

## Hotel Reservation System

Nandasiri K.G.N.D<sup>1</sup>, Gamage M.I.S<sup>2</sup>, Wanniarachchi M.Y<sup>3</sup>, Lakshani N.V.M<sup>4</sup>, D.I De Silva<sup>5</sup> and Dulanji Cooray<sup>6</sup>

<sup>1</sup>Student, Department of Software Engineering, Sri Lanka Institute of Information Technology, SRI LANKA

<sup>2</sup>Student, Department of Software Engineering, Sri Lanka Institute of Information Technology, SRI LANKA

<sup>3</sup>Student, Department of Software Engineering, Sri Lanka Institute of Information Technology, SRI LANKA

<sup>4</sup>Student, Department of Software Engineering, Sri Lanka Institute of Information Technology, SRI LANKA

<sup>5</sup>Senior Lecturer, Department of Software Engineering, Sri Lanka Institute of Information Technology, SRI LANKA

<sup>6</sup>Instructor, Department of Software Engineering, Sri Lanka Institute of Information Technology, SRI LANKA

<sup>1</sup>Corresponding Author: [nayani.dn@gmail.com](mailto:nayani.dn@gmail.com)

### ABSTRACT

The essay "Hotel Management System" focuses on managing hotel functions between the customer and the hotel. This system is divided into four components food, hall, room, and taxi. An administrator would be capable of managing taxis, menus, halls, and rooms, whereas a customer would be able to view menus and book a hall, room, or taxi. This paper discusses how the system manages all of these functionalities. In addition, the system has been designed to generate reports on every functionality for the administrator to effectively administer the system. This system includes several features, each of which is accessible to either the administrator or the customer. The main goal of this paper is to make the hotel's fundamental functions with customers easier and to provide an online presentation of the hotel for customers to connect more.

**Keywords**– Customer, Hotel, Reports, Booking, Online

out of the bargain, low occupancy rate and low profit “three low” strange circle. Therefore, linking true business and information technology management to facilitate customer service and manage service delivery helps to build the hotel industry successfully.

Jingda Yang d divided the hotel management system into four main parts to manage all functions of the system, basic information management, reception management, complete desk cashier management and system user management. [1] Their system uses the Microsoft DNA structure, which breaks the application into three layers: the presentation layer, the business rules layer, and the data layer.

The proposed Hotel Management System before are include comparative evaluation factors such as privacy, security and data protection, Usage simplicity, Accessibility to system [2]. These factors are the highest turnaround time, an important measure of efficiency.

## I. INTRODUCTION

Technology integrations is continuing to evolve the hotel management systems. This web-based application is designed for hotel management system. Customer can do their task very easily and the administrator can manage system and provide service to the customer via system. The system easier to use than manual system.

And while building this app, good practices such as interface are designed by focusing on one specific task, using easy to read and understand words without using complex or system-oriented words for menu names, adding relevant labels for input fields, Capital letters have been used for the beginning of each word in each label, and labels and buttons have been given meaningful names to make the site more user-friendly.

## II. RELATED WORK

Day by day the hotel industry has become more competitive in the business sector. Until recently, the traditional hotel industry has found it difficult to pull itself

## III. METHODOLOGY

### A. Technologies

This developed application used Express JS, Node.js for server side, Rest API and ReactJS for server side and mongo dB as database. Figure 1 will show how these technologies connect with each other's.

#### *Express JS, Node JS*

Node J's is a fast JavaScript run-time environment, but it doesn't support request handling, HTTP methods, or saving files, so Express JS comes into the picture. ExpressJS makes node JS easier. Express provides a wide range of features for developing both web and mobile applications. It is used to build single-page, multi-page, and hybrid-page web applications. Express writing in Java Script is very easy to understand for developers. Express is a Node.js framework designed for building APIs, web applications, and cross-platform mobile apps. Express is fast, and you believe there is no structured way to write code; Express is not forced to write code in a fox manner; and Express is lightweight.

