



Hotel Management System

CNET343SL Distributed Systems | Report

Ms. RAMAS Rambukwella

Group 40

Name:

Student Reference Number:

Module Code: CNET343SL	Module Name: Distributed Systems
Coursework Title: Hotel Management System	
Deadline Date: 21st April 2021	Member of staff responsible for coursework: Mr. Rasika Alahakoon
Programme: BSc (Hons) Software Engineering	
Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook .	
<p>Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.</p> <p>10673081 D S Bartholomeusz 10673098 D A K H Dedigamuwa 10673125 M G A Geethanjana 10673289 R A M A S Rambukwella 10638325 A V Rajapaksha</p> <p><i>We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.</i></p> <p>Signed on behalf of the group: D S Bartholomeusz</p>	
<p>Individual assignment: <i>I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.</i></p> <p>Signed :</p>	
<p>Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.</p> <p>I *have used/not used translation software.</p> <p>If used, please state name of software.....</p>	
<p>Overall mark ____% Assessors Initials ____ Date _____</p>	

Table of Contents

List of Figures.....	4
List of Tables.....	5
1.0 Introduction.....	6
1.1 Project Scope.....	7
1.1.1 Mobile Application	7
1.1.2 Web Application.....	7
1.2 Project Objectives.....	7
2.0 Business Case.....	8
2.1 Business Needs of the Project	8
2.2 Business Aims and Objectives	8
3.0 System Requirements.....	9
3.1 Functional Requirements.....	9
3.1.1 Web Application - Admin.....	9
3.1.2 Web Application - Counter	10
3.1.3 Web Application - Cashier.....	10
3.1.4 Web Application - Chef	10
3.1.5 Mobile Application - Customer.....	10
3.2 Non-Functional Requirements.....	10
4.0 System Design.....	11
4.1 System Architecture Diagram.....	11
4.2 Entity Relationship Diagram.....	13
4.3 Class Diagram	14
4.3.1 Web Application.....	14
4.3.2 Mobile Application	15
4.4 Use Case Diagram	16
5.0 Tools and Technologies	17
5.1 Middleware.....	17
5.1.1 Selection and Justification.....	17

5.2	API.....	18
5.3	Templates Used	18
5.4	Components Used.....	18
5.4.1	JavaScript Object Notation (JSON)	18
5.4.2	Server	19
5.4.3	Frontend Technology.....	19
5.4.4	Backend Technology	19
5.4.5	Database.....	19
6.0	Hardware and Software Requirements	19
6.1	Hardware Requirements.....	19
6.2	Software Requirements	19
7.0	System Development.....	20
7.1	Web Application.....	20
7.2	Mobile Application	26
8.0	Issues Faced and Approach of Resolving.....	32
9.0	Conclusion.....	33
9.1	Summary.....	33
9.2	Future Implementations	33
10.0	References	34
11.0	Team Details and Contribution	35

List of Figures

Figure 1. System Architecture Diagram.....	12
Figure 2. Entity Relationship Diagram	13
Figure 3. Web Application Class Diagram	14
Figure 4. Mobile Application Class Diagram	15
Figure 5. Use Case Diagram.....	16
Figure 6. Apache Tomcat Logo.....	17
Figure 7. Web App - Common Login Page for Admins, Counters, Cashiers, and Chefs.....	20
Figure 8. Web App - Admin Home Page.....	20
Figure 9. Web App - Notification Manager Page.....	21
Figure 10. Web App - Create a new notification message.....	21
Figure 11. Web App - New Food/Meal Category Creating Page.....	22
Figure 12. Web App - Check for new Food Order Page	22
Figure 13. Web App - Food Order History Page	23
Figure 14. Web App - Create a new Ad Page.....	23
Figure 15. Web App - Check Ratings/Feedback from Customers Page.....	24
Figure 16. Web App - Generate Report Page.....	24
Figure 17. Web App - Room Service Orders from Customers Page	25
Figure 18. Mobile App - Landing Page	26
Figure 19. Mobile App - Home Page	26
Figure 20. Mobile App - Calling for Room Service	27
Figure 21. Mobile App - Searching for Room Service	27
Figure 22. Mobile App - Search Option for Hot Dish	28
Figure 23. Mobile App - Search Results for Hot Dish	28
Figure 24. Mobile App - Quantity Selection for Hot Dish	29
Figure 25. Mobile App - Confirm Order Page.....	29
Figure 26. Mobile App - Confirm Food Order Room Page.....	30
Figure 27. Mobile App - Current Food Orders Page.....	30
Figure 28. Mobile App - Food Order History Page	31
Figure 29. Mobile App - Feedback Page	31

List of Tables

Table 1. Middleware Selection and Justification	18
Table 2. Issues Faced and Approach of Resolving	32

1.0 Introduction

Hotels are very well known as a place for accommodation for guests. People tend to use hotels as their accommodation to stay when they travel either for leisure or for other purposes. Hotels offer various facilities to their customers such as relaxation, meals from breakfast to dinner, entertaining events for visitors, accommodation to stay, and some facilities to conduct functions and gatherings. The exact specifications and services offered to customers can differ significantly from a hotel to the next hotel. Hotel managing entails managing all aspects of the establishment. This necessitates a thorough understanding of customer relations, personnel management, promotion, and distribution plan among other things. Hotel management entails assessing results in all aspects of the sector regularly and making appropriate changes.

The quality and the appearance of a hotel are sharpening from the rates and feedback given by its customers. To capture the confidence of the customer, the hotel must provide a better service, retain a high-quality outlook in every aspect. The quality of operation of a hotel varies from the way it provides its services to clients. It is also important for them to maintain a proper means of dealing with the customer to satisfy the needs of the customer when they reach the hotel. Efficient hospitality industry will, in the end, mean that the hotel not only remains in operation but also profits and multiplies exponentially. As a result, hotels must migrate from their outdated methods of approaching customers to the most up-to-date technological methods. Customers will have a more comfortable stay as a result of the improvement, rather than stressing about the services they need.

Therefore, this delivery hotel management system was put in place to address all the basic needs that appear when staying in a hotel. The implemented distributed system consists of a web application and a mobile application that is specially designed for different users such as customers, hotel admins, counters, cashiers, and chefs.

The mobile application was especially implemented for customers to do multiple functions such as order food, call for room services, check for other hotels, etc. The benefit of this mobile application is that the customer can enter their needs through an equipped smart device (a mobile phone/ tablet) with the application inside the room in the hotel. The customer does not need to carry the application on their personal mobile device.

The web application is for the admins, counters, cashiers, and chefs in the hotel. The web application portal maintains the administration functions to link the necessary actions between the customers and the system with other services such as connecting to the available worker to fulfill room services, accepting and connecting food orders to the chefs, finalizing the bills and food orders, etc. Through the web application, the admin can manage food orders, manage, manage advertisements, manage notifications and messages, etc. The counter

manages all the orders by the customers and directs the orders to chefs. The cashier manages and finalizes all the bills and chefs can manage and deliver food orders.

1.1 Project Scope

1.1.1 Mobile Application

The mobile application is designed and developed for the visitors who check-ins to the hotel. The customers have the authority to use the smart device placed inside the hotel room to request room service, search and order the food they like, maintain a list of favorite food, look for similar hotels, review ads from the admins, receive notifications, and review their order history.

1.1.2 Web Application

The web application is designed and developed for the usage of hotel admins, counters, cashiers, and chefs. Different employees have a different set of authentications over their designation. The login is created according to their profession and by then the web application displays only the necessary information on particular fields. For instance, the admin has whole access to the entire functionality of the web application while the counter maintains the transactions of the food orders to chef and service calls to works, and cashier handles and generates all the bills, and chefs see the customer's favorite meals the counter has directed from the ordered customer.

1.2 Project Objectives

The system mainly functions the facilities below.

- Visitors can use a smart device located inside their room to,
 - Request a room service,
 - Search for appropriate meals and food,
 - Order food,
 - Add favorite meals to a list,
 - Leave feedback of the hotel,
 - Receive notifications from the hotel admins,
 - Get information on their food orders and time estimation of receiving.
- Administration and hotel staff have better knowledge of,
 - Free rooms,
 - Customer's food orders,
 - Room service needed rooms,
 - A proper review of individual feedback from the customers,
 - and other stuff since all of them are recording in the database.

2.0 Business Case

2.1 Business Needs of the Project

Hotels are the most common resorts of tourists and families who enjoy spending vacation times. Visitors like to spend their leisure time without asking a waiter or a service operator to bring something or where to go next. Visitors expect to have their privacy as comfortable as in their homes. Visitors also expect decent customer service and should not expect to do business on their own. Most hotels do not have the option for their customers to order items to their rooms or function areas. To do that, the customer needs to engage with their room service facility or contact the operation unit of the management or walk into the reception to inform their needs and wants.

In that case, the precious time and resources of the customer will be wasted, and the service given to the customer will be disappointed. Customers usually come to the hotel to spend their spare time enjoying themselves. Therefore, an unmanned operator should need to bridge this gap between the customer and the hotel service. Every hotel owner tries to deliver the best hospitality to their customers and an easy environment for their employees to work. To please the client and elevate the hotel to higher heights, the existing operating system may have to be updated.

