HOTEL RESERVATION SYSTEM

-Team 11

Team Members

Chakradhar Reddy Donuri (E949F496)

Tilak Gurappagari (S829K279)

Anirudh Reddy Singireddy (P727R596)

Ram Charit Vyas (V249P387)

Akshay Thaduri (K769S596)

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope of the Project 5

1.3 Definition, Acronyms and Abbreviations 5

1.4 References 4

1.5 Overview.. 6

2 Overall Description 7

2.1 Product Perspective 7

2.2 Product Functions 7

2.3 User Characteristics.. 9

2.4 Constraints 9

2.5 Assumption and Dependencies 10

3 Specific Requirement.. 11

3.1 Functional Requirement 11

3.2 Non-Functional Requirements 11

3.2.1 Safety Requirements 11

3.2.2 Security Requirements 11

3.2.3 Other Requirements 12

3.3 External Interface 12

3.3.1 User Interface 12

3.3.2 Hardware Interfaces 15

3.3.2 Software Interfaces 15

3.3.3 Database Interface 16

3.3.4 Communications Interfaces 16

3.4 Performance Requirement 16

3.5 Logical Database Requirements.. 17

3.6 Attributes 18

3.6.1 Reliability.. 18

3.6.2 Availability 18

3.6.3 Security 18

3.6.4 Maintainability.. 18

3.6.5 Portability.. 18

4 Design And Analysis 19

4.1 Use case 19

4.2 Class Diagram 20

4.3 Sequence Diagram …………………………………………………………………………………… 21

4.4 Dataflow Diagram …………………………………………………………………………………. 22

5 Implementation ……………………………………………………………………………..……. 23

6 Testing………………………………………………………………………………………………….. 24

1 Introduction

This is the final report document for developed hotel reservation system. It consists of the milestones in development of finalized hotel reservation system. Iterative waterfall method was used as the software development life cycle. Coding was handled through an Object-oriented approach. Above mentioned methodologies made project work load light and provided the ease of developing. The system was evaluated by several people regarding user levels of the developed system. Results of the evaluation helped for further maintenance of the product. Fully functional Hotel Reservation System will fulfil the main objectives and all the events of the hotel.

1.1 Purpose

The Software Requirements Specification (SRS) will provide a detailed description of the requirements for the Hotel Reservation System (HRS). This SRS will allow for a complete understanding of what is to be expected from the newly introduced system which is to be constructed. The clear understanding of the system and its functionality will allow for the correct software to be developed for the end user and will be used for the development of the future stages of the project. This SRS will provide the foundation for the project. From this SRS, the Hotel Reservation System can be designed, constructed, and finally tested.

This SRS will be used by the system development team which is constructing the HRS and the hotel end users. The Project team will use the SRS to fully understand the expectations of this HRS to construct the appropriate software. The hotel end users will be able to use this SRS as a "test" to see if the constructing team will be constructing the system to their expectations. If it is not to their expectations the end users can specify how it is not to their liking and the team will change the SRS to fit the end users' needs.

1.2 Scope of the Project

The introducing software, Hotel Reservation System which is going to be implemented for Westin hotel will automate the major operations of the hotel. The Reservation System is to keep track in room reservation and check availability. The Room Reservation System is to manage all room types and room services. There are two End Users for HRS. The End Users Are Admin and Customer. Admin can access to all system functionalities without any restrictions. Customer can only access to the Reservation section. To keep restrictions for each End User levels HRS can create different Login functions.

The objectives of the automated Hotel Reservation System is to simplify the day to day processes of the hotel. The system will be able to handle many services to take care of all customers in a quick manner. As a solution to the large amount of file handling happening at the hotel, this software will be used to overcome those drawbacks. Safety, ease of using and most importantly the efficiency of information retrieval is, some benefits the development team going to present with this system. The system should be user appropriate, easy to use, provide easy recovery of errors and have an overall end user high subjective satisfaction.

1.3 Definition, Acronyms and Abbreviations

H.R.S. – Hotel Reservation System

SRS – Software Requirements Specification

End users – The people who will be using the system

1.4 References

Internet

https://reactjs.org/docs/create-a-new-react-app.html

http://react-icons.github.io/react-icons/

https://material-ui.com/

https://www.mongodb.com/

https://dashboard.emailjs.com/

1.5 Overview

This SRS is organized into two parts the first is the overall description and the second section is the specific requirement.

The overall description will describe the requirement of Hotel Reservation System.

The specific requirement section describes the detail of the system.

2 Overall Description

2.1 Product Perspective

The Hotel Reservation System is a new self-contained software product which will be produced by the project team in order to overcome the problems that have occurred due to the current manual system. The newly introduced system will provide an easy access to the system and it will contain user friendly functions with attractive interfaces. The system will give better options for the problem of handling large scale of physical file system, for the errors occurring in calculations and all the other required tasks that has been specified by the client. The final outcome of this project will increase the efficiency of almost all the tasks done at the Hotel in a much convenient manner.

2.2 Product Functions

- Register Customer
- Make Reservations
- Search Rooms
- Add Additional Facilities (Jacuzzi, Swimming pool, Meals, Additional beds)
- Make Payment
- Manage Room Details
- Generate and e-mail Receipt

Functional Requirements

Function 1	**Register Customer**
Input	Customer Username, Contact, Company, Name, E-mail, Gender, Address
Output	Database Record, Database successfully updated pop-up
Workflow	Validate the given details and record the information in to the database.

Function 2	**Make Reservations**
Input	Username, Total child's, Total Adults, check-in date, check out date, status, Number of nights.
Output	Database Record, Database successfully updated pop-up
Workflow	Validate the given details and record the information in to the database.

Function 3	**Search Rooms**
Input	Period, Check-in, Check-out, Guest
Output	Display a pop-up with available room details
Workflow	Validate the given details and check for the available rooms in a given time period and return its availability.

Function 4	**Add additional facilities**
Input	Period, Check-in, Check-out, Guest
Output	Display a pop-up with available facilities details
Workflow	Validate the given details and check for the available rooms in a given time period and return its availability.

Function 5	**Make Payment**
Input	Total,1 pay time, Credit card details
Output	Database Record, Database successfully updated pop-up
Workflow	Validate the given details and record the information in to the database.

Function 6	**Generate and E-mail Receipt**
Input	Billing no, Quantity, Price, Taxes, Date, Services, Unit
Output	Printed version of the bill
Workflow	Validate the given details and total cost is calculated according to the Services gain by the customer.

2.3 User Characteristics

2.3.1 Admin

Admin can view all the reservations made by the user. He can check the rooms availability

2.3.2 Customer

New Customer can register in to the website, if already registered customers can login to the website and perform the following functionalities. Customers can search for rooms within the selected check-in and checkout dates. The customer pays the total cost at the payment gateway, the total cost is calculated for the selected room type, number of guests and extra facilities selected.

2.4 Constraints

Software development crew provides their best effort in developing the system. In order to maintain the reliability and durability of system, some design and implementation constraints are applied. Availability of an android app for hotel Reservation system could make the system portable but due to time constraint it is not possible. System will need a minimum memory of 512MB. But it is recommended to have a memory of 1GB. Considering the client's budget, we decided to create those interfaces in a simple realistic manner using affordable technology.

