

**CEN 308 SOFTWARE ENGINEERING**

PROJECT DOCUMENTATION

Hotel Reservation System

Prepared by:

**Haris Hukara**

Proposed to:

**Nermina Durmić, Assist. Prof. Dr.**

**Aldin Kovačević, Teaching Assistant**

Date of submission

24.06.2022

# 1. Introduction

## 1.1. About the Project

This project was meant to be a hotel reservation system where guests could make a reservation with or without account. On the other side we have hotel employees which have access to current reservation requests, so they can preview and admit reservations.

Project link ( <https://hotel-sea.herokuapp.com/index.html#homepage> )

## 1.2. Project Functionalities and Screenshots

Login, registration, forgot password, reset password.

Profile preview and update.

Reservation preview for registered users.

Editing of current reservations for hotel employees.

Rooms preview.

Graphical user interface, website

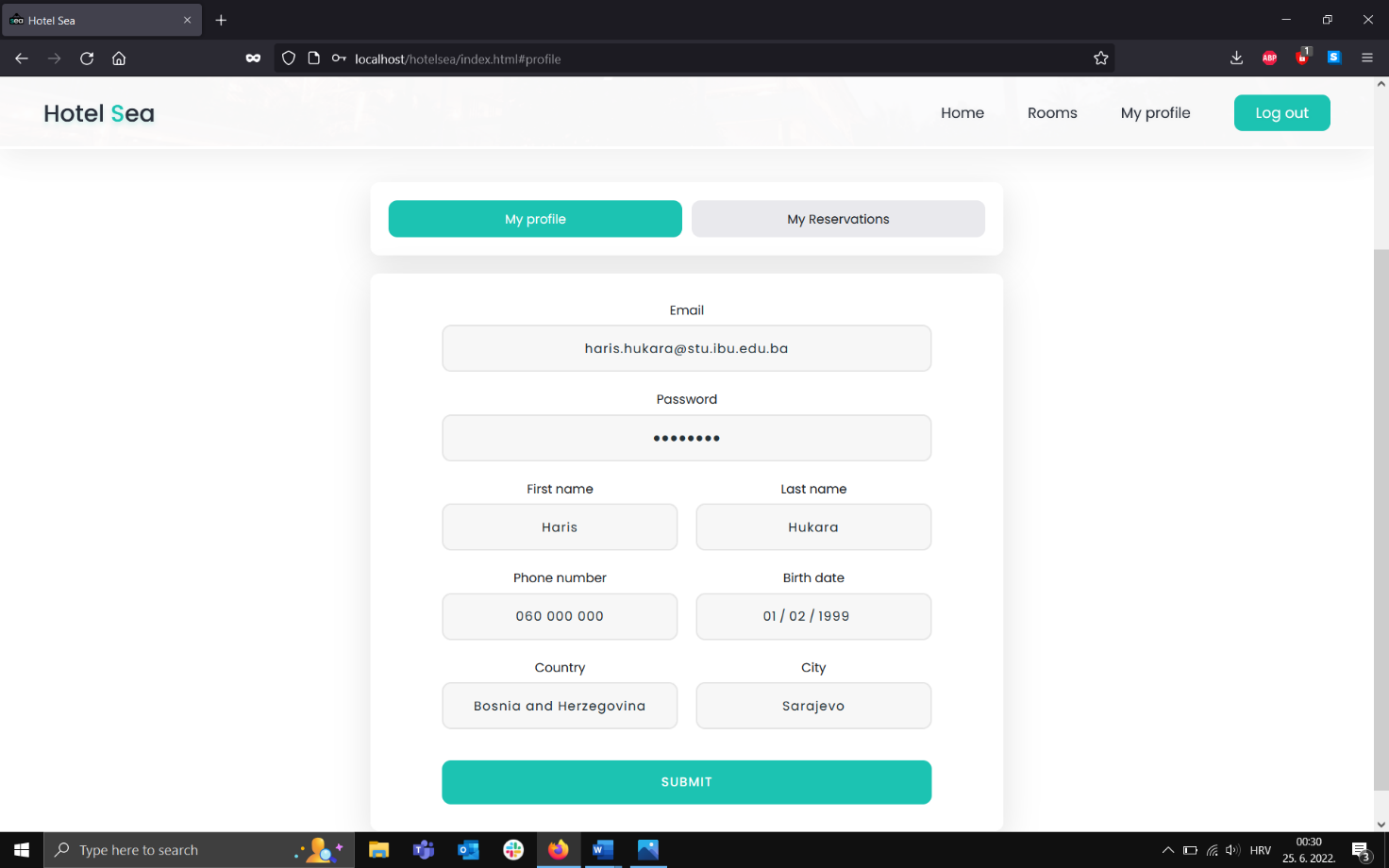
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a bedroom