2.2 Business Aims and Objectives

This where the businesses should adopt to modern technology or automate their traditional ways of managing hospitality services. This modern implementation, which uses a distributed network, will cover all aspects of hotel management and provide essential services to customers from the time they arrive until they depart the venue. Below shows the objectives of the newly implemented system to manage the functions of the hotel.

- By introducing the new method, the hotel can enhance the efficiency of the operations of the hotel.
- The food waste can be cut down by admins keeping proper control over the food orders.
- The managers can review the information and generate reports to make decisions and judgments since all operations and transactions are stored inside a local server.
- The new system will assist in providing excellent customer service and increase customer loyalty over the hotel.
- Through processing productivity, the executives can reinvest the cost of hiring new employees.

- This new process will make the hotel a relaxing environment for both visitors and employees.

3.0 System Requirements

3.1 Functional Requirements

3.1.1 Web Application - Admin

Admin can log into the system using given credentials and access all the available operations of the system including;

- **User Management**
 - Ability to add and remove admins from the system.
 - Edit user details
- **Dish Management**
 - **Category Management** where admins have the authority to add a meal category to the hotel system, search a category from the system, edit the category, and delete it from the system.
 - **Dishes Management** where admins can add, edit, and delete dishes from the system, search for dishes, and enable or disable the availability of the dish.
- **Order Management**
 - Admins have the authority to check for **new food orders** from the customers.
 - Admins can view the **confirmed orders** by the customers, cashiers and other admins of the system.
 - Admins can view the orders that has been **delivered** from the chefs to customers.
 - Admins can check for an **order history** received from the customers of the system.
- **Ads Management**
 - Only admins can add a **new advertisement** to the system to display on smart devices provided to the customers.
- **Greeting Message Management**
 - Admins can send **quick message** to the smart devices provided to the customers.
- **Feedback**
 - Admins have the authority to view all the **ratings and feedbacks** from customers.
- **General Configurations**
 - All settings of **tax** where a charge is calculated, settings of **table numbers** and where to allocate them and other **unit** setting can be access by the admins.

3.1.2 Web Application - Counter

- Counter staff can view new orders receiving from the customers and confirm and direct them to either chefs or admins to process with the functions.
- To get the necessary authentication to operate the facility, the counter needs to log into the system using his/her given credentials.

3.1.3 Web Application - Cashier

- Cashier is responsible and can be accessed to function all the bills and end calculations of all the processes for the customers.
- To get the necessary authentication to operate the facility, the cashier needs to log into the system using his/her given credentials.

3.1.4 Web Application - Chef

- The chef has the access to only view the food orders directed by the counters.
- To get the necessary authentication to operate the facility, the chef needs to log into the system using his/her given credentials.

3.1.5 Mobile Application - Customer

The customer has a vast and variety of facilities delivered from the system. Customer can,

- Order new food
- Select a dish/meal
- Save or add the meal items to a separate favorite list
- Receive notifications and advertisements from the admins of the hotel
- Call for a room services
- View for other similar hotels
- View details of the current hotel
- Leave feedback and ratings

3.2 Non-Functional Requirements

- The hotel administrators oversee all supervision and surveillance, and they can keep track of all hotel activity procedures.
- This network provides excellent and comprehensive care to all of the hotel's stakeholders involved. As a result, the hotel can provide excellent service to its customers.
- Scalability is handling every amount of workload without any issues caused. The system is very stable function upon a large number of users increased.

- Availability - Since the system needs to run 24/7, the hotel is very much aware of the components, hardware and serves that function the system all the time.
- Performance
- Heterogeneity - Customers does not need to worry about using the system on their personal devices since the system is already installed inside the device provided by the hotel.
- Security - Customers are logging into the hotel system using a special mobile device offered by the hotel, which does not require the customer to provide any sensitive information to operate.

4.0 System Design

4.1 System Architecture Diagram

The system architecture of this distributed system defines how it interacts with the hotel's customers and communicates to the API, and as well as how it functions. There is a mobile application as well as four web application portals for various occupations such as admin, counter, cashier, and chef in this hotel management scheme. The database is accessible to all client machines through a network connection. As seen in the diagram below, all client-side devices are linked to the database network via a RESTful API server.