**React JS**

The app itself is the client, or front-end part. Under the hood, it needs to talk to a server or the backend to get or save the data. This communication happens using the HTTP protocol, the same protocol that powers our website. So, on the server, we expose a bunch of services that are accessible via the HTTP protocol. Then the client can directly call the services by sending an HTTP request. This is where the rest is needed. "Rest" is short for "Representational State Transfer. The Rest is basically a converter for building these HTTP services. We use simple HTTP protocol principles to provide support for creating, reading, updating, and deleting data. These operations are all together called "crud operations."

**Rest API**

The client, or frontend, component is the app itself. To obtain or save the data, it needs to communicate with a server or the backend. This communication takes place over the HTTP protocol, which also underpins our web. As a result, we expose a number of services on the server that may be accessed using the HTTP protocol. The client can then make a direct HTTP request to the services. The downtime is required here. It stands for "Representational State Transfer" or "REST." In essence, Rest serves as a converter when creating these HTTP services. We provide support for creating, reading, updating, and removing data using basic HTTP protocol concepts. All these operations are collectively referred to as "crud operations."

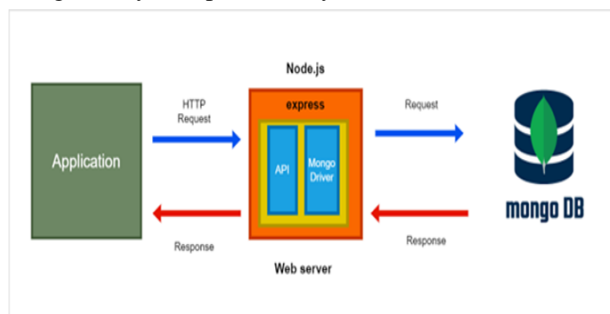
**Mongo dB**

Alright, so this is what's called a no SQL database and there are different types of no SQL databases, but to be more specific, it's called a document database. Data or records are stored as documents and they use JSON like syntax, so if you know what Jason is, its JavaScript object notation. No SQL databases differ significantly from relational databases such as MySQL. With a relational database, we must map out everything required to determine the exact schema, including which tables and fields are used. MongoDB is the most advanced and most popular document-oriented database. MongoDB is easy to scale compared to SQL or relational databases, and they are also much faster in most types of operations. There are some things that would benefit from using relational databases. When creating a new document in MongoDB, it assigns an object ID that is unique for that collection. Multiple fields within the document can have values of various data types such as strings, arrays, objects, and so on.

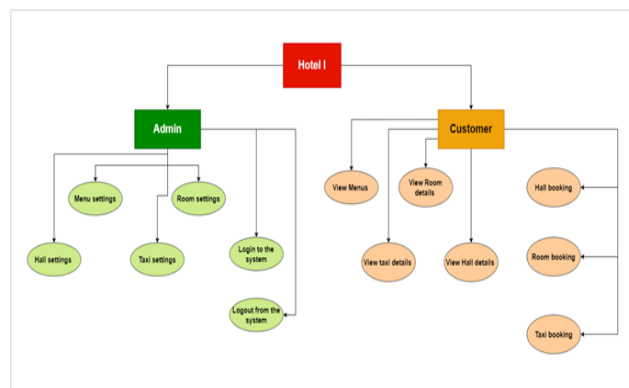
**B. Proposed System**

According to the figure 2 this proposed system mainly focuses on room management, menu management, hall management, and taxi management. And cover the

admin side and customer side. A customer can get service through the system provided by the admin.



