

Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre



A Project Report
on
“Smart Restaurant”
[Course Code: COMP 206]

(For partial fulfillment of 2nd Year/ 1st Semester in Computer Engineering)

Submitted by:

Ashish Adhikari (04)

Shusant Sapkota (43)

Gaurav Singh Thagunna (55)

Submitted To:

Nabin Ghimire

Department of Computer Science and Engineering

Submission Date:

February 1, 2018

Bona Fide Certificate

**This project work titled as
“Smart Restaurant”
has been submitted to the Department of Computer Science and Engineering
for the partial fulfillment of COMP 206 as a 2nd year 1st semester project.**

**This project work is the bona fide work of
“Ashish Adhikari, Gaurav Singh Thagunna and Shusant Sapkota”
who carried out the project work under my supervision.**

Project Supervisor

Name: Prof. Sushil Shrestha

Academic Designation:

Project Coordinator

Name: Nabin Ghimire

Academic Designation:

ACKNOWLEDGEMENT

It is our privilege to express our sincerest regard to our project supervisor and coordinator, **Prof. Sushil Shrestha** for his appreciable guidance, encouragement, valuable inputs and productive criticism throughout the duration of the project.

We would like to thank **Department of Computer Science and Engineering (DoCSE)** and the whole university for providing us a chance to work on the project.

Abstract

The project “Smart Restaurant” is implemented to reduce the manual work and enhances the accuracy of work in a restaurant. This system helps the customers to reserve a seat, place an order, enroll their location, track their order and pay their bill through online payments by using a restaurant webpage. This software has been made in a user friendly interface. So that normal customers can perform all the tasks easily.

This project is also designed with full consideration to help the users in an easy manner without any unnecessary wastage of time. This system can be implemented in any type of restaurant. The menu card consists of various food varieties available in the restaurant which will be available in the web page. Through the menu, the customer can simply click and order the food. The messaging system tells the customers about the placement of the order. The billing system prepares the bill according to the delivered food and can be paid using online services. This system entirely is user friendly and saves lot of time.

List of Figures.

Figure No.	Figures.	Page No.
2.1	Rkeeper.	3
2.2	Toast.	3
3.1	Use Case Diagram.	4
3.2	ER Diagram.	5
3.2.1.1	Login Panel.	8
3.2.1.2	Food Menu.	9
3.2.1.3	Cart.	9
3.2.1.4	Admin Dashboard.	10
3.2.1.5	Database Login Admin Session.	10
3.2.1.6	Database Customer.	10
3.2.1.7	Database Customer and Order.	10
3.2.2.1	Flowchart(Customer View).	11
3.2.2..2	Flowchart(Admin View).	12
5.1	Gantt Chart.	14

Acronyms/Abbreviations

The list of all abbreviations used in the documentation is included in this section.

Short Form	Full Form
HTML	Hyper Text Markup Language
JS	JavaScript
PHP	Hypertext Preprocessor
CSS	Cascading Style Sheets
AJAX	Asynchronous JavaScript and XML
ERD	Entity Relationship Diagram
UI	User Interface

Table of Contents.

Acknowledgement.	i
Abstract.	ii
List of figures.	iii
Acronyms/Abbreviations.	iv
Chapter 1: Introduction.	1
1.1 Background.	1
1.2 Objectives.	1
1.3 Motivation and Significance.	2
Chapter 2: Related Works.	3
Chapter 3: Design and Implementation.	4
3.1 System Requirements and Specifications.	6
3.1.1 Software Specification.	6
3.1.1.1 Front End Tools.	6
3.1.1.2 Back End Tools.	7
3.1.2 Hardware Specification.	7
3.2 System Design.	8
3.2.1 Program Architecture.	8
3.2.2 System Architecture.	11
Chapter 4: Discussion on the Achievements.	13
4.1 Challenges Faced.	13
4.2 Features.	13
Chapter 5: Time Allocation.	14
Chapter 6: Conclusion.	15
References.	16
Appendices.	17