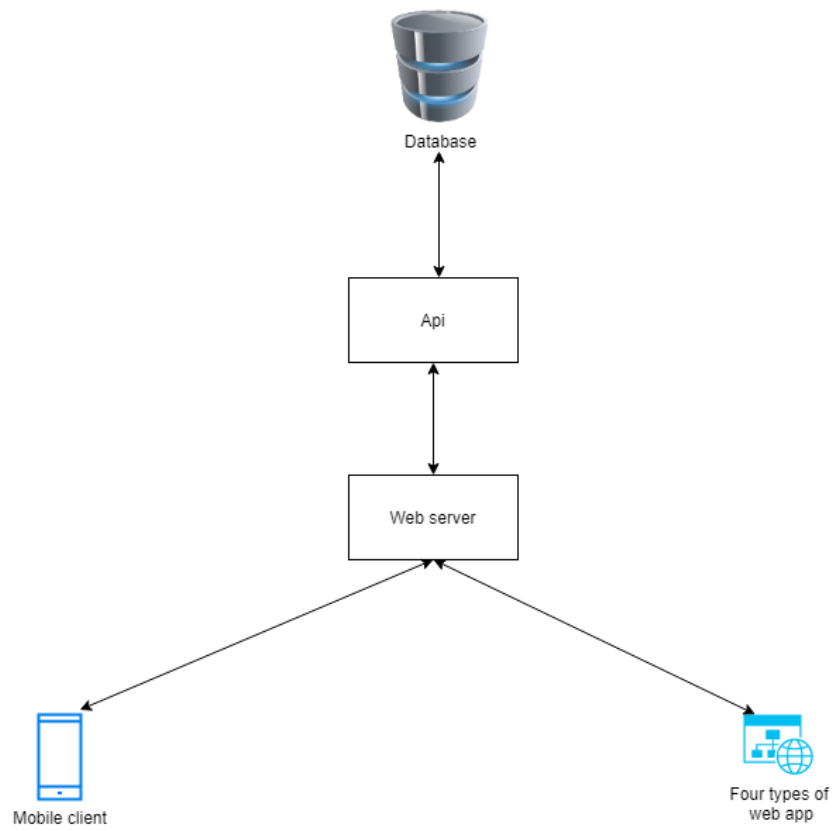


Figure 1. System Architecture Diagram

4.2 Entity Relationship Diagram

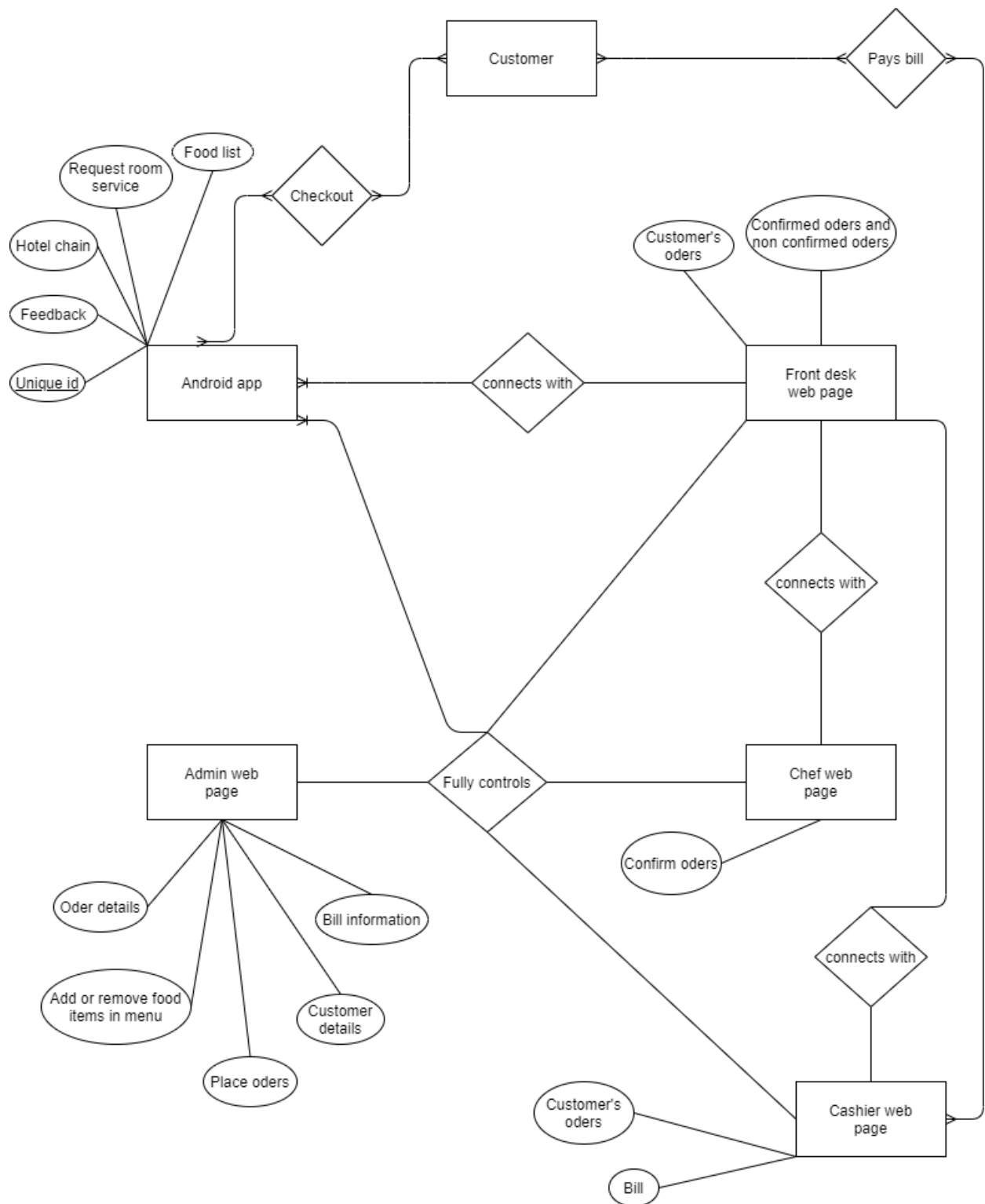


Figure 2. Entity Relationship Diagram

4.3 Class Diagram

4.3.1 Web Application

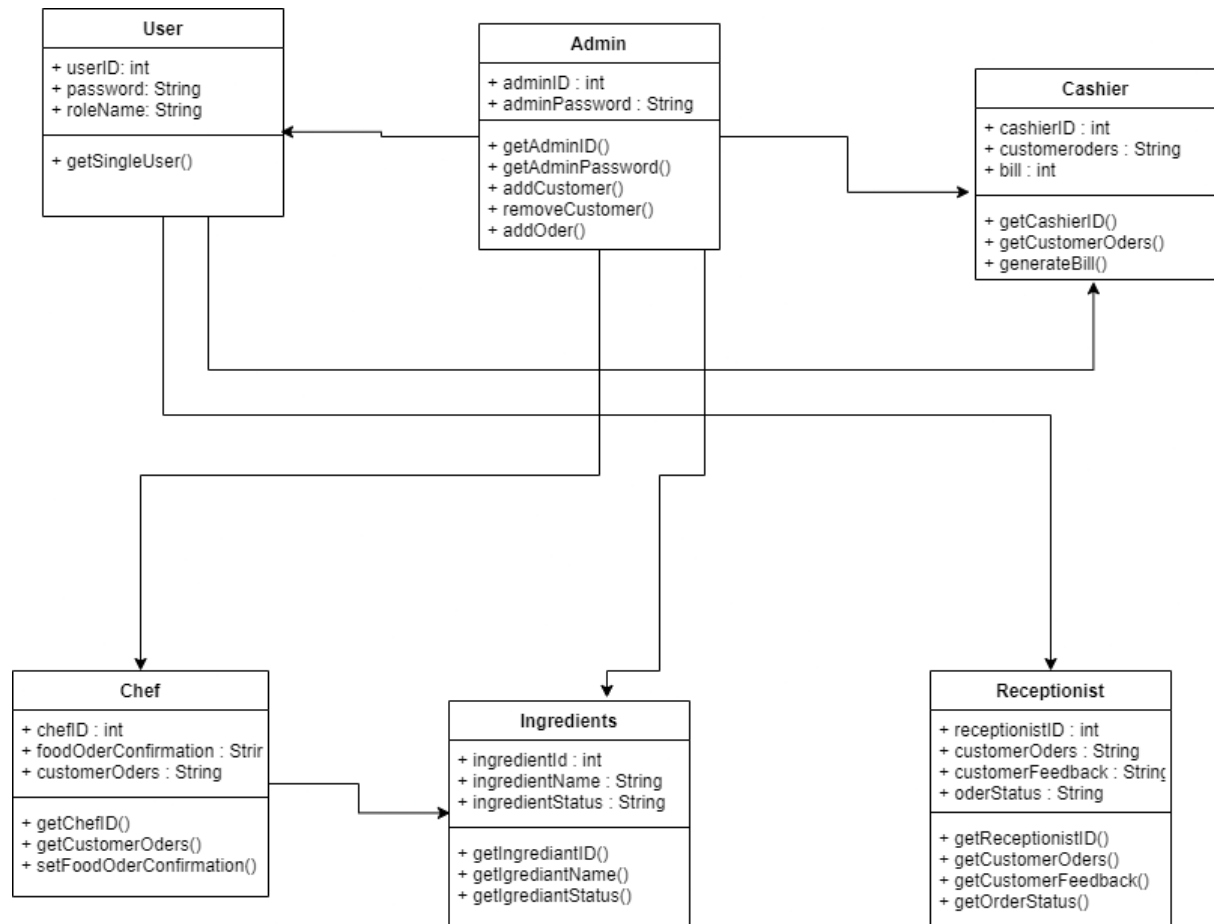


Figure 3. Web Application Class Diagram

4.3.2 Mobile Application

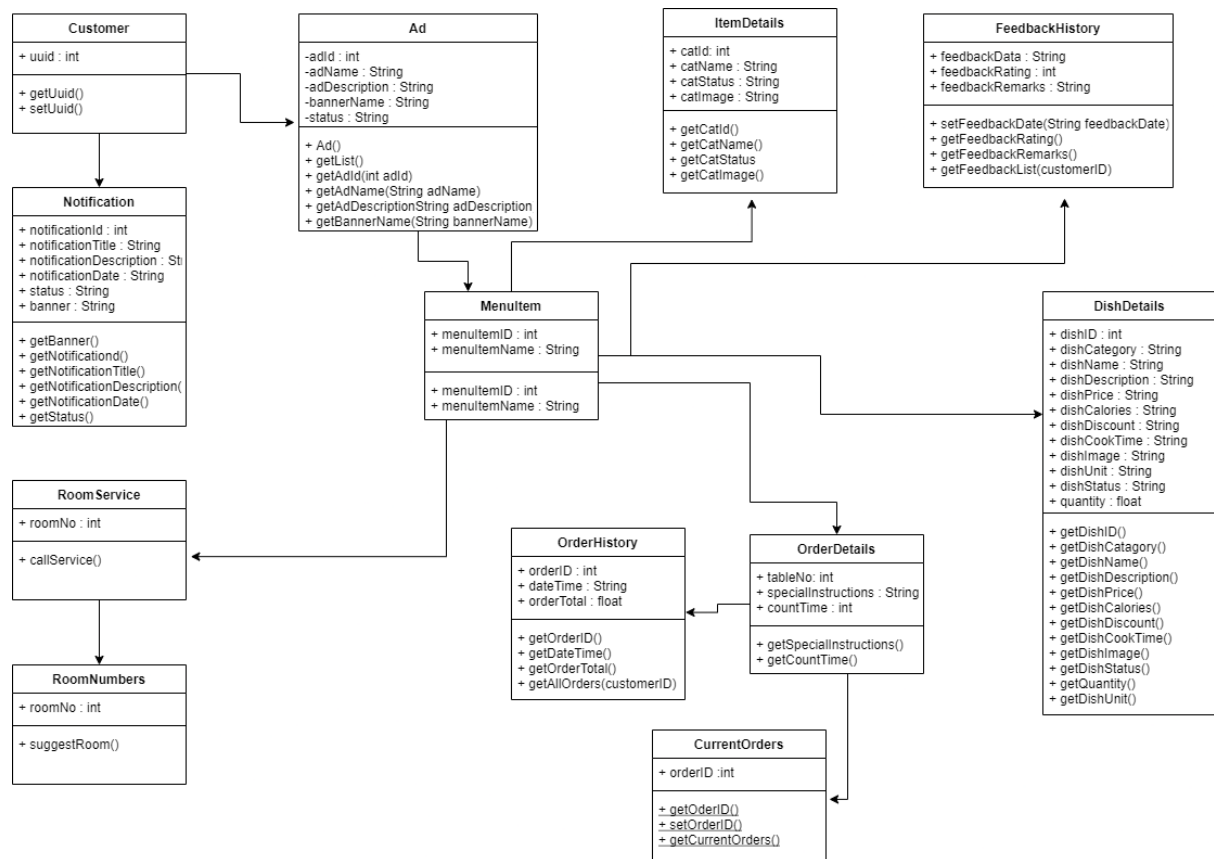


Figure 4. Mobile Application Class Diagram

4.4 Use Case Diagram

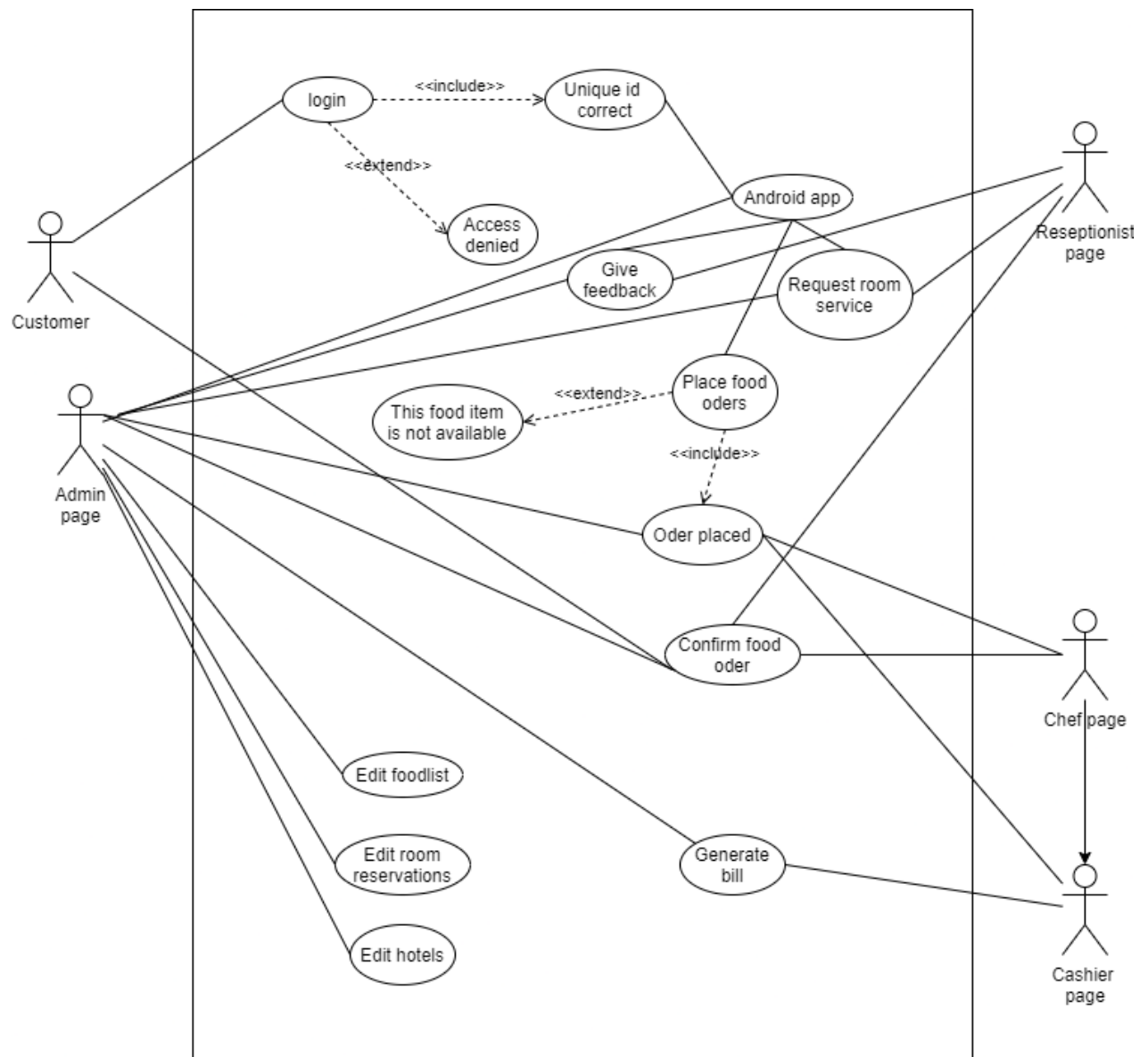


Figure 5. Use Case Diagram

5.0 Tools and Technologies

5.1 Middleware

It is a software layer that is situated in between applications and operating systems. Middleware simplifies software development by hiding the intricacies of distributed applications and hiding the heterogeneity of hardware, operating systems, and protocols. It also provides uniform and high-level interfaces used to make interoperable, reusable, and portable applications and also provides a set of common services that minimize duplication of efforts and enhances collaboration between applications¹.

5.1.1 Selection and Justification

The Apache Tomcat has been selected as the middleware of this distributed hotel management system and the Tomcat server serves as a bridge between two computing systems. Tomcat is a piece of middleware that lies between the client's HTML/JavaScript surface and the server's database layer.



Figure 6. Apache Tomcat
Logo

Tomcat is an application server designed to execute Java servlets and render web pages that use Java Server page coding. Accessible as either a binary or a source code version, Tomcat has been used to power a wide range of applications and websites across the Internet. Below describes few reasons to select Tomcat as the middleware for this project.

Selection	Justification
Lightweight	Tomcat is an extremely lightweight program, also with JavaEE certification. It only contains the absolute necessities of server capabilities, allowing it to launch and redirect faster than many of its competitors. Because of its lightweight, it has a much faster execution time.
Open-Source	Tomcat is a free service, and the source code can be used by anybody who wants it. Tomcat gives developers complete control of what they can do with their Tomcat installation.
Flexibility	Tomcat is very versatile due to its lightweight architecture and a collection of wide, built-in customizable features. Tomcat can be configured in about every way the developer wants, and it will always function properly.
Stability	Tomcat is a very robust framework to develop on and running applications on it can help to keep the server secure. Since Tomcat operates

¹ <https://www.techopedia.com/definition/450/middleware>

	independently of Apache, even if Tomcat fails disastrously, the remainder of the server will continue to function normally.
Security	Many people want to put the Tomcat server under a separate firewall that is only available from the Apache server. Based on how we set up Tomcat, we can apply additional protection to the server, which is still a positive thing.
