attention and service. This results in improved service quality and reduced waiting times.

Additionally, the appointment system enables the service center to maintain better records and track customer interactions more efficiently. They can capture essential customer information during the appointment booking process, allowing them to personalize their services and address specific customer needs. This data can also be utilized for future marketing initiatives and customer relationship management. They can optimize their workflow and staff schedules based on the scheduled appointments, reducing idle time and maximizing productivity. This leads to improved resource utilization and cost-effectiveness for the service center.

Overall, the camera service center appointment system offers a win-win situation for both customers and the service center. It enhances customer satisfaction, reduces waiting times, optimizes resource allocation, and improves overall operational efficiency. Implementing such a system is a valuable investment for any camera service center, providing a seamless and hassle-free experience for customers while improving the service center’s performance.