**Figure 1 : System Overview**

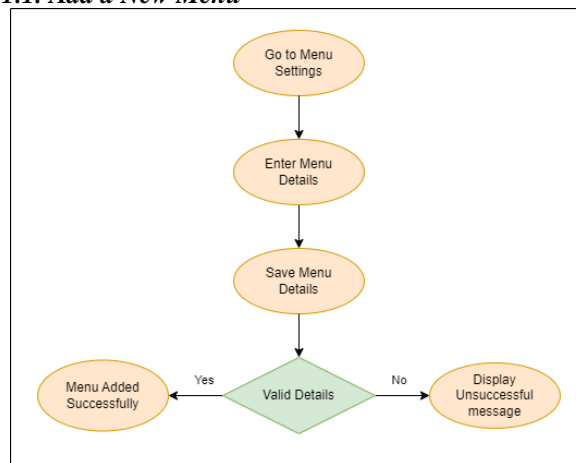


**Figure 2 : Whole System**

**1. Food Menu Functions**

**1.1. Admin-Side**

**1.1.1. Add a New Menu**



**Figure 3: Add New Menu**

Admin can add new food menus to the system using this function. According to the Figure 3 admin can enter required details and submit it by clicking 'Save Details' Button.

1.1.2. View already added menus, update and delete them

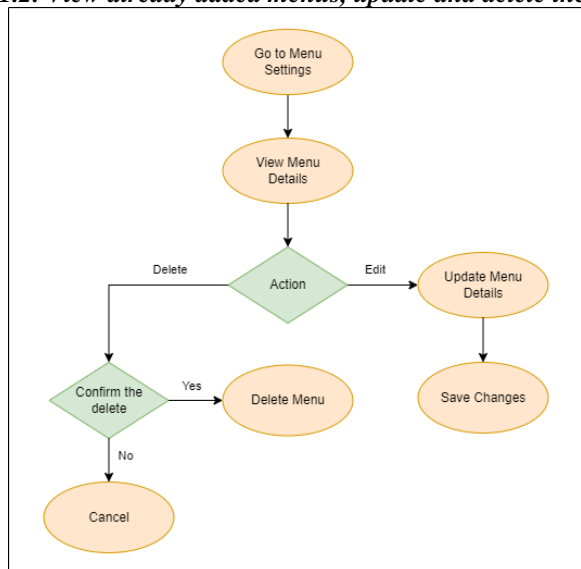


Figure 4: Update, Delete Menu

According to the figure 9 admin can view already added food menus, update and delete them. If admin wants to update a menu, admin should click update button of respective menu cards. Then Update Menu Details interface will appear, and admin can make the necessary changes and update the details by clicking the 'Update Details' button. If the menu details updated successfully, the alert message 'Menu details updated successfully' will be displayed.

When the admin clicks the delete button in the corresponding menu card, popup interface will appear, asking the administrator for the deletion approval. If admin want to delete a menu from system permanently admin can click 'Delete' button or if admin accidentally click delete button, admin can click 'Cancel' button and go back to all food menu page. If the menu is deleted successfully, the alert message 'Menu details deleted successfully' will be displayed.

1.2. Client-Side

1.2.1. View all menus and details of a selected menu

As Figure 8 customers can view all the food menus and view details of relevant menus by clicking 'Menu Details' button.

2. Rooms Functions

2.1. Admin-Side

2.1.1. Add a new room

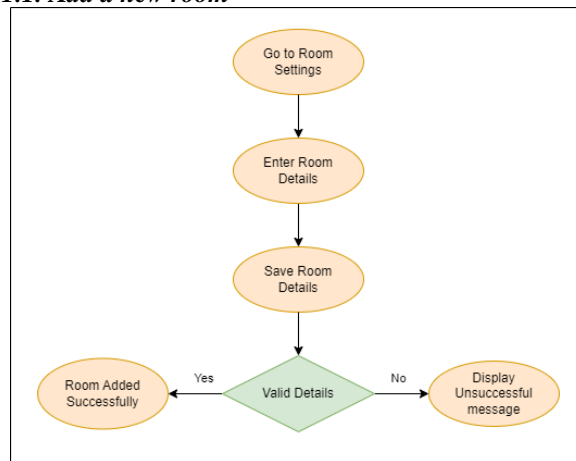


Figure 5: Add New Room

Admin can add new room to the system using this function. According to the Figure 3 admin can enter required details and submit it by clicking 'Save Details' Button.