Chapter 1: INTRODUCTION

1.1 Background:

Smart Restaurant is a web application that enables customers and staffs to communicate with each other without any delay of time and makes the management task easier. Whenever we visit restaurant, the waiter brings that old menu with lots of item out of service. Even sometimes we have to wait the waiter to make our order. Same might be for the final bills. Recently, a few (very few) restaurants have begun offering a fully automated experience. It connect the customers and staffs virtually all the time through local Wi-Fi and any time the order can be placed. The bills of our food is updated whenever we placed order.

Most frequently adopted these days are self-order technologies, both at restaurants- provided kiosks and from guests' mobile devices but these services are not available at our country. Restaurants will be now using this services for the fulfilment and real-time inventory management. Implementing this service there will be less face-to-face guest interaction, but improved speed of service. This process is quite and user friendly. Now it's time to replace people by smart system in restaurants.

Smart Restaurant will provide a platform for the guests to reserve seat, to see the menu, to track the order, pay their bills. This will save a lot of time and working hand and will upgrade restaurant system into new level. The delayed process and human errors will now be removed. Some of the existing smart restaurants are Barrel Republic (California), Tokyo Smart Restaurant (Japan), and The Robot Restaurant (China).

1.2 Objectives:

1. To make live communication between customer and staff and save the time.
2. To deliver faster services.
3. To minimize the human labor.

1.3 Motivation and Significance:

In today's competitive hospitality industry, more restaurant owners are seeking a competitive edge than ever before. Restaurant operators are turning to software to streamline operations and deliver a better customer experience. The National Restaurant Association found that 73% of restaurant operators feel technology provides a competitive advantage and makes a restaurant more profitable but 47% feel they're lagging when it comes to having technology in their restaurants.

It's a common situation we everyone face in restaurant about order, bills and so on. Our world is being more technological. These days online order being delivered through drones in some places. In near future it might be more. We might have a robot as waiter in hotels, restaurant. These are the major things that inspire about this project. This tool supports wide future extension in terms of Artificial Intelligence.

The significance of our projects are:

1. User friendly interface.
2. Wide range scope.
3. Support future extension including online payment.

Chapter 2: Related Works

Nowadays the web based Restaurant system is getting popular with a tremendous speed. As the technology is spreading throughout the world, big Hotels and Restaurant are running their online service facilities at a very high rate.

The most Restaurant Industries are like to use a Restaurant Management Software point-of sale system, called a POS system, for any size operation. Most restaurant POS system are cloud based software programs that include all of your tableside ordering, kitchen, payment, business reporting tools into one system. Most popular POS software are Rkeeper, Toast.



Figure 2.1: Rkeeper

Rkeeper is used by the restaurant staff to place the order. Though this is smart ordering system, customer have to wait for their turn.

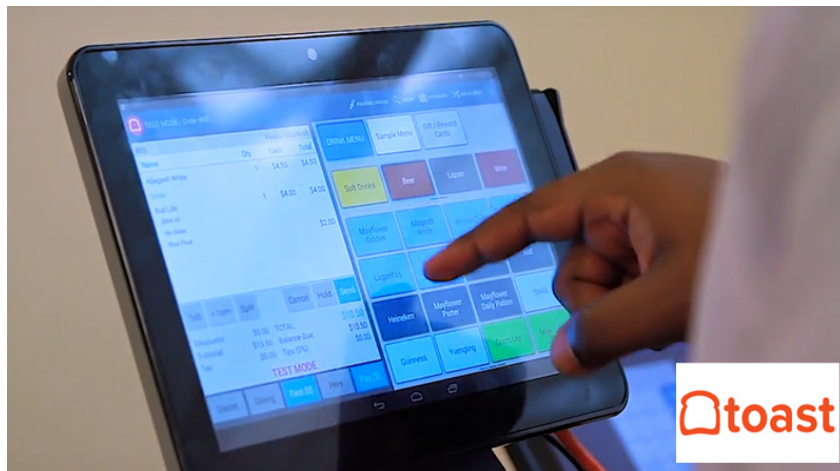


Figure 2.2: Toast

The working scope of Toast is quite similar to the Rkeeper.