2.4.1 Hardware

1. Operating System Supports all known operating systems, such as Windows, Linux
2. Computer 512MB+ RAM, monitor with minimum resolution of 1024x768, keyboard, and mouse.
3. Hard Drive should be in NTFS file-system formatted with minimum 10 GB of free space.
4. A Laser printer will need to be used to print these reports and notes

2.4.2 Software

1. Software is designed to run on any platform above Microsoft Windows 7 (32bit).
2. Node JS, Express
3. Database: MongoDB
4. SMTP: EmailJS

2.4.3 High level Language

1. ReactJS
2. HTML, CSS
3. NoSQL - MongoDB

2.5 Assumption and Dependencies

Some software used in implementing the system is with high cost and the client has agreed to afford the amount of money needed to purchase them. It's assumed that client won't change that decision on the next phases of the software development. Although we assume that client is using windows 7 or above. Otherwise if client use an open source operating system, there is a need of changing the SRS accordingly.

3 Specific Requirement

3.1 Functional Requirement

- Register Customer
- Make Reservations
- Search Rooms
- Add Additional Facilities (Jacuzzi, Swimming pool, Meals, Additional beds)
- Make Payment
- Manage Room Details
- Generate and E-mail Receipt

3.2 Non-Functional Requirements

3.2.1 Safety Requirements

There are several user levels in hotel reservation system, Access to the various subsystems will be protected by a user log in screen that requires a user name and password. This gives different views and accessible functions of user levels through the system. Maintaining backups ensure the system database security. System can be restoring in any case of emergency.

3.2.2 Security Requirements

Customers and Admins will be able to log in to the Hotel Reservation System. Customers will have access to the Reservation/Booking and subsystems. Admins have the maximum privilege to all subsystems. Access to the various subsystems will be protected by a user log in screen that requires a user name and password.

3.2.3 Other Requirements

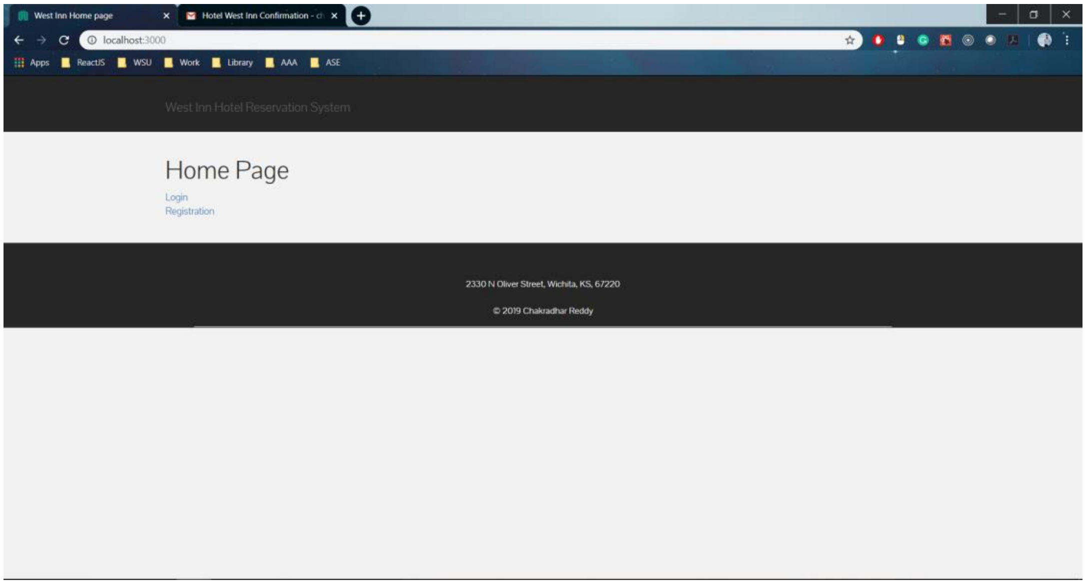
When the system is completely developed and submitted to the client, few sessions will be required to make the users of the system understand about the functionality of it and some time to adapt to the system. After those sessions, it’s required that a member from the development team should spend some time in the system background for an agreed time period. That time period will be used in identifying new bugs that could not be reached in the earlier phases of the development process.

Client should have a valid e-mail account in order to receive reservation e-mail notifications.