Well-documented	Tomcat comes with a wealth of helpful resources, such as several free videos that can be accessed or downloaded. As a result, it's a common option for application servers in almost all Java web apps. Tomcat has everyone covered when it comes to initialization configurations, hardening and security manuals, installation guidance, and server setup documentation.

Table 1. Middleware Selection and Justification

5.2 API

The elements of the system are displayed on various platforms as in the proposed system. Each part shall communicate over a network to accomplish the desired outcome. This framework operates on a local server to provide an API for accessing data from the database to web and mobile applications. When customers use the mobile application to enter their requests, the web application will automatically be modified and in which to do so the API will be used to connect both services in real-time. All the orders and ratings are done by the customers will be updated respectively to the local database with the help of the REST APIs.

Since it converts distributed servers into RESTful web services, Representational State Transfer Architecture (REST) has been used for the necessary intention. REST is a platform and technology independent infrastructure. Client-Server, standardized interfaces, stateless, cacheable, layered system, and other architectural constraints are specified for the order to design a Restful web service. The REST architecture aids in the creation of a high-quality distributed system by making it simple to test and read.

5.3 Templates Used

Bootstrap templates have been used to make sure the attractive and user-friendly approach of the page.

5.4 Components Used

5.4.1 JavaScript Object Notation (JSON)

The data between the client-side and the server is managed using JSON objects in this framework. To transfer the data, JSON provides the method, URL, and data as parameters.

Client-side programs are used to capture the encrypted data. JSON is used as a data extraction technology because it is simpler and quicker than XML.

5.4.2 Server

The system is designed to be used in a hotel and runs on a high-performance Tomcat 7.0 server. Apache Software Foundation created Tomcat 7.0, an administration server. It requires the use of Java servlets, which are capable of running websites.

5.4.3 Frontend Technology

The mobile application is implemented using Java programming language. JavaScript technology.

5.4.4 Backend Technology

The backend is built with Java J2EE technology, which is the most stable language for developing APIs. It sends data from the servers to the client computers using the high-performance Tomcat server.

5.4.5 Database

The most common and dependable open-source MySQL database has been used as the database to function this whole process. MySQL is flexible with handling huge amount of data and resource easily and it is one of the many reasons to choose as the database of the system.

6.0 Hardware and Software Requirements

6.1 Hardware Requirements

An android smart device either a mobile phone or a tablet. and four desktops are required to finalize the said system. All the desktops need to be connected with a local host because of the systems' own security.

6.2 Software Requirements

Need to use Tomcat 7.0 server to run the system with a MySQL database and the computers must install Java JRE on it.