Chapter 3: Design and Implementation

STEP 1: All possible requirements of the system to be developed are captured and documented in a requirement of specification of document.

STEP 2: The requirement specifications are studied. This system design and system requirements were analyzed.

STEP 3: The system's front end is developed first and then back end is developed, which are integrated in the next phase.

STEP 4: Each part is developed and tested for its functionality. Then these sections were combined and tested.

STEP 5: The product is developed in the customer environment.

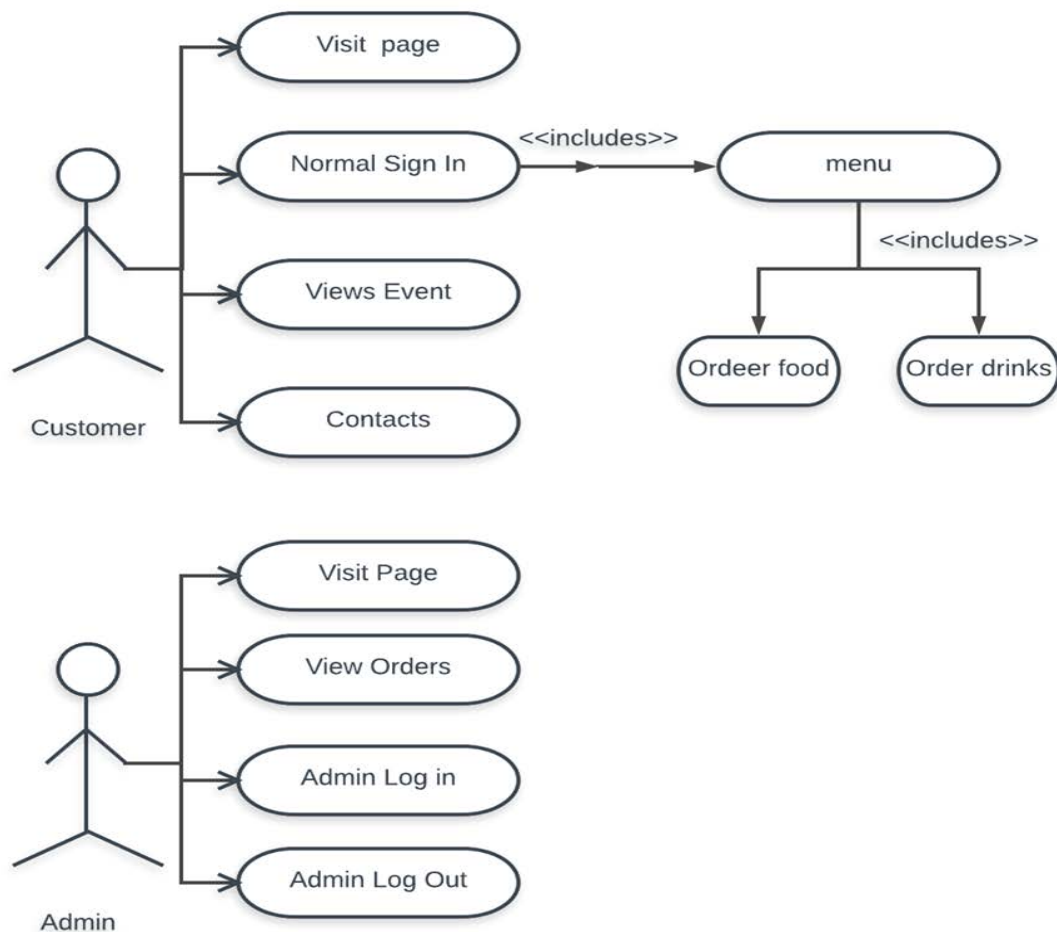


Figure 3.1: Use Case Diagram.

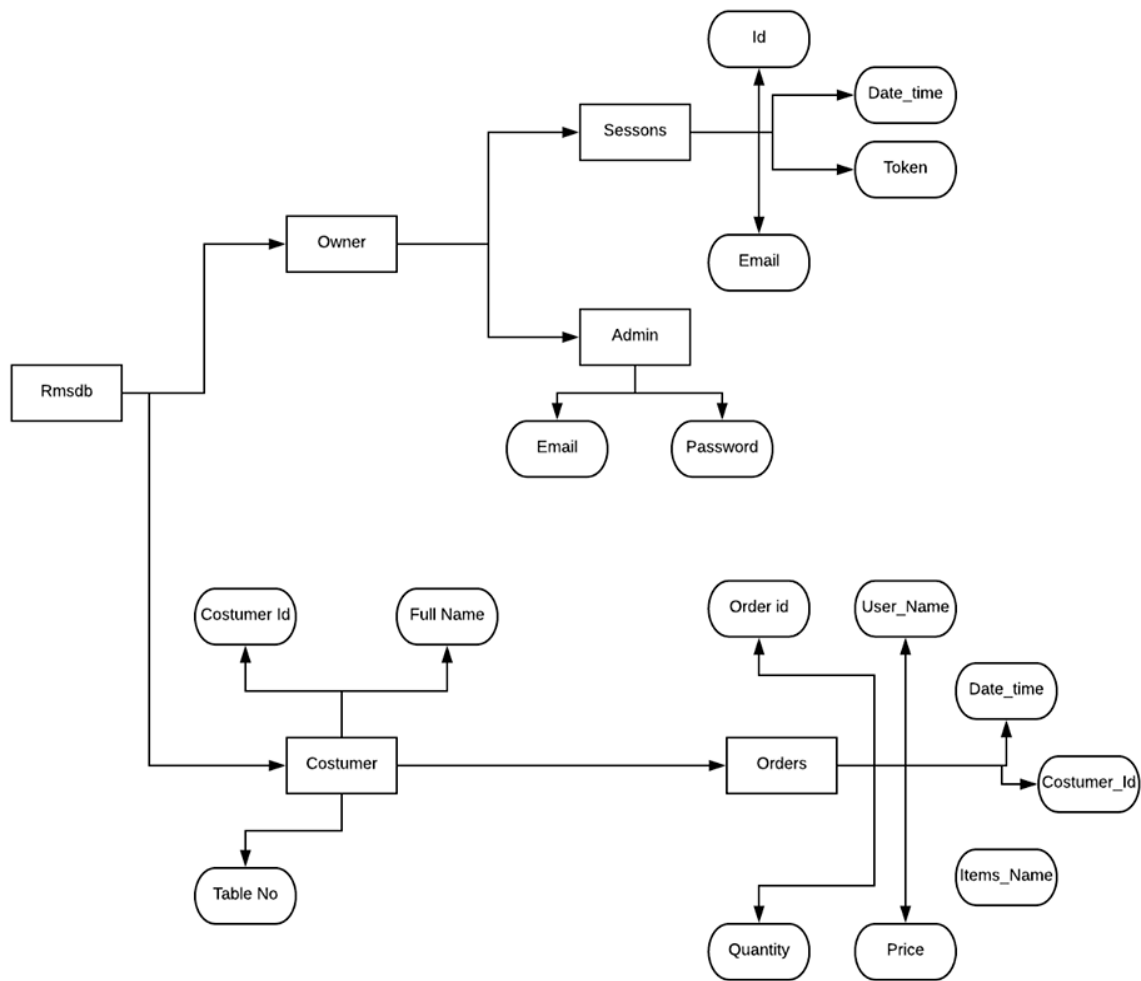


Figure 3.2: ER Diagram.

3.1 System Requirements and Specifications

3.1.1. Software Specification

3.1.1.1 Front End Tools.

3.1.1.1.1 HTML.

HTML is the standard markup language for creating web pages and web applications. With CSS and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. TML describes the structure of a web page semantically and originally included cues for the appearance of the document.

3.1.1.1.2 CSS.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

3.1.1.1.3 JavaScript.

JavaScript is a high-level, interpreted programming language that conforms to the ECMAScript specification. It is a language that is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles.

3.1.1.2 Back End Tools.

3.1.1.2.1 Node.JS.

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. JavaScript is used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

3.1.1.2.2 Express.js.

Express.js, or simply Express, is a web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

3.1.1.2.3 MySQL.

MySQL is an open source relational database management system. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress.

3.1.1.2.4 AJAX.

Ajax is a set of Web development techniques using many web technologies on the client side to create asynchronous Web applications. With Ajax, web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page.

3.1.2. Hardware Specification

The hardware requirements include a laptop or smartphone with the following things:

PC:

Operating system : Windows, Linux or MAC.

Processor : 2 GHz.

RAM : 1 GB minimum.

Screen Resolution : 1020*720 or larger.

CELL PHONE:

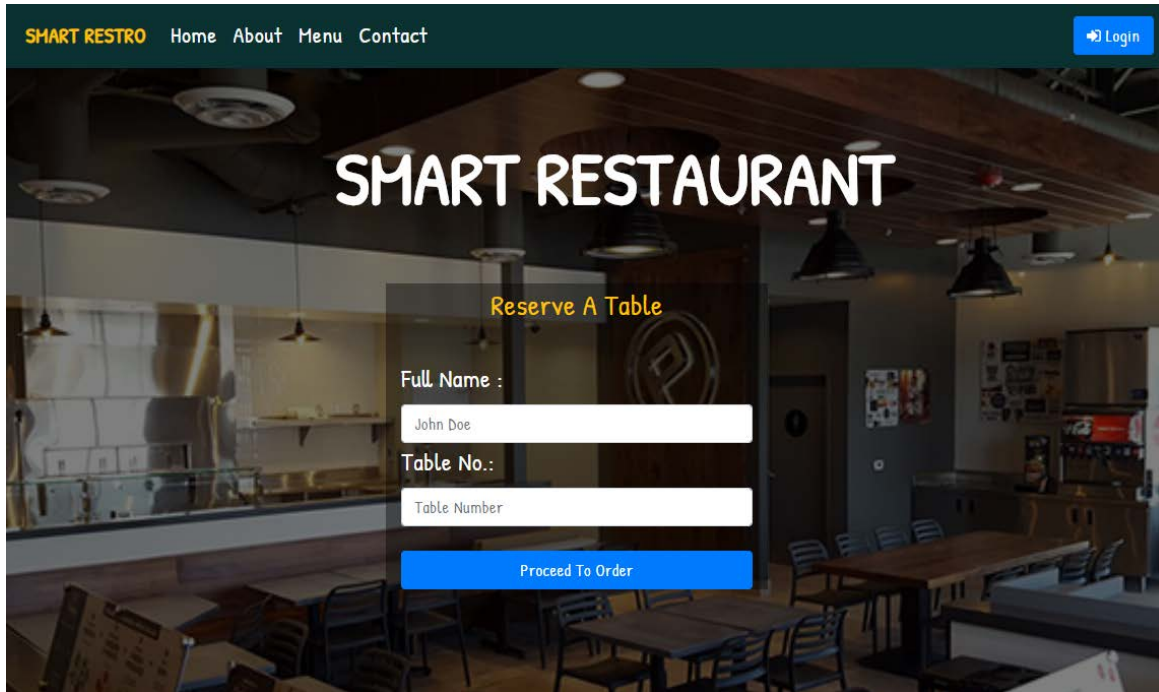
Operating system : Android, IOS.

Processor : 1.5 GHz.

RAM : 1 GB minimum.

3.2 System Design

3.2.1 Program Architecture.



The screenshot displays the SMART RESTAURANT website interface. At the top, a dark green navigation bar contains the text "SMART RESTRO" in yellow, followed by "Home About Menu Contact" in white. A blue "Login" button with a user icon is positioned in the top right corner. The main header area features the text "SMART RESTAURANT" in large, white, bold letters. Below this, a semi-transparent modal box titled "Reserve A Table" in yellow text is centered. The modal contains two input fields: "Full Name :" with the text "John Doe" and "Table No.:" with the text "Table Number". A blue "Proceed To Order" button is located at the bottom of the modal. The background of the website is a dimmed image of a restaurant interior with tables and chairs.

Figure 3.2.1.1: Login Panel

DRINKS

Coca Cola



50

ADD TO CART

Pepsi



50

ADD TO CART



Coffee



Lasii



Figure 3.2.1.2: Food Menu

CART			
ITEM	RATE	QUANTITY	
 Coffee	150	<input type="text" value="2"/>	REMOVE
 Lasii	150	<input type="text" value="1"/>	REMOVE
			Total RS:450

[ORDER](#)

Smart Restaurant



Figure 3.2.1.3: Cart

Smart Restro

Dashboard

Home

Post An Item

View Orders

Logout

Orders:

Order ID	Customer ID	Full Name	Table No	Item Ordered	Quantity	Price	Time
25	131	Ashish adhikari	1	Momo	1	200	Wed Jan 30 2019 17:47:06 GMT+0545 (Nepal Time)
26	131	Ashish adhikari	1	Chowmin	1	200	Wed Jan 30 2019 17:47:06 GMT+0545 (Nepal Time)
27	131	Ashish adhikari	1	Burgar	1	300	Wed Jan 30 2019 17:47:06 GMT+0545 (Nepal Time)
28	132	Gaurav Thaguna	9	Momo	4	800	Wed Jan 30 2019 17:50:41 GMT+0545 (Nepal Time)
29	132	Gaurav Thaguna	9	Chowmin	1	200	Wed Jan 30 2019 17:50:41 GMT+0545 (Nepal Time)
30	134	Ashish adhikari	5	Momo	3	600	Thu Jan 31 2019 10:27:15 GMT+0545 (Nepal Time)
31	134	Ashish adhikari	5	Chowmin	1	200	Thu Jan 31 2019 10:27:15 GMT+0545 (Nepal Time)

Figure 3.2.1.4: Admin Page.

+ Options									
← T →									
<input type="checkbox"/>	Edit	Copy	Delete	28	smartrestro321@gmail.com	ad07e0df4f93ee0fd861d0ef748e90b1	0000-00-00 00:00:00		
<input type="checkbox"/>	Edit	Copy	Delete	29	smartrestro321@gmail.com	fd4f339399ecc765bd03baba812bc3b4	0000-00-00 00:00:00		

Figure 3.2.1.4: Database Admin Login Session.

+ Options									
← T →									
<input type="checkbox"/>	Edit	Copy	Delete	125	Shusant Sapkota	3	2019-01-25 18:36:29		
↑	<input type="checkbox"/>	Check all	With selected:	Edit	Copy	Delete	Export		

Figure 3.2.1.4: Database Customer.

+ Options									
← T →									
<input type="checkbox"/>	Edit	Copy	Delete	11	Shusant Sapkota	Momo	200	125	2
↑	<input type="checkbox"/>	Check all	With selected:	Edit	Copy	Delete	Export		

Figure 3.2.1.4: Database Customer and Order.

3.2.2 System Architecture.

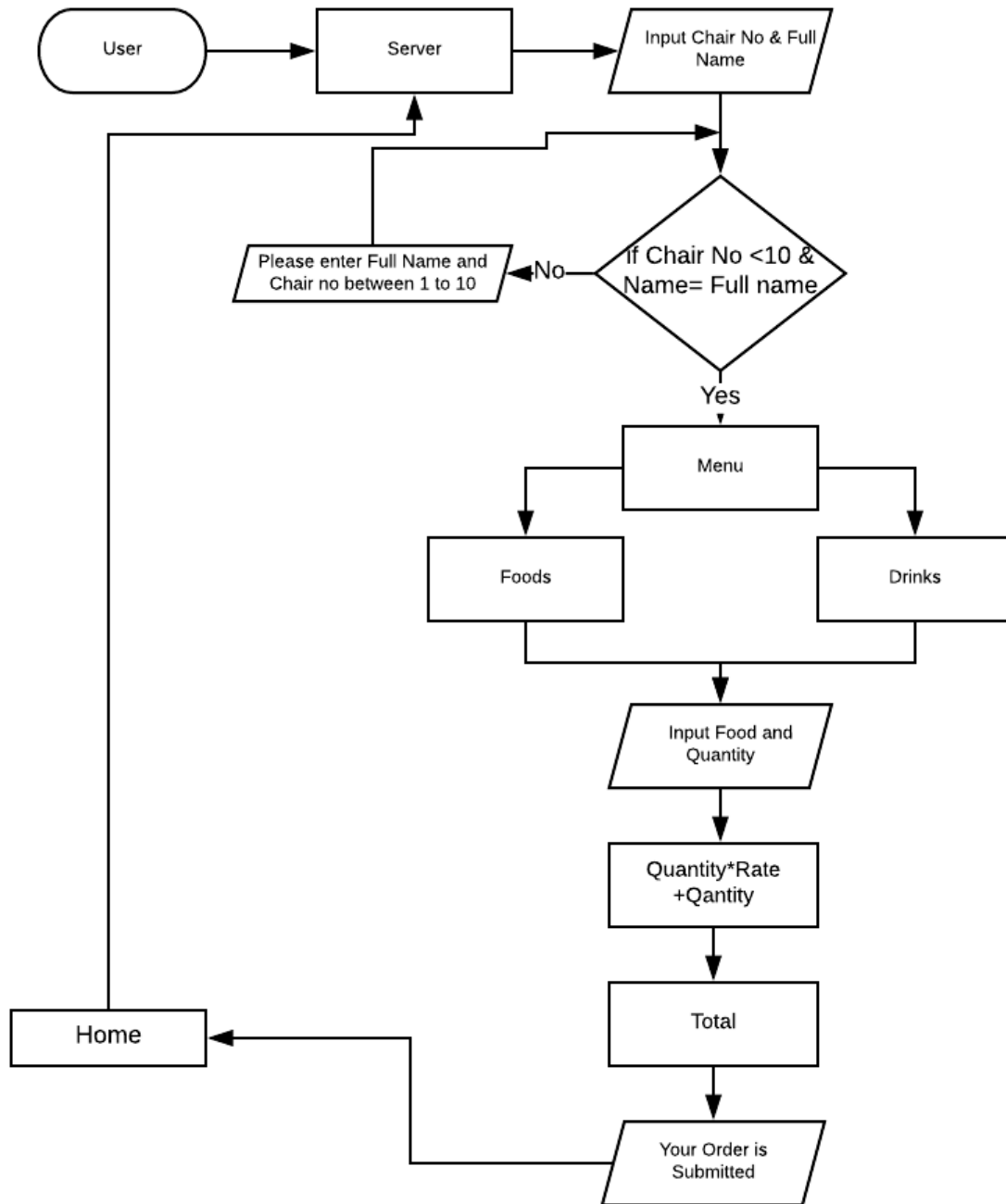


Figure 3.2.2.1: Flowchart (Customer View)