3.3 External Interface

3.3.1 User Interface

Home Page



West Inn Hotel Reservation System

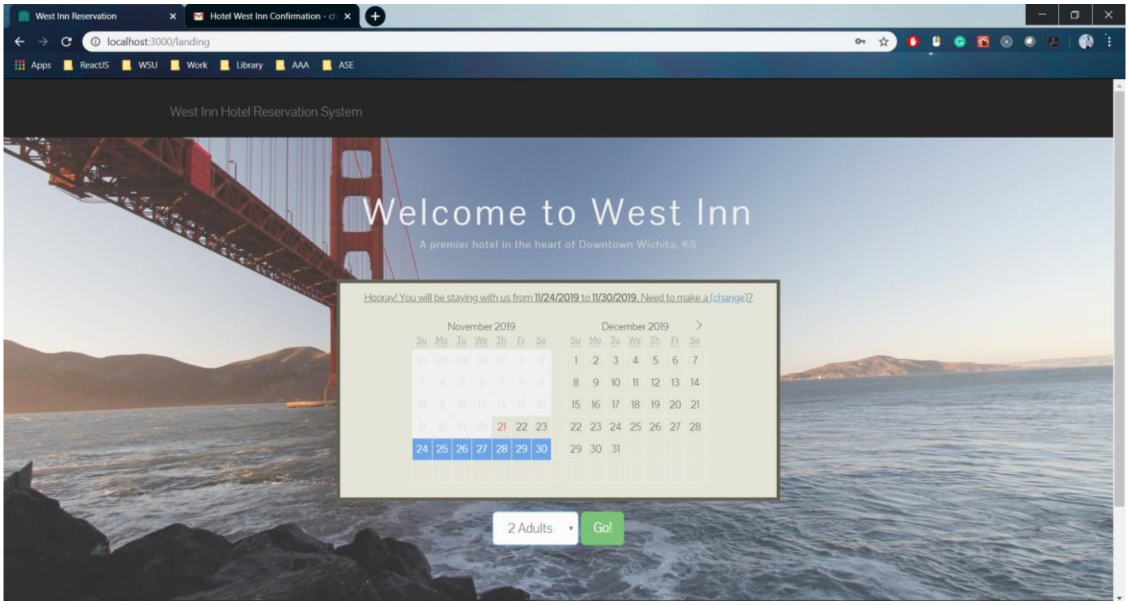
Home Page

Login
Registration

2330 N Oliver Street, Wichita, KS, 67220

© 2019 Chakradhar Reddy

Check-in and Checkout section interface



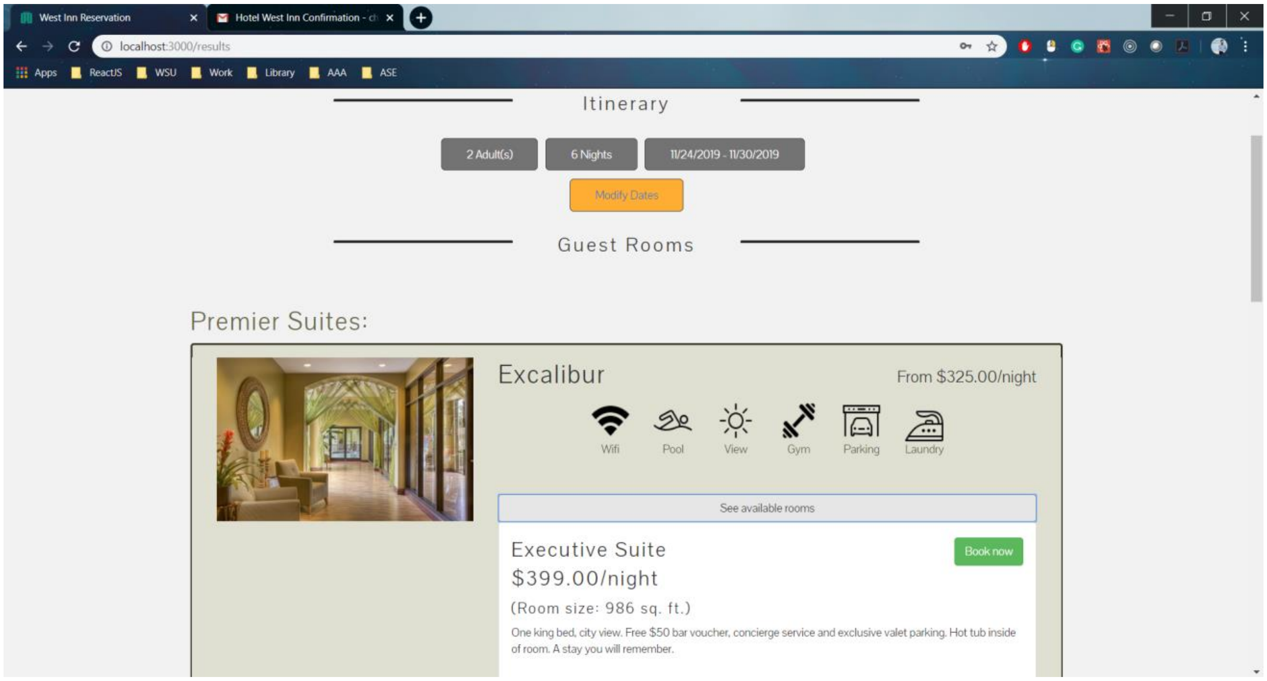
West Inn Hotel Reservation System

Welcome to West Inn

A premier hotel in the heart of Downtown Wichita, KS

Hooray! You will be staying with us from 11/24/2019 to 11/30/2019. Need to make a (change)?

Room selection interface



Itinerary

2 Adult(s) | 6 Nights | 11/24/2019 - 11/30/2019

Modify Dates

Guest Rooms

Premier Suites:

Excalibur

From $325.00/night

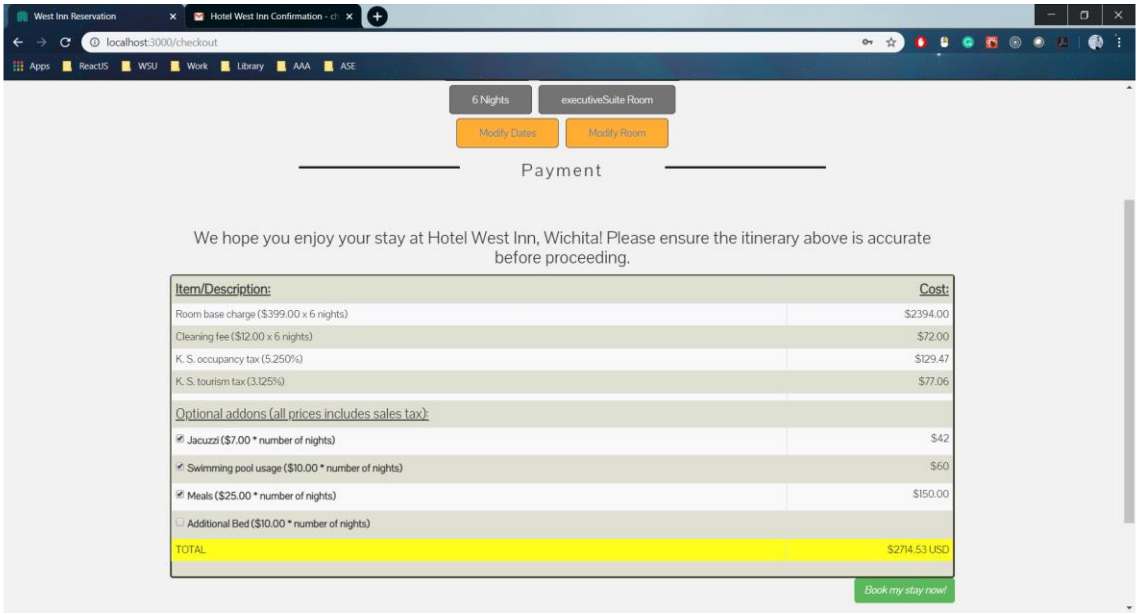
Executive Suite

$399.00/night

(Room size: 986 sq. ft.)

One king bed, city view. Free $50 bar voucher, concierge service and exclusive valet parking. Hot tub inside of room. A stay you will remember.

Total Payment Cost

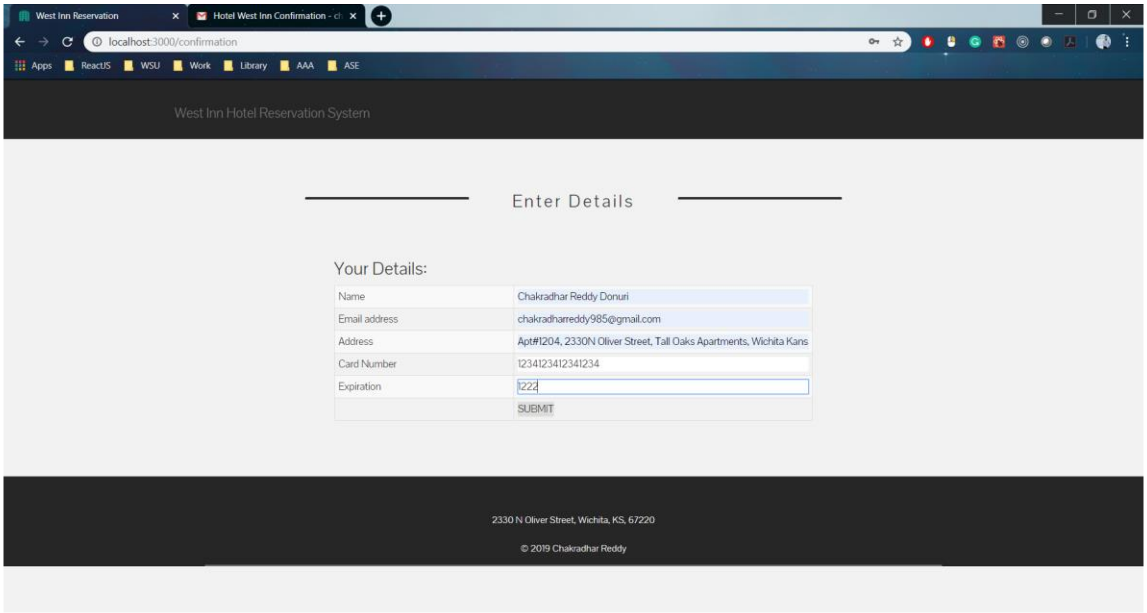


Payment

We hope you enjoy your stay at Hotel West Inn, Wichita! Please ensure the itinerary above is accurate before proceeding.