2.1.2. View already added rooms, update and delete them

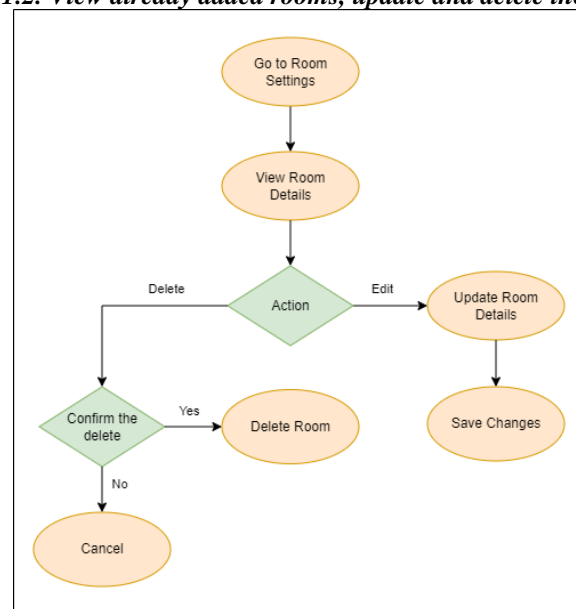
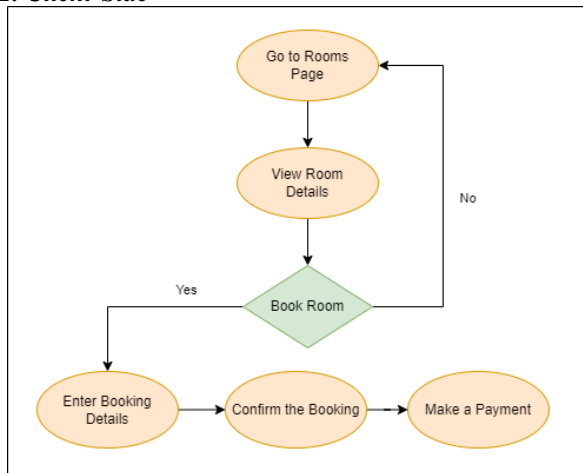


Figure 6: Update, Delete Room

According to the figure 6 admin can view already added rooms, update and delete them. If admin wants to update a room, admin should click update button of respective room cards. Then Update Room Details interface will appear, then admin can make the necessary changes and update the details by clicking the 'Update Details' button. If the room details updated successfully, the alert message 'Room details updated successfully' will be displayed.

When the admin clicks the delete button in the corresponding room card, popup interface will appear, asking the administrator for the deletion approval. If admin want to delete a room from system permanently admin can click 'Delete' button or if admin accidentally click the delete button in the room card, admin can click 'Cancel' button and go back to 'All Rooms' page. If the room is deleted successfully, the alert message 'Room details deleted successfully' will be displayed.

**2.2. Client-Side**



**Figure 7: Book a Room and Make the Payment**

**2.2.1. View all rooms**

This function focused to display available room options to the customers. In here users can view the rooms list. Users can also search rooms by selecting relevant room types, guest count, check-In date, and Check-out date.

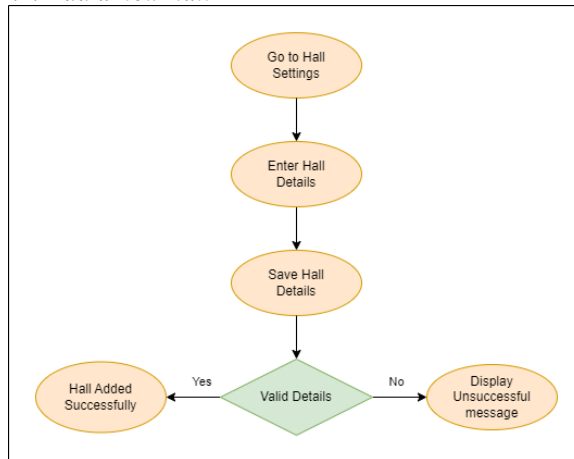
**2.2.2. View details of the selected room and book them by making payments**

In here users can view the details of the selected room. Users can book the room by filling the required information and they will be redirected to the payment page by clicking the "BOOK" Button. Then customers must fill required details and provide the correct payment information. Then they can click the "CONFIRM THE BOOKING" Button after rechecking the information and the Total amount. After clicking the button, they will get an email with the booking information. When clicking the 'Reservation Details' button customer can view the report of reservation details. It will work as Figure 7.