7.0 System Development

7.1 Web Application

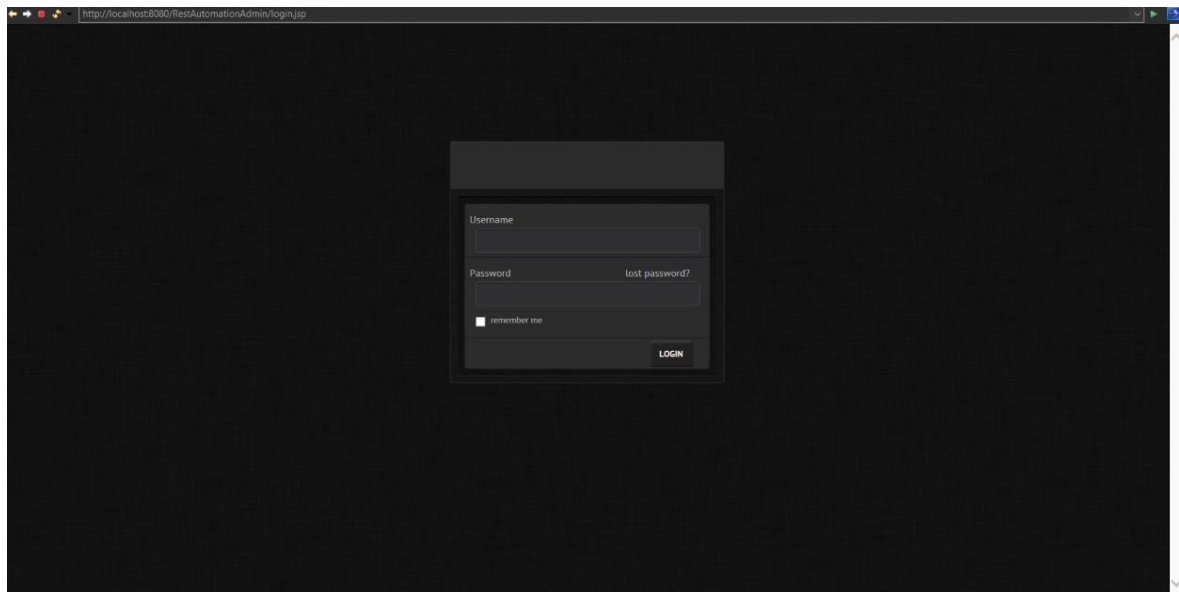


Figure 7. Web App - Common Login Page for Admins, Counters, Cashiers, and Chefs

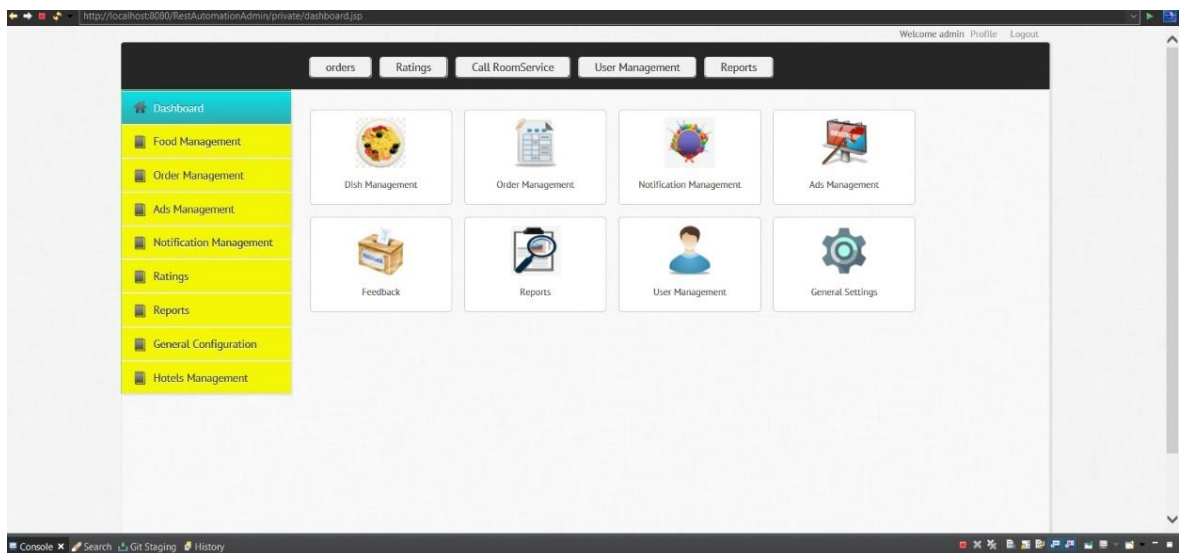


Figure 8. Web App - Admin Home Page

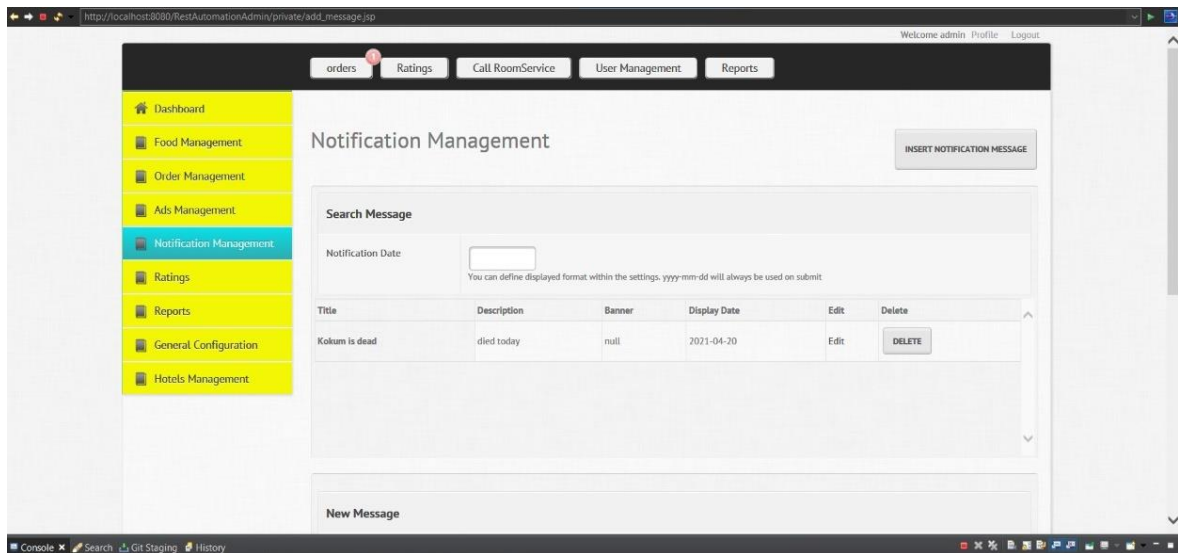


Figure 9. Web App - Notification Manager Page

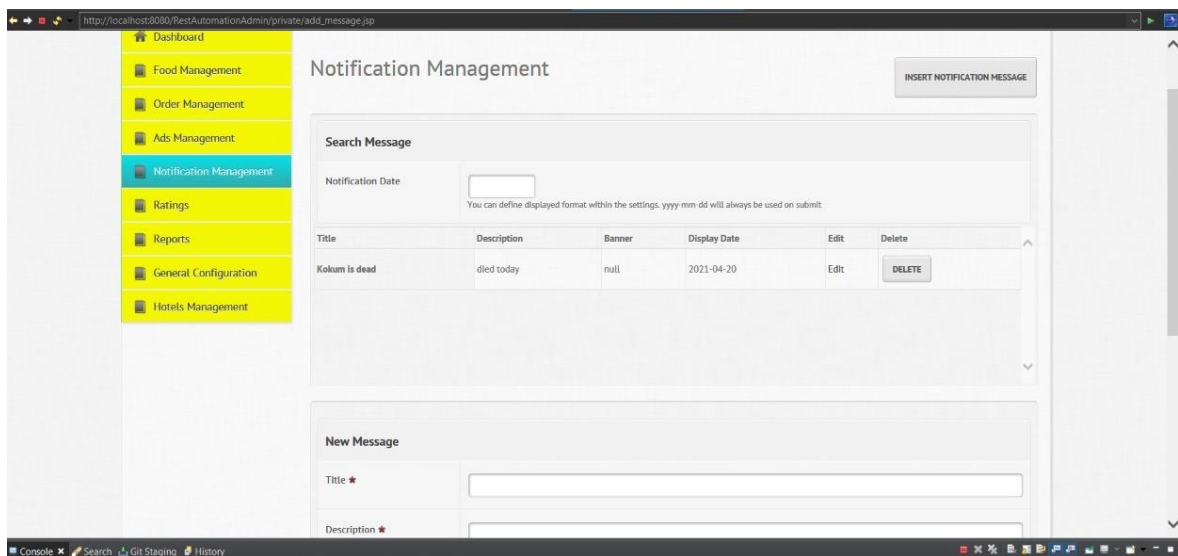


Figure 10. Web App - Create a new notification message

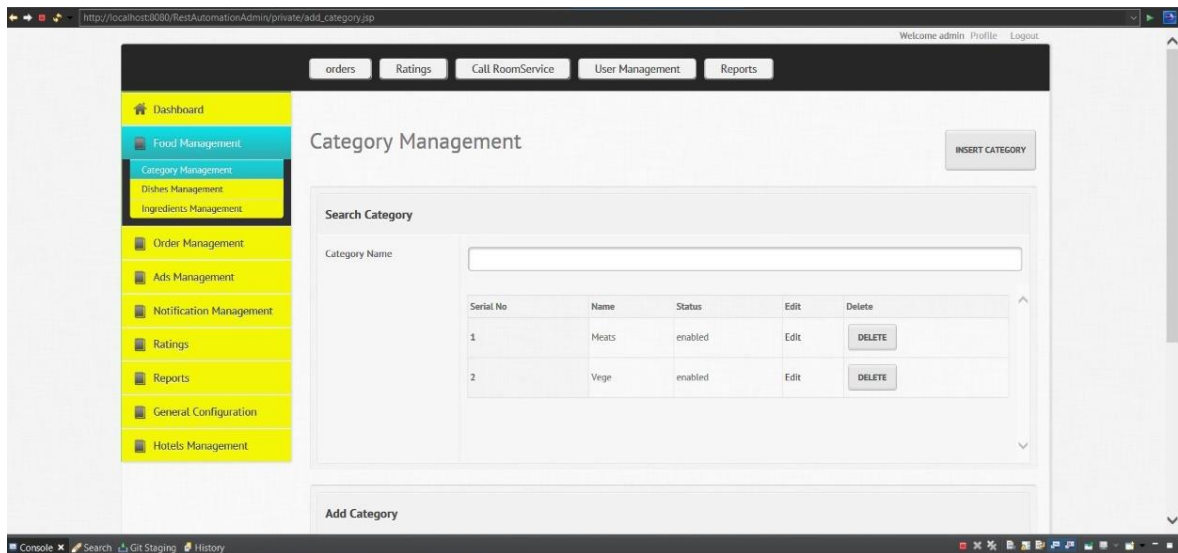


Figure 11. Web App - New Food/Meal Category Creating Page

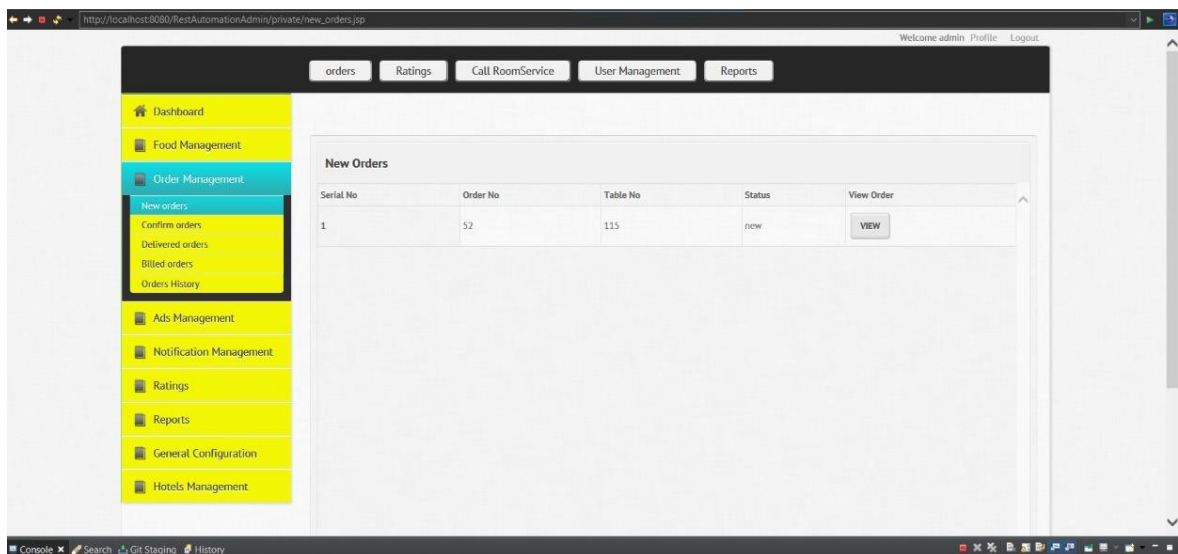


Figure 12. Web App - Check for new Food Order Page

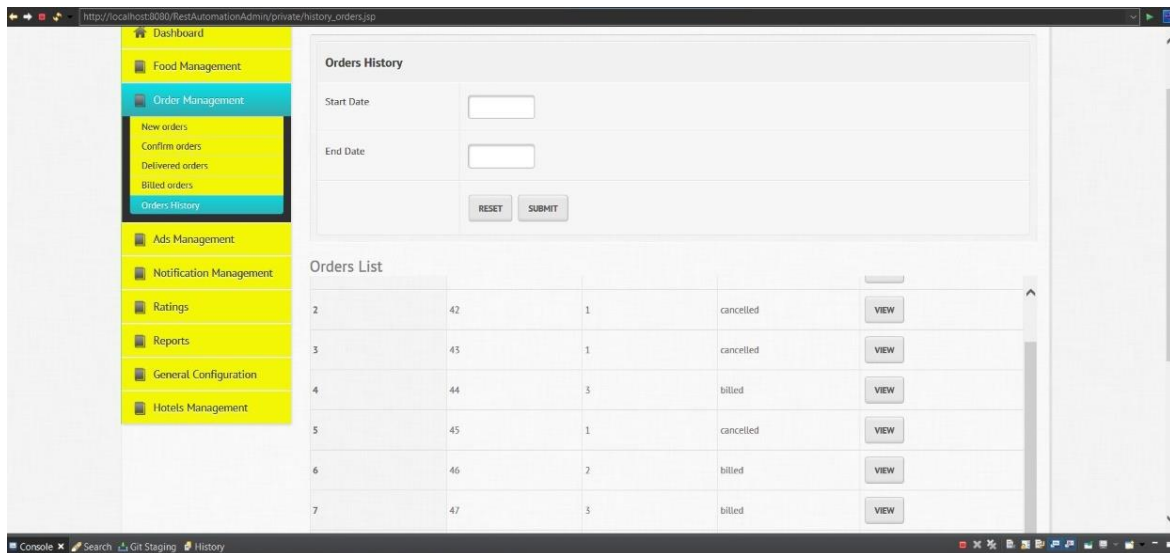


Figure 13. Web App - Food Order History Page

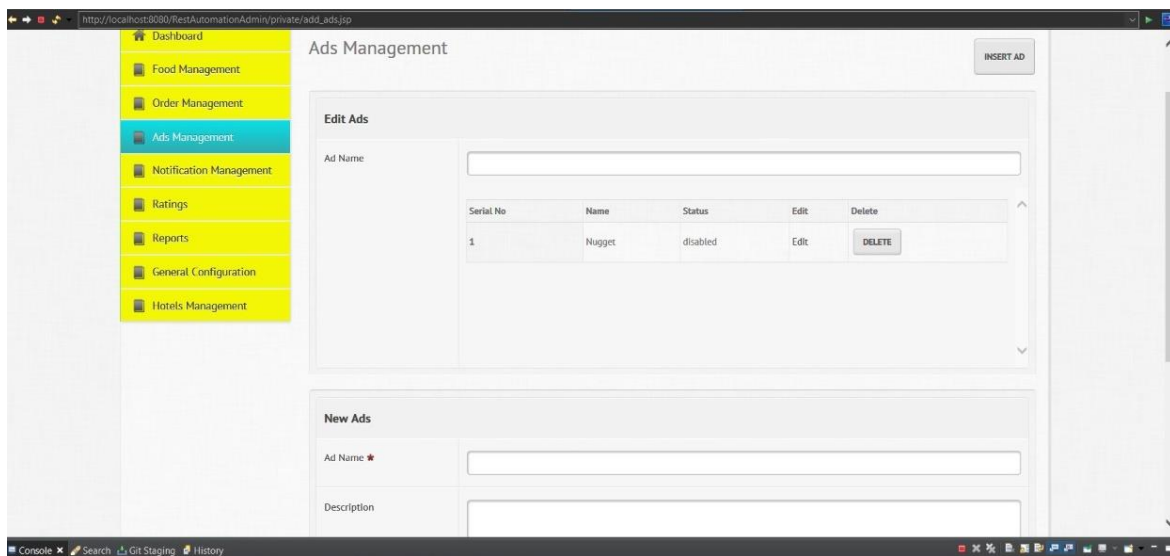


Figure 14. Web App - Create a new Ad Page

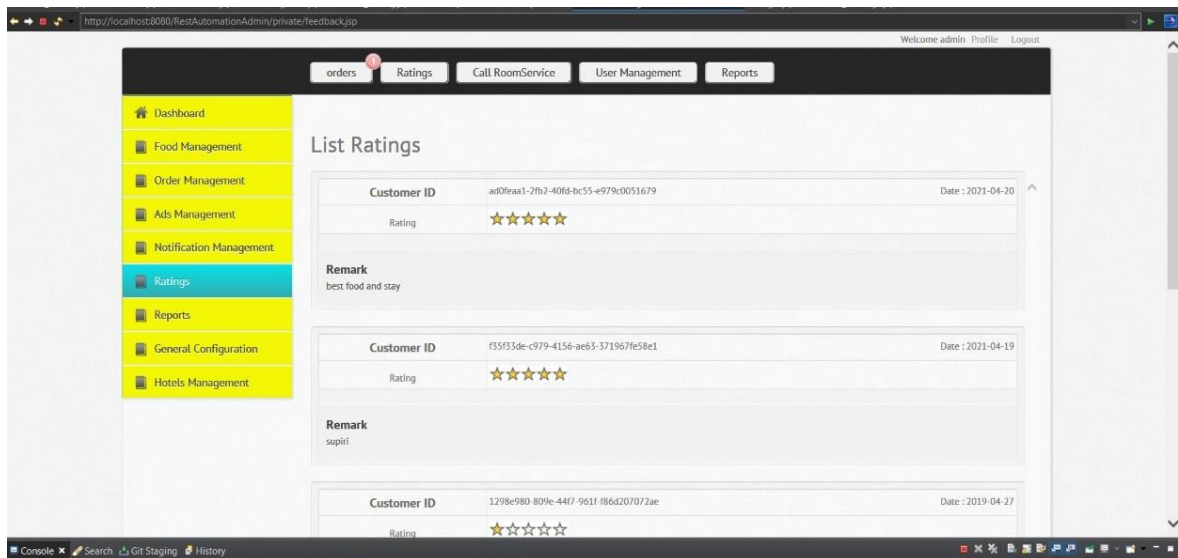


Figure 15. Web App - Check Ratings/Feedback from Customers Page

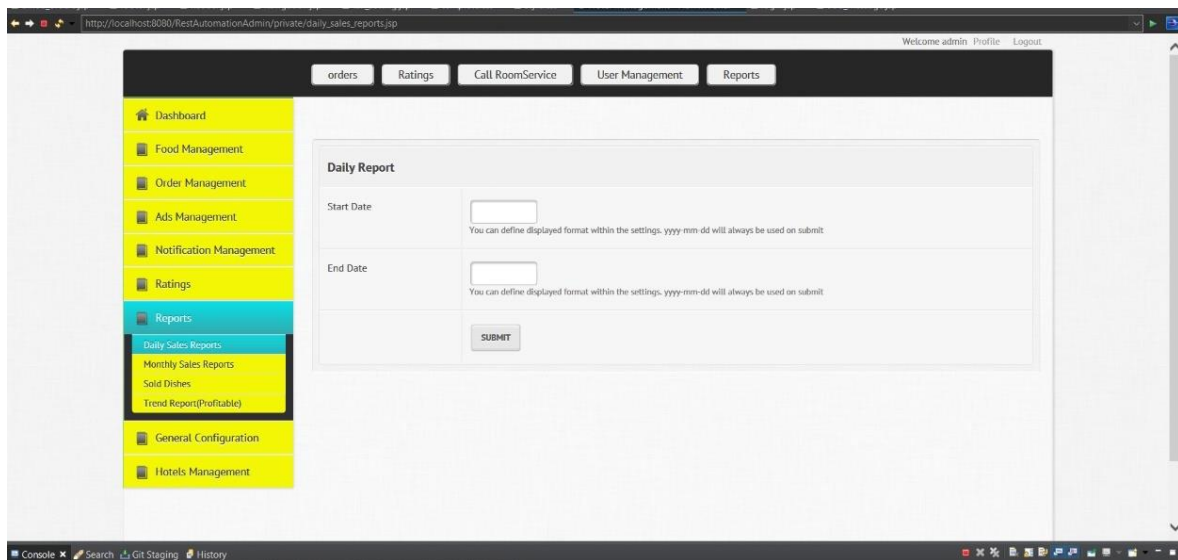


Figure 16. Web App - Generate Report Page

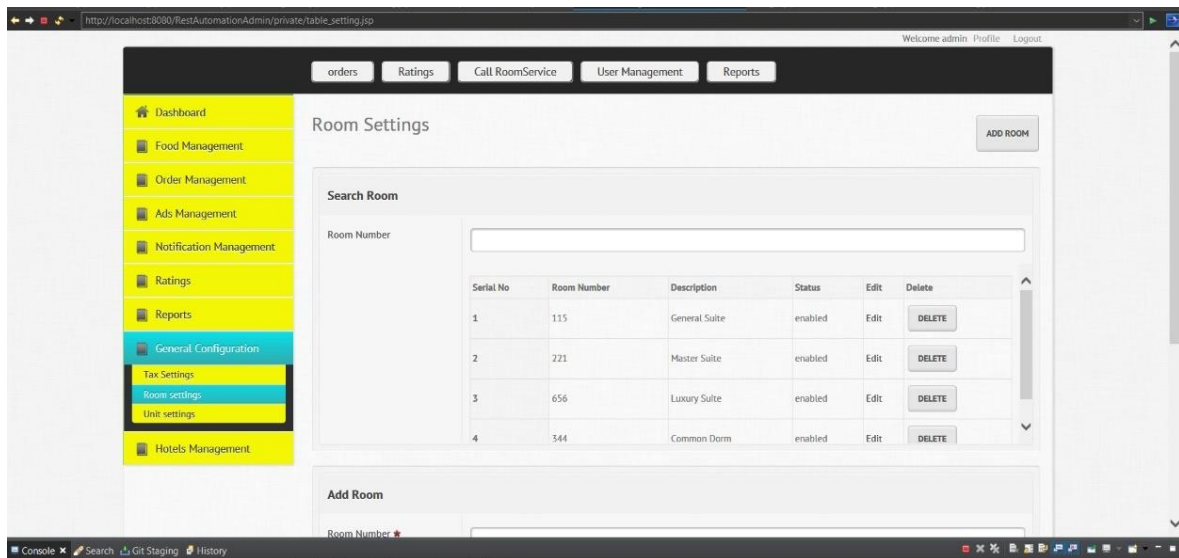


Figure 17. Web App - Room Service Orders from Customers Page

7.2 Mobile Application



Figure 18. Mobile App - Landing Page

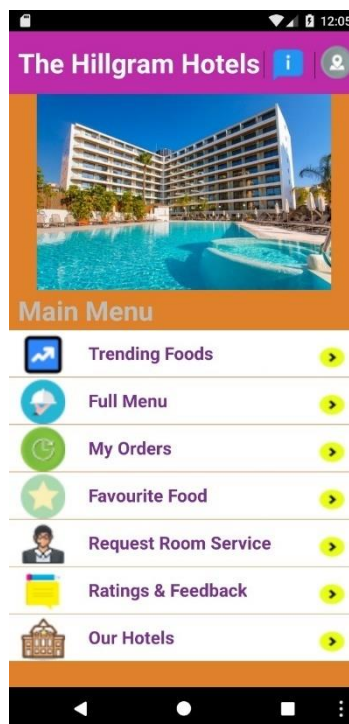


Figure 19. Mobile App - Home Page

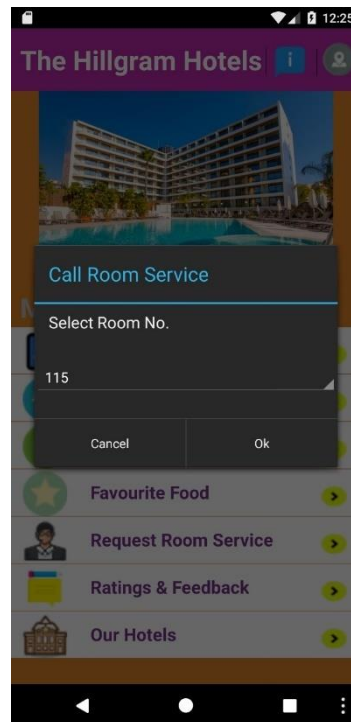


Figure 20. Mobile App - Calling for Room Service

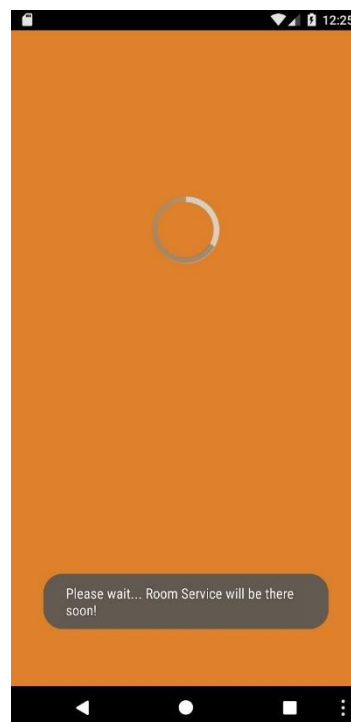


Figure 21. Mobile App - Searching for Room Service



Figure 22. Mobile App - Search Option for Hot Dish

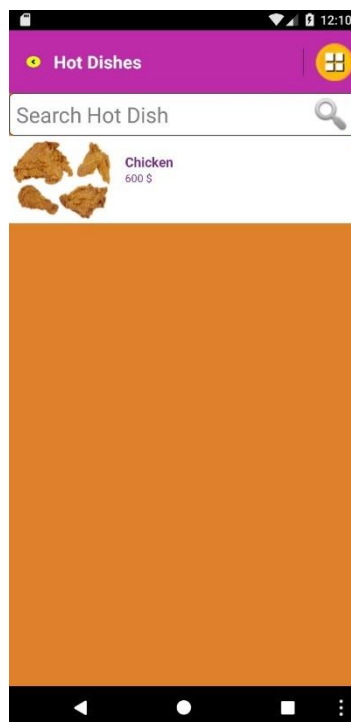


Figure 23. Mobile App - Search Results for Hot Dish

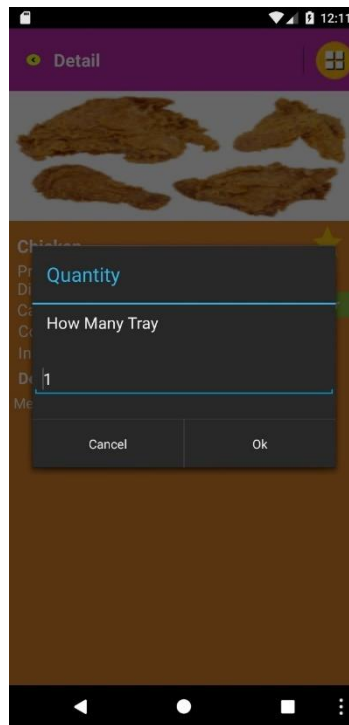


Figure 24. Mobile App - Quantity Selection for Hot Dish

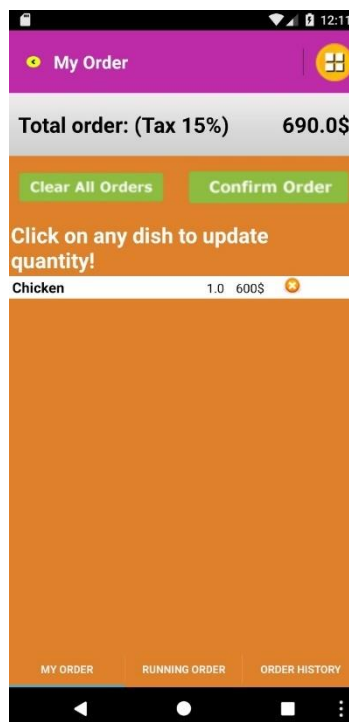


Figure 25. Mobile App - Confirm Order Page

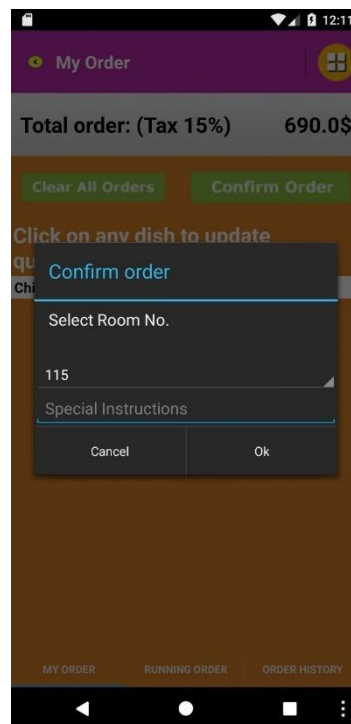