Item/Description:	Cost:
Room base charge ($399.00 x 6 nights)	$2394.00
Cleaning fee ($12.00 x 6 nights)	$72.00
K. S. occupancy tax (5.250%)	$129.47
K. S. tourism tax (3.125%)	$77.06
Optional addons (all prices includes sales tax):	
☑ Jacuzzi ($7.00 * number of nights)	$42
☑ Swimming pool usage ($10.00 * number of nights)	$60
☑ Meals ($25.00 * number of nights)	$150.00
☐ Additional Bed ($10.00 * number of nights)	
TOTAL	$2714.53 USD

Payment Details



West Inn Reservation
Hotel West Inn Confirmation - ch
localhost:3000/confirmation
Apps
ReactJS
WSU
Work
Library
AAA
ASE
West Inn Hotel Reservation System

Enter Details

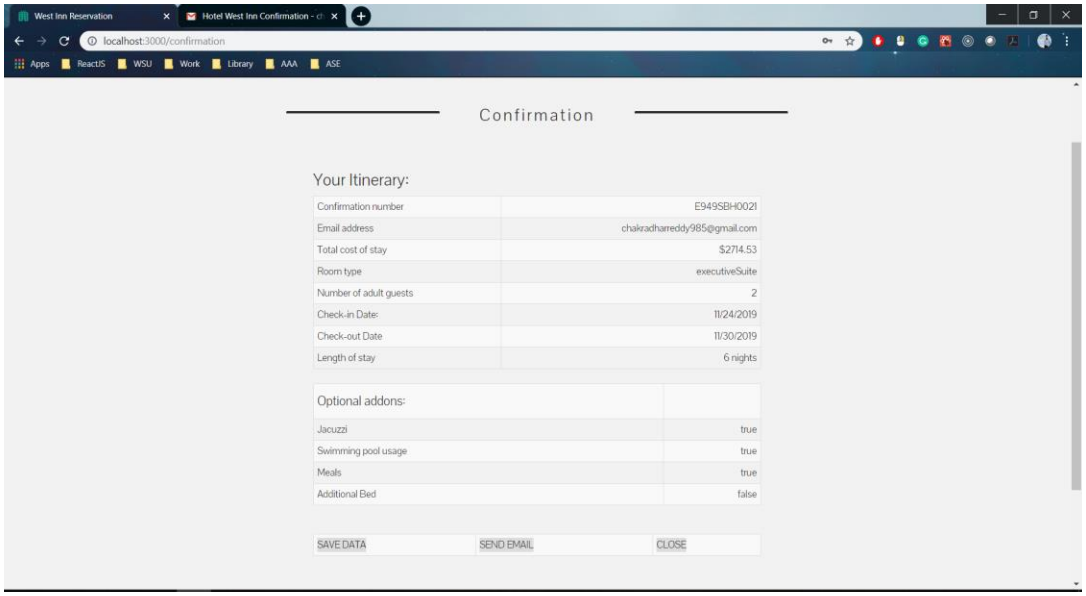
Your Details:

Name	Chakradhar Reddy Donuri
Email address	chakradharreddy985@gmail.com
Address	Apt#1204, 2330N Oliver Street, Tall Oaks Apartments, Wichita Kans
Card Number	1234123412341234
Expiration	1222
	SUBMIT

2330 N Oliver Street, Wichita, KS, 67220

© 2019 Chakradhar Reddy

Payment Confirmation



Confirmation

Your Itinerary:

Confirmation number	E949SBH0021
Email address	chakradharreddy985@gmail.com
Total cost of stay	$2714.53
Room type	executiveSuite
Number of adult guests	2
Check-in Date:	11/24/2019
Check-out Date	11/30/2019
Length of stay	6 nights

Optional addons:	
Jacuzzi	true
Swimming pool usage	true
Meals	true
Additional Bed	false

3.3.2 Hardware Interfaces

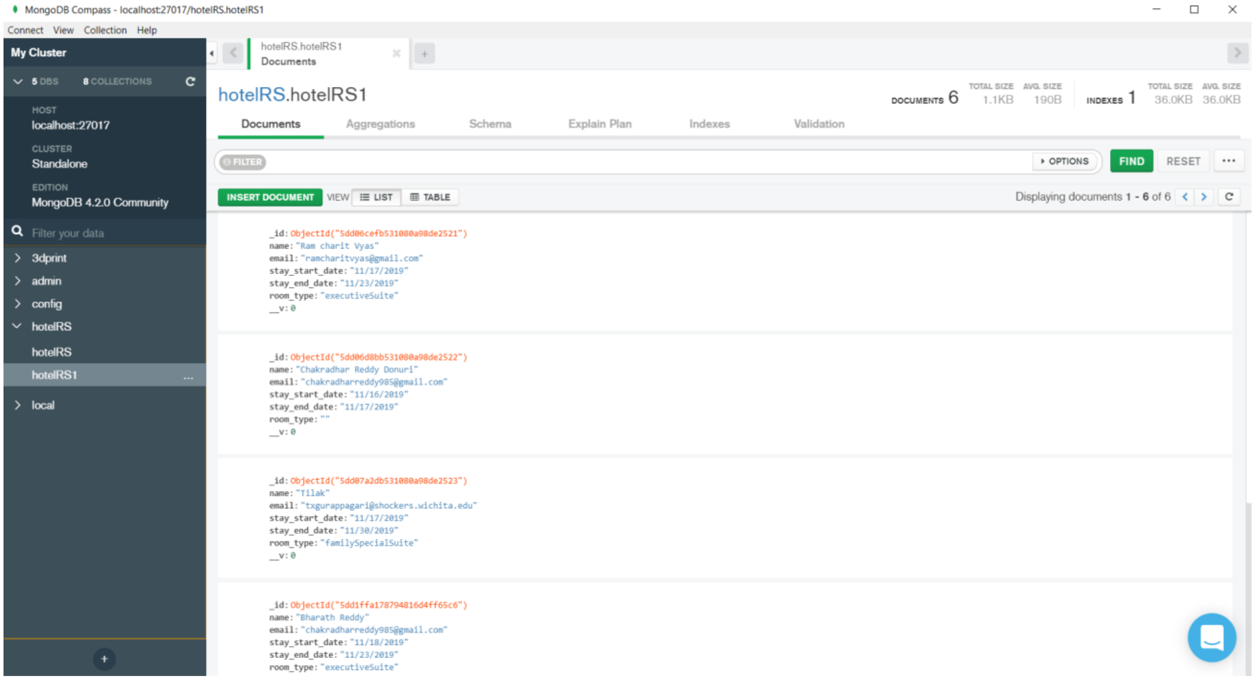
Reservation alerts will be sent to the admin as an e-mail notification. So, there is a need of broadband internet connection. Client should able to keep a stable internet connection.