**3. Halls Functions**

**3.1. Admin-Side**

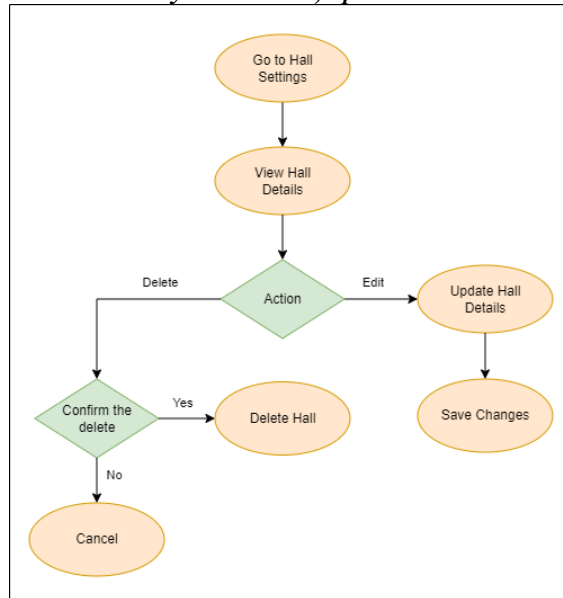
**3.1.1. Add a new hall**



**Figure 8: Add New Hall**

Admin can add new halls to the system using this function. According to the Figure 8 admin can enter required details and submit it by clicking 'Save Details' Button.

**3.1.2. View already added halls, update and delete them**



**Figure 9: Update, Delete Hall**

According to the figure 9 admin can view already added halls, update and delete them. If admin wants to update hall, admin should click update button of respective hall cards. Then Update Hall Details interface will appear, and admin can make the necessary changes and update the details by clicking the 'Update' button. If the room details updated successfully, the alert message 'Details updated successfully' will be displayed.

When the admin clicks the delete button in the corresponding hall card, popup interface will appear,

asking the administrator for the deletion approval. If admin want to delete hall from system permanently admin can click 'Delete' button or if admin accidentally click delete button, admin can click 'Cancel' button and go back to the halls page. If the hall is deleted successfully, the alert message 'Hall removed successfully' will be displayed.

**3.2. Client-Side**

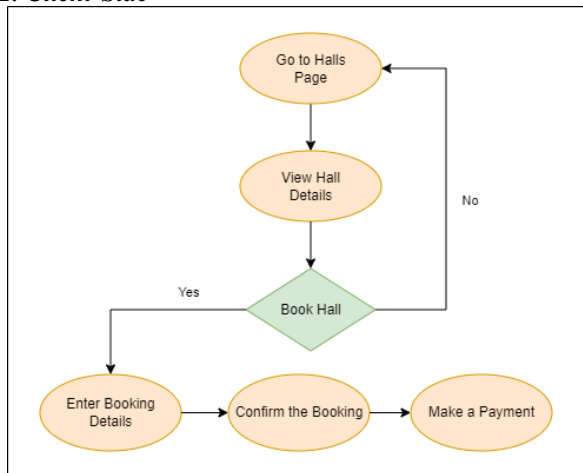


Figure 10 : Book a Hall and Make the Payment

**3.2.1. View all halls**

This function focused to display available halls options to the users. In here Customers can see all the halls are available in our hotel. And brief introduction of hall and image of hall in here. Furthermore, users can see the details of relevant halls by clicking the 'Explore More' button.