Description automatically generated with medium confidence



# 2. Project Structure

## 2.1. Technologies

Front end – HTML, CSS, JS, jQuery

Back end – PHP, Flight PHP, MySQL

Documentation - Swagger

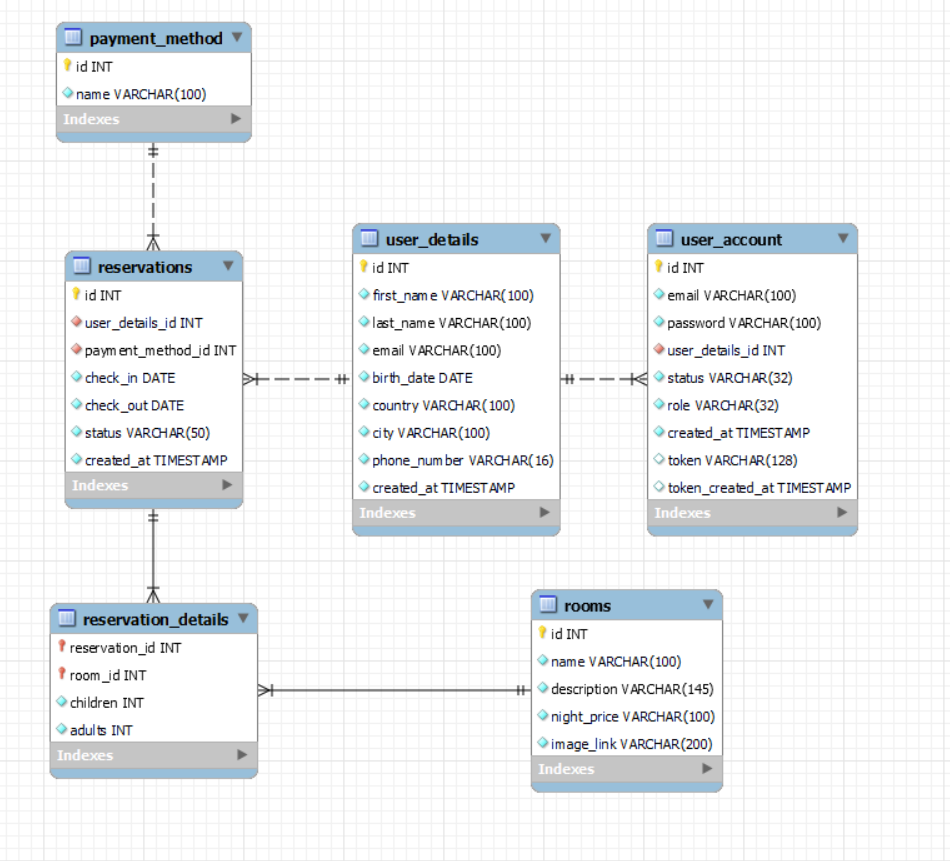
Tools and testing – Postman, C#, RestSharp

RestFull api is implemented.

Frontend is fully responsive.

I tried to keep code as clean as possible and avoided repetitive code. Comments are included only when necessary. I followed PHP convention. Variables are as descriptive as they could be.

## 2.2. Database Entities

**

**User account** ( basic account which is used when user wants to login )

**User details** ( since we don’t want to force user to create account in order to make reservation we need user details, on the other hand registered users can have one user details and use it always for each reservation )

**Reservation** ( table of reservations )

**Reservation details** ( since user can make one reservation of multiple rooms we need details in order to achieve many to many relationship )

**Rooms** ( Basic room info )

**Payment method**

## 2.3. Architectural Pattern

Layered design pattern was used on the backend where we have DB layer, Dao layer, Business Logic One of the powerful features of the layered architecture pattern is the separation of concerns among components. Components within a specific layer deal only with logic that pertains to that layer. For example, components in the presentation layer deal only with presentation logic, whereas components residing in the business layer deal only with business logic. This type of component classification makes it easy to build effective roles and responsibility models into your architecture, and also makes it easy to develop, test, govern, and maintain applications using this architecture pattern due to well-defined component interfaces and limited component scope.

Also I implemented MVC architecture for front end. Different components of the application in MVC can be independently deployed and maintained.

# 3. Conclusion

*To be honest this was my first time dealing with Front end stuff and it was very challenging. It took me so long to learn about it and to build UI. Since I did this project on my own and had a lot less time since I started working, it was very difficult to achieve everything I imagined. If I had more time or a teammate much more work could have been done. In the end I implemented most important stuff regarding project architecture and I learned a lot. Maybe I took more than I could chew, but I will keep working on this project, hopefully all functionalities will be implemented in upcoming months.*