A laser printer will be needed when printing bills and several reports

3.3.2 Software Interfaces

The computer this software going to be install need to have Windows Operating System equal or above, Windows 7. On that Windows platform any latest web browser will be installed and that will be the platform the particular software will be run. There will be an ADO.NET data transmission with the Microsoft SQL Server Management Studio Express 2010 R2 edition that will be installed in the same computer.

3.3.3 Database Interface



HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 4.2.0 Community

hotelRS.hotelRS1

DOCUMENTS 6 TOTAL SIZE 1.1KB AVG. SIZE 190B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Displaying documents 1 - 6 of 6

```
_id: ObjectId("5dd06cefb531080a98de2521")
name: "Ram charit Vyas"
email: "ramcharitvyas@gmail.com"
stay_start_date: "11/17/2019"
stay_end_date: "11/23/2019"
room_type: "executiveSuite"
__v: 0

_id: ObjectId("5dd06d8bb531080a98de2522")
name: "Chakradhar Reddy Donuri"
email: "chakradharreddy985@gmail.com"
stay_start_date: "11/16/2019"
stay_end_date: "11/17/2019"
room_type: ""
__v: 0

_id: ObjectId("5dd07a2db531080a98de2523")
name: "Tilak"
email: "txgurappagari@shockers.wichita.edu"
stay_start_date: "11/17/2019"
stay_end_date: "11/30/2019"
room_type: "familySpecialSuite"
__v: 0

_id: ObjectId("5dd1ffa178794816d4ff65c6")
name: "Bharath Reddy"
email: "chakradharreddy985@gmail.com"
stay_start_date: "11/18/2019"
stay_end_date: "11/23/2019"
room_type: "executiveSuite"
```

3.3.4 Communications Interfaces

When a specific reservation reserved at the same time an e-mail notification will be sent to both admin’s e-mail account and customer’s account. Customer will be notified in the check-out date. To achieve that functionality, it requires having a stable internet connection. Mostly a broadband connection with the client’s computer will provide the efficient service.

3.4 Performance Requirement

Performance requirements define acceptable response times for system functionality. Although the system is developed suiting for the least system performances, the performance of the system will highly depend on the performance of the hardware and software components of the installing computer. When consider about the timing relationships of the system the load time for user interface screens shall take no longer than two seconds. It makes fast access to system functions. The log in information shall be verified within five seconds causes’ efficiency of the system. Returning query results within five seconds makes search function more accurate.

3.5 Logical Database Requirements

The logical database requirements include the retention of the following data elements.

- Customer first name
- Customer last name
- Customer Code
- Customer address
- Customer phone number
- Number of occupants
- Room no
- Floor no
- Building no
- Room status
- Employee id
- Bill no
- Default room rate
- Rate description
- Guaranteed room (yes/no)
- Expected check-in date
- Actual check-in time
- Expected check-out time
- Actual check-out date
- Customer feedback
- Payment type
- Total Bill

3.6 Attributes

3.6.1 Reliability

Specify the factors required to establish the required reliability of the software system at time of delivery. Mean time between failures and mean time to recovery.

3.6.2 Availability

The system shall be available during normal hotel operating hours.

3.6.3 Security

The extent to which the Hotel Reservation System is safe from outside non-allowed user or attacks.

3.6.4 Maintainability

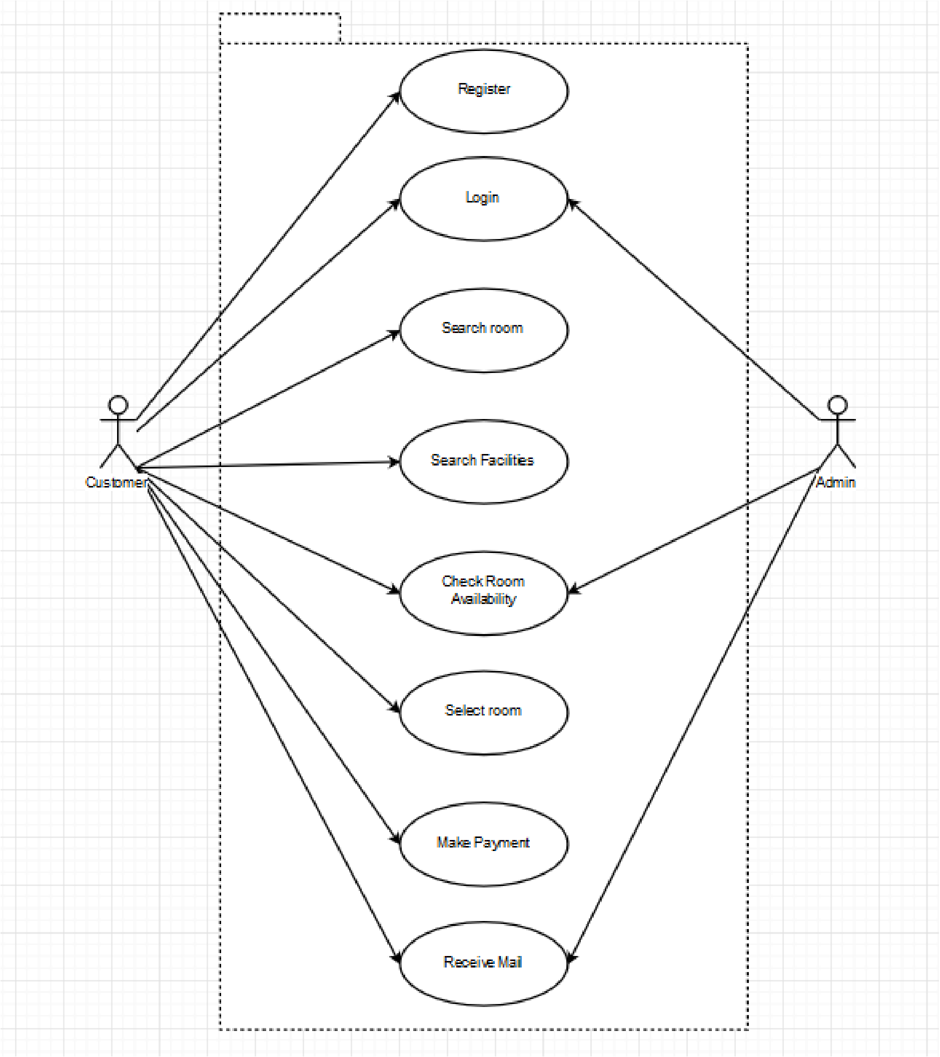
What design, coding standards must be adhered to exclusions created