**3.2.2. View details of the selected hall and book them by making payments**

According to the figure 10, this function is used to display details of the selected hall to the customers and book them by making payments. In here users can view the details of the selected hall. Users can book the hall by filling the required information and they will be redirected to the reservation details page by clicking the "PROCEED" Button. Customer can see the details enter by him/her when make hall reservation. After checking the details customer must confirm it for make payment. Customer redirects to payments page when clicking the 'Pay Now' button. Then customer have to provide the correct payment information and they can click the "PAY NOW" Button after rechecking the information. When clicking the 'Reservation Details' button customer can view the report of reservation details.

**4. Taxi Functions**

**4.1. Admin-Side**

**4.1.1. Add a new taxi**

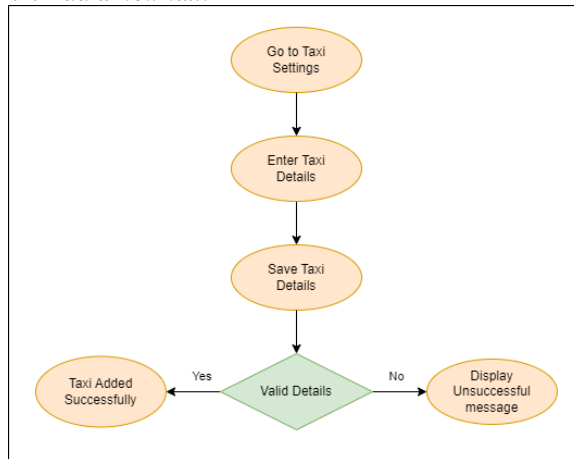


Figure 11: Add New Taxi

Admin can add new taxis to the system using this function. According to the Figure 11 admin can enter required details and submit it by clicking 'Save Details' Button.

**4.1.2. View already added taxis, update and delete them**

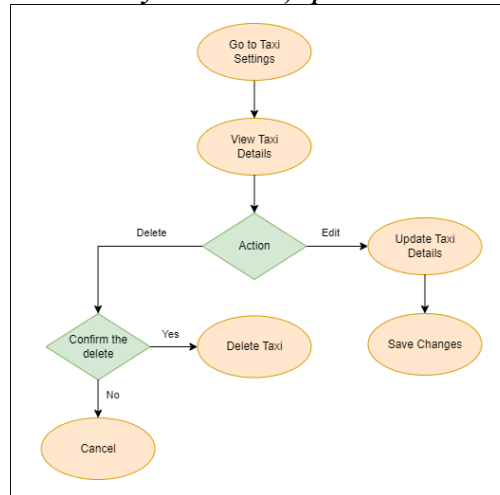


Figure 12: Update, Delete Taxi

Admin can view already added taxis, update and delete them as in the figure 12. If admin wants to update taxi, admin should click update button of respective taxi cards. Then Update Taxi interface will appear, and admin can make the necessary changes and update the details by clicking the 'Update Taxi' button.

When the admin clicks the delete button in the corresponding taxi card it will delete the selected taxi from the system.

4.2. Client-Side

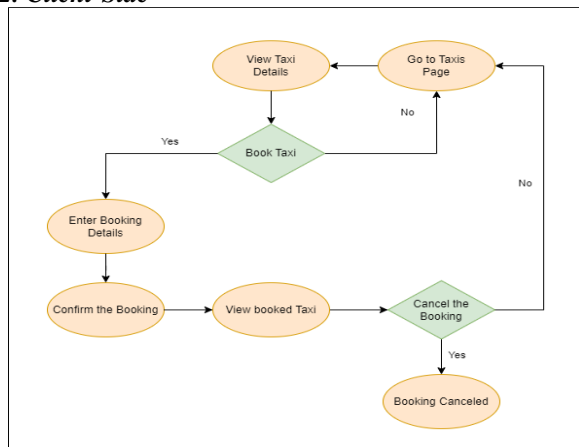


Figure 13: Book a Taxi and Cancel Booking

4.2.1. View all taxis

This function is used to display available taxi options to the users. In here Customers can see all the taxis available. Furthermore, users can see the details of relevant taxi and book them by clicking the ‘Book’ button.