Figure 26. Mobile App - Confirm Food Order Room Page



Figure 27. Mobile App - Current Food Orders Page

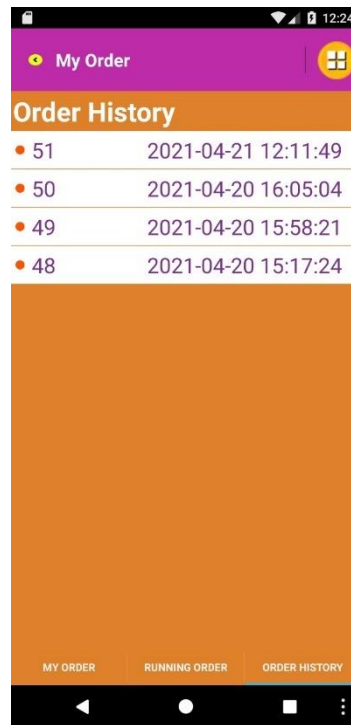


Figure 28. Mobile App - Food Order History Page

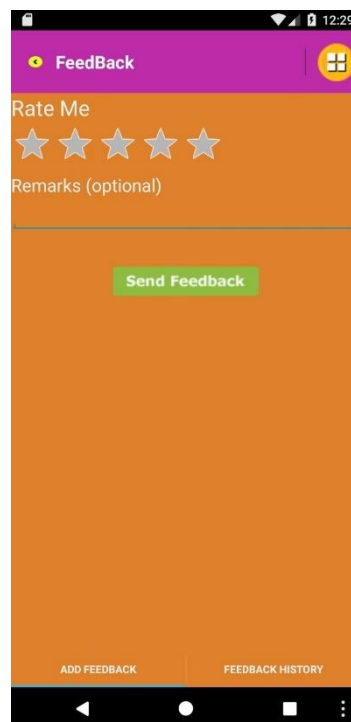


Figure 29. Mobile App - Feedback Page

8.0 Issues Faced and Approach of Resolving

Issue No	Issues Faced	Approach of Resolving
1	<p>The initial middleware “MongoDB” of the proposed plan had to be changed due to its limitations and restrictions with the proposed solution to implement the Hotel Management System.</p> <p>Some restrictions and limitations of the proposed middleware:</p> <ul style="list-style-type: none">• Uses high data memory to store data and therefore, the application will occur a latency.• No transaction support.• Data duplication is high, and it is hard to keep track and get notified.• Hard to combine documents in MongoDB.• Several queries need to pull data from collections, and it will invariably result in disorganized code and lengthy processing times (SaasHub, 2021).	<p>With all due facts, Tomcat 7.0 was chosen as the middleware to implement the Hotel Management System. Since Tomcat is purely Java based application as well as a Java Servlets and consist with JSP, it was lot easier to work with implementing the Java based web and mobile application to the said system.</p>
2	<p>Data failed to pass through the Tomcat server at the initial stage of implementing the system.</p>	<p>To fix the issue, we had to manually insert the Tomcat .jar files to the build config of the backend of the server.</p>
3	<p>Mobile application crashes when upon null entry of images in the database.</p>	<p>We had to insert dummy images to the database when there are no images in the database.</p>

Table 2. Issues Faced and Approach of Resolving

9.0 Conclusion

9.1 Summary