3.6.5 Portability

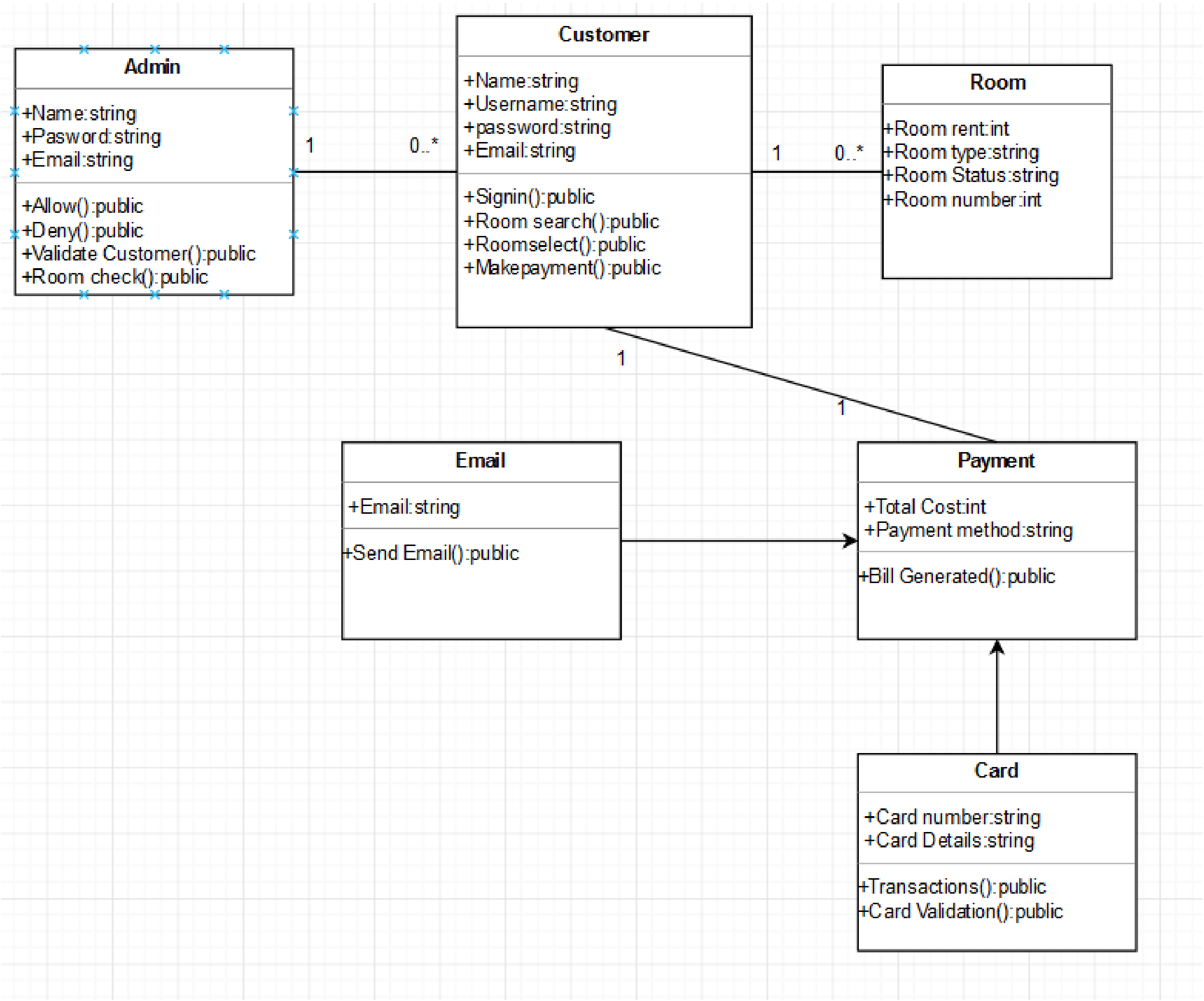
The Hotel Reservation System shall run in any Microsoft Windows environment