4.2.2. View details of a single taxi and book taxis by providing requested details

This function is used to display Taxi/ Vehicle details to the user and to book that taxi/ vehicle. Once user selects a taxi from the “Book Your Vehicle” page’s list, it will redirect to this page. User can find all the relevant details regarding the selected taxi. If user wants to book this vehicle, he/she can click “Book” button and it will redirect to the book your vehicle page. Then user can fill the form and book the Taxi/Vehicle if they want. This function act as figure 13.

4.2.3. View already booked taxis of the current user and cancel the bookings if they want

This function is used to display and delete all the booked taxis for the user. This page will display all the taxis booked by a user with the details user provided when he booked that specific taxi. If user wants to delete a booked taxi, he can click “Delete” button. If user wants to find a booked taxi quickly, he can search it through the search bar. The flow of this function is showed in figure 13.

C. Process of the project

We organized a meeting with some clients via Microsoft teams and gather requirements for the hotel management system and called them daily scrum meetings. At the scrum meetings, we discussed the team plan, the problems that members were facing, and the solutions that came up. We plan our whole project according to the number of two sprints.

When developing the system, we used version control for simultaneous work on our project. We used Git as our version control. Because of this version control,

everyone has access to new modifications and the latest code for the project. Each member was working on their own branches. Commit History of members is showing in the figure 14

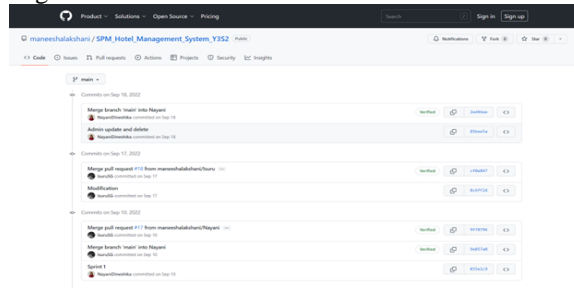


Figure 14: Commit History

We used azure boards for organized our work item. It supported track the tasks, features and bug in the project. It is easy to run by using product backlog or Kanban board. The user interface makes it simple to build and personalize Kanban boards, task boards, and delivery plans. And also Backlog items of the Sprint 1 shown in the figure 15.

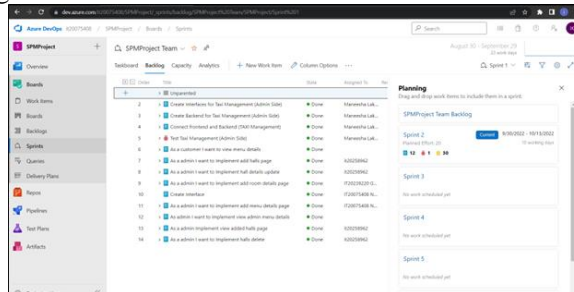


Figure 15: Backlog items of the Sprint 1

For checking the Code Quality Assurances tool, we used SonarQube. It collects and analyzes source codes as well as provides reports about the code quality of a project. Overview status of analyzed projects shown in figure 16.

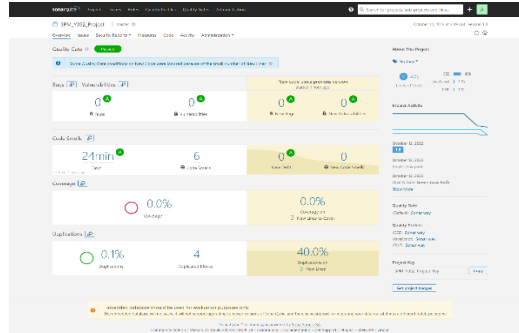


Figure 16: Overview status of analyzed projects

For the testing of the app, we used Selenium, which shows live automated tests being performed on a

computer screen. We did get some test cases and tested them using selenium. Successfully admin login test is shown in the figure 17.