This Hotel Management System was developed with the expectations of travelers who enjoy staying in hotels in opinion, as well as the convenience of hotel staff members in maintaining and providing proper hospitality service to their guests. The implementation will address all of the drawbacks of existing hotel systems, which still rely on outdated methods for managing customer-staff interactions rather than new technologies. The system's distributed network manages the customer-staff relationship's environment, ensuring a seamless flow from the customer's arrival to departure. The built system provides complete transparency to the customer and makes it very easy for them to enjoy their stay.

9.2 Future Implementations

Since this distributed project must be achieved within a certain time frame, some functionalities must be limited, however, certain suggestions were made to expand this system into a full hotel-chain management system.

We hope to develop another client-side application to place hotel reservations before coming to the hotel and a portal where customers can pay their hotel bills using the same smart device available inside their rooms.

For the moment, customers cannot cancel a food order once they confirm the order. However, the implementation should need to be developed for them to cancel a food order at a given time upon their change of taste.

10.0 References

Beatty, C., 2020. *Apache Tomcat: An Introduction (And How It Can Benefit You)*. [online] Available at: <<https://www.a2hosting.com/blog/apache-tomcat-introduction/>> [Accessed 19 Apr. 2021].

Bodkhe, S., 2018. *How to Develop RESTful Web Services Using MySQL and Spring Boot - DZone Integration*. [online] Available at: <<https://dzone.com/articles/application-using-spring-boot-restful-webservices>> [Accessed 19 Apr. 2021].