4. Design

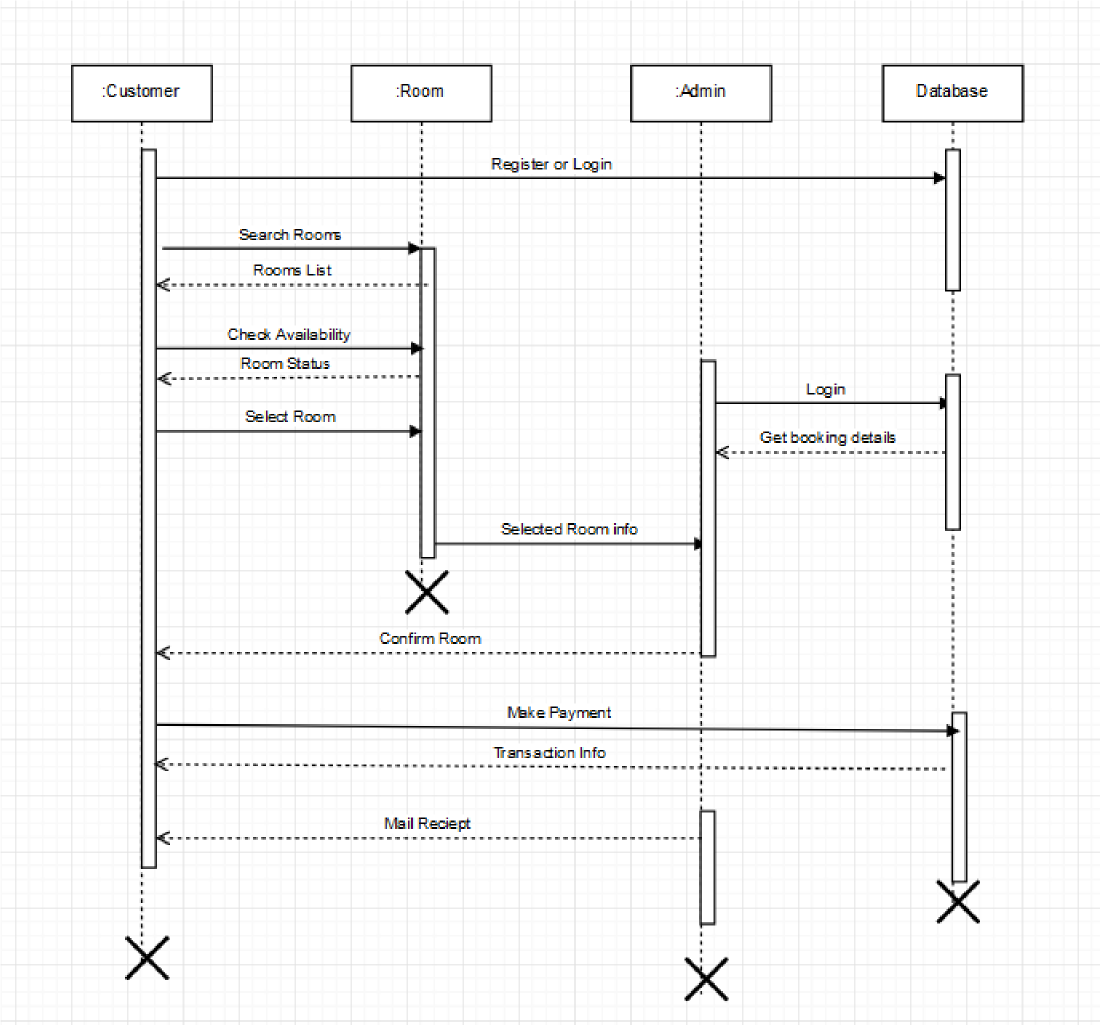
4.1 Use case



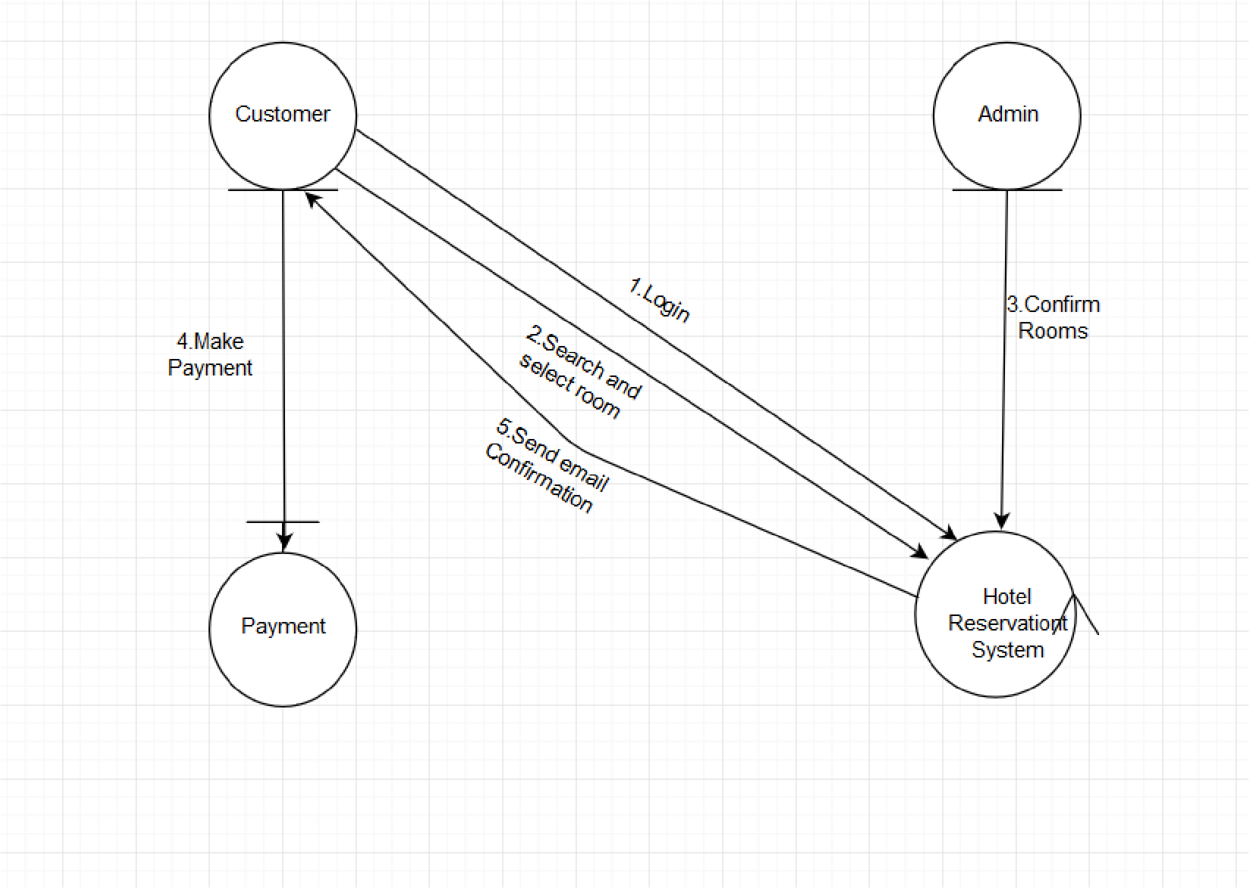
4.2 Class Diagram



4.3 Sequence Diagram

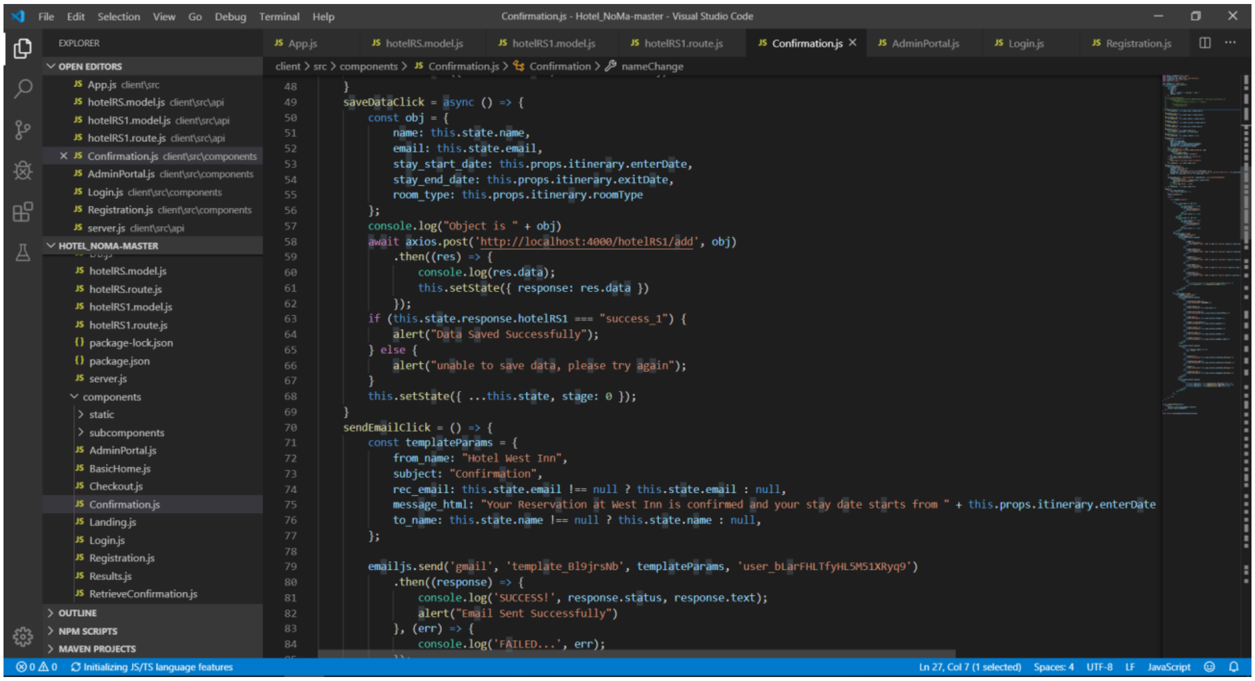


4.4 Data Flow Diagram



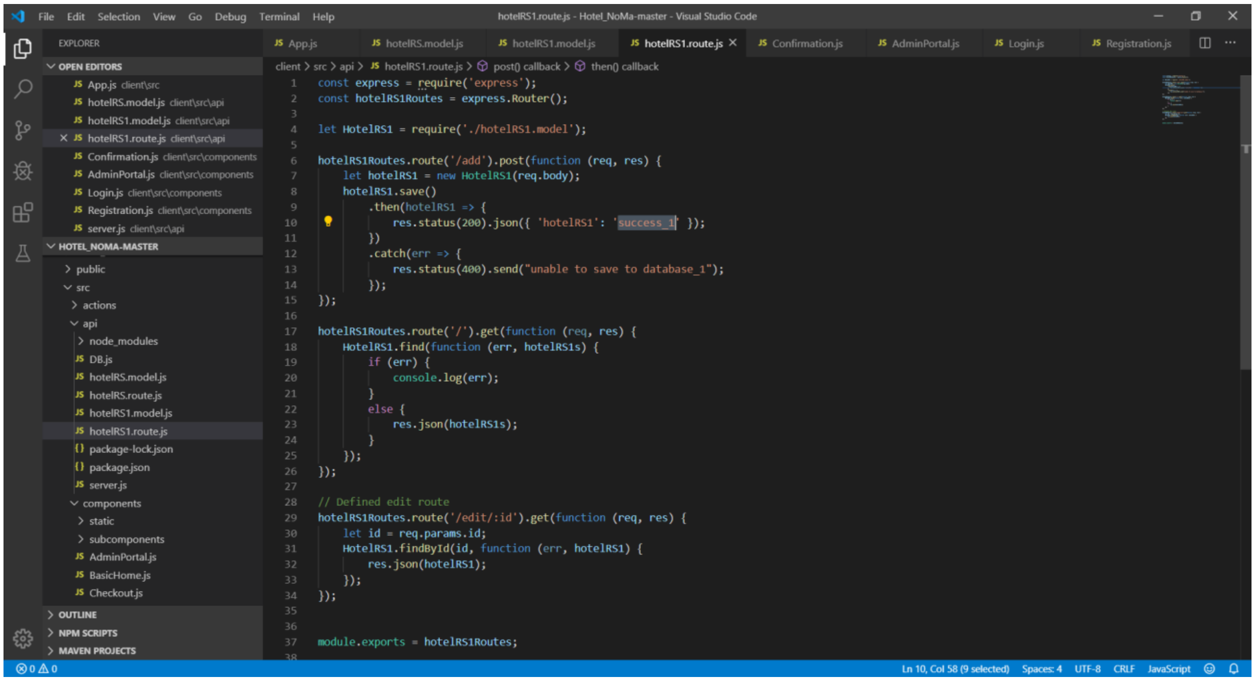
5 Implementation

Code Snippets


File Edit Selection View Go Debug Terminal Help
Confirmation.js - Hotel_NoMa-master - Visual Studio Code
EXPLORER
OPEN EDITORS
App.js client\src
hotelRS.model.js client\src\api
hotelRS1.model.js client\src\api
hotelRS1.route.js client\src\api
Confirmation.js client\src\components
AdminPortal.js client\src\components
Login.js client\src\components
Registration.js client\src\components
server.js client\src\api
HOTEL_NOMA-MASTER
hotelRS.model.js
hotelRS.route.js
hotelRS1.model.js
hotelRS1.route.js
package-lock.json
package.json
server.js
components
static
subcomponents
AdminPortal.js
BasicHome.js
Checkout.js
Confirmation.js
Landing.js
Login.js
Registration.js
Results.js
RetrieveConfirmation.js
OUTLINE
NPM SCRIPTS
MAVEN PROJECTS
App.js
hotelRS.model.js
hotelRS1.model.js
hotelRS1.route.js
Confirmation.js
AdminPortal.js
Login.js
Registration.js
client > src > components > Confirmation.js > Confirmation > nameChange
⊗ 0 ⚠ 0
Initializing JS/TS language features
Ln 27, Col 7 (1 selected) Spaces: 4 UTF-8 LF JavaScript

```
    }
    saveDataClick = async () => {
        const obj = {
            name: this.state.name,
            email: this.state.email,
            stay_start_date: this.props.itinerary.enterDate,
            stay_end_date: this.props.itinerary.exitDate,
            room_type: this.props.itinerary.roomType
        };
        console.log("Object is " + obj)
        await axios.post('http://localhost:4000/hotelRS1/add', obj)
            .then((res) => {
                console.log(res.data);
                this.setState({ response: res.data })
            });
        if (this.state.response.hotelRS1 === "success_1") {
            alert("Data Saved Successfully");
        } else {
            alert("unable to save data, please try again");
        }
        this.setState({ ...this.state, stage: 0 });
    }
    sendEmailClick = () => {
        const templateParams = {
            from_name: "Hotel West Inn",
            subject: "Confirmation",
            rec_email: this.state.email !== null ? this.state.email : null,
            message_html: "Your Reservation at West Inn is confirmed and your stay date starts from " + this.props.itinerary.enterDate
            to_name: this.state.name !== null ? this.state.name : null,
        };

        emailjs.send('gmail', 'template_Bl9jrsNb', templateParams, 'user_bLarFHLTfyHL5M51XRyq9')