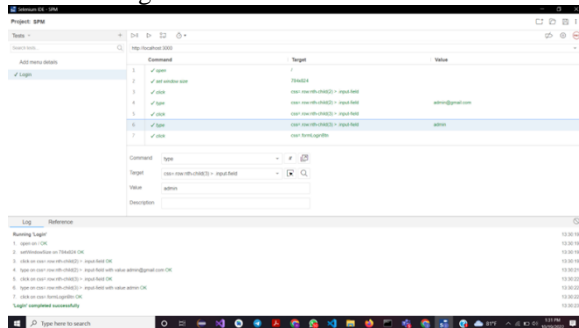


Figure 17: Successfully admin login

#### D. Access privilege

When admin logs in to the system with admin login credentials, admin can access all the managing tasks in the system. And the customer login with customer credentials will not be able to access any admin functions. This privileged access allowed us to secure the application. Both admins and customers can logout of the system, and any user can't register with the same password for the system.

## IV. DISCUSSION

We faced many challenges while building this application such as setting goals before build the system, designing UI as user friendly, making the application easy to navigate, making sure all functions on the website works well and then integrate the application.

As the solutions we research for specific goals based on the vision behind the topic of web application. When goals were set, made sure everyone in the group discussed how to build the application according to fulfill the goals.

Always focus on the user-friendliness of the application. Researched other web application with the same functionality and researched user reviews of those applications.

The application was integrated from the beginning and the problem was solved as soon as possible. Issues arising while building the application did not proceed with building the application.

The manual system for hotel management systems always takes more time and costs more. From this web application, clients can do tasks in the hotel industry without experiencing more complexity. This application always focuses on the main problems that users face in the hotel management system. So, this application makes it very easy for the administrator to provide services to customers, and customers can easily obtain services.

## V. CONCLUSION

In a conclusion, we believe that this hotel management system fulfilled its goals. This system makes customer-hotel interaction more effective, fast, transparent, and trustworthy. As a hotel, they were able to attract more clients than ever before since their online presence grew and people were able to quickly share and recommend them. Furthermore, the hotel was able to handle all its functions since they could see what was available and what was reserved. They might also simply update or introduce new options to clients. Clients were able to reserve rooms and other facilities simply even without physically being there. This made traveling easy for the customer. Overall, the system enabled both the hotel and the customer to manage and complete their tasks with a single click, saving money on paper, phone calls, and pre-hotel visits. The system also saved time. As a result, this approach has proven to be a valuable tool for both sides.

## REFERENCES

- [1] Y. Jingda. (2013). Research and design of hotel management system model. *International Conference on Education Technology and Information System*, pp. 1140.
- [2] Sanni Abubakar Omuya Ogirima, Tolulope Reuben Awode & Olajide Olusegun Adeosun. (2014). Online computerized hotel management system. *J. Comput. Biosci. Eng.*, 2348-7321.
- [3] <https://nodejs.org/en/docs/>.
- [4] G. LIM. (2021). *Beginning MERN stack: Build and deploy a full stack MongoDB, Express, React, Node.js App. Independently Published.*
- [5] "Hooks API Reference," [Online]. Available: <https://reactjs.org/docs/hooks-reference.html>. [Accessed 12 August 2022].
- [6] "Introducing Hooks," [Online]. Available: <https://reactjs.org/docs/hooks-intro.html>. [Accessed 22 July 2022].
- [7] "Routing," [Online]. Available: <https://expressjs.com/en/guide/routing.html>. [Accessed 15 August 2022].
- [8] "About pull requests," [Online]. Available: <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests>. [Accessed 20 August 2022].
- [9] "MERN Stack Authentication Tutorial (Part 1 - The Backend)," [Online]. Available: <https://dev.to/salarc123/mern-stack-authentication-tutorial-part-1-the-backend-1c57>. [Accessed 12 September 2022].
- [10] <https://reactjs.org/docs/getting-started.html>.
- [11] "Welcome to the MongoDB Documentation," [Online]. Available: <https://www.mongodb.com/docs/>. [Accessed 20 July 2022].