Cookson, R., 2019. *The Pros and Cons of MongoDB - Virtual-DBA Remote DBA Services & Support - Certified Database Experts*. [online] Available at: <<https://www.virtual-dba.com/blog/pros-and-cons-of-mongodb/>> [Accessed 19 Apr. 2021].

JTB Business Travel, 2020. *What Business Travelers Want From Hotels*. [online] Available at: <<https://www.jtbusinessstravel.com/what-business-travelers-want-in-hotels/>> [Accessed 19 Apr. 2021].

Lam, M., 2014. *Java EE 7 / JAX-RS 2.0: Simple REST API Authentication & Authorization with Custom HTTP Header*. [online] Available at: <<https://www.javacodegeeks.com/2014/10/java-ee-7-jax-rs-2-0-simple-rest-api-authentication-authorization-with-custom-http-header.html>> [Accessed 19 Apr. 2021].

SaaSHub, 2021. *Apache Tomcat VS MongoDB - compare differences & reviews?* [online] Available at: <<https://www.saashub.com/compare-apache-tomcat-vs-mongodb>> [Accessed 19 Apr. 2021].

Techopedia, 2021. *What is Middleware? - Definition from Techopedia*. [online] Available at: <<https://www.techopedia.com/definition/450/middleware>> [Accessed 20 Apr. 2021].

TrustRadius, 2021. *Pros and Cons of Apache Tomcat 2021*. [online] Available at: <<https://www.trustradius.com/products/apache-tomcat/reviews?qs=pros-and-cons>> [Accessed 19 Apr. 2021].

11.0 Team Details and Contribution

Index Number	Name	Email
10673081	D S Bartholomeusz	10673081@students.plymouth.ac.uk
10673098	D A K H Dedigamuwa	10673098@students.plymouth.ac.uk
10673125	M G A Geethanjana	10673125@students.plymouth.ac.uk
10673289	R A M A S Rambukwella	ramasrambukwella@students.nsbm.lk
10638325	A V Rajapaksha	adip.rajapaksha@students.plymouth.ac.uk

Individual Contribution	10673081	10673098	10673125	10673289	10638325
Proposal and Report writing	✓	✓	✓	✓	✓
UI/UX Design	✓	✓	✓	✓	✓
Web application development	✓	✓	✓	✓	✓
Mobile application development	✓	✓	✓	✓	✓
Web services	✓	✓	✓	✓	✓