            .then((response) => {
                console.log('SUCCESS!', response.status, response.text);
                alert("Email Sent Successfully")
            }, (err) => {
                console.log('FAILED...', err);
```


File Edit Selection View Go Debug Terminal Help
hotelRS1.route.js - Hotel_NoMa-master - Visual Studio Code
EXPLORER
OPEN EDITORS
App.js client\src
hotelRS.model.js client\src\api
hotelRS1.model.js client\src\api
hotelRS1.route.js client\src\api
Confirmation.js client\src\components
AdminPortal.js client\src\components
Login.js client\src\components
Registration.js client\src\components
server.js client\src\api
HOTEL_NOMA-MASTER
public
src
actions
api
node_modules
DB.js
hotelRS.model.js
hotelRS.route.js
hotelRS1.model.js
hotelRS1.route.js
package-lock.json
package.json
server.js
components
static
subcomponents
AdminPortal.js
BasicHome.js
Checkout.js
OUTLINE
NPM SCRIPTS
MAVEN PROJECTS
App.js
hotelRS.model.js
hotelRS1.model.js
hotelRS1.route.js
Confirmation.js
AdminPortal.js
Login.js
Registration.js
client > src > api > hotelRS1.route.js > post() callback > then() callback
⊗ 0 ⚠ 0
Ln 10, Col 58 (9 selected) Spaces: 4 UTF-8 CRLF JavaScript

```
const express = require('express');
const hotelRS1Routes = express.Router();

let HotelRS1 = require('./hotelRS1.model');

hotelRS1Routes.route('/add').post(function (req, res) {
    let hotelRS1 = new HotelRS1(req.body);
    hotelRS1.save()
        .then(hotelRS1 => {
            res.status(200).json({ 'hotelRS1': 'success_1' });
        })
        .catch(err => {
            res.status(400).send("unable to save to database_1");
        });
});

hotelRS1Routes.route('/').get(function (req, res) {
    HotelRS1.find(function (err, hotelRS1s) {
        if (err) {
            console.log(err);
        }
        else {
            res.json(hotelRS1s);
        }
    });
});

// Defined edit route
hotelRS1Routes.route('/edit/:id').get(function (req, res) {
    let id = req.params.id;
    HotelRS1.findById(id, function (err, hotelRS1) {
        res.json(hotelRS1);
    });
});


module.exports = hotelRS1Routes;
```

6 Testing

Test case 1: User tries to register

Success: Registered Successfully

Failure: Unable to register Please try again

Test case 2: User tries to Login

Success: Goes to Landing Page

Failure: Stays back in same page

Test case 3: User selects specific dates, number of people and particular room of his choice and can add additional facilities and amount reflects respectively

Test case 4: When submits payment

Success: Data saved Successfully and Email Sent successfully

Failure: Unable to save data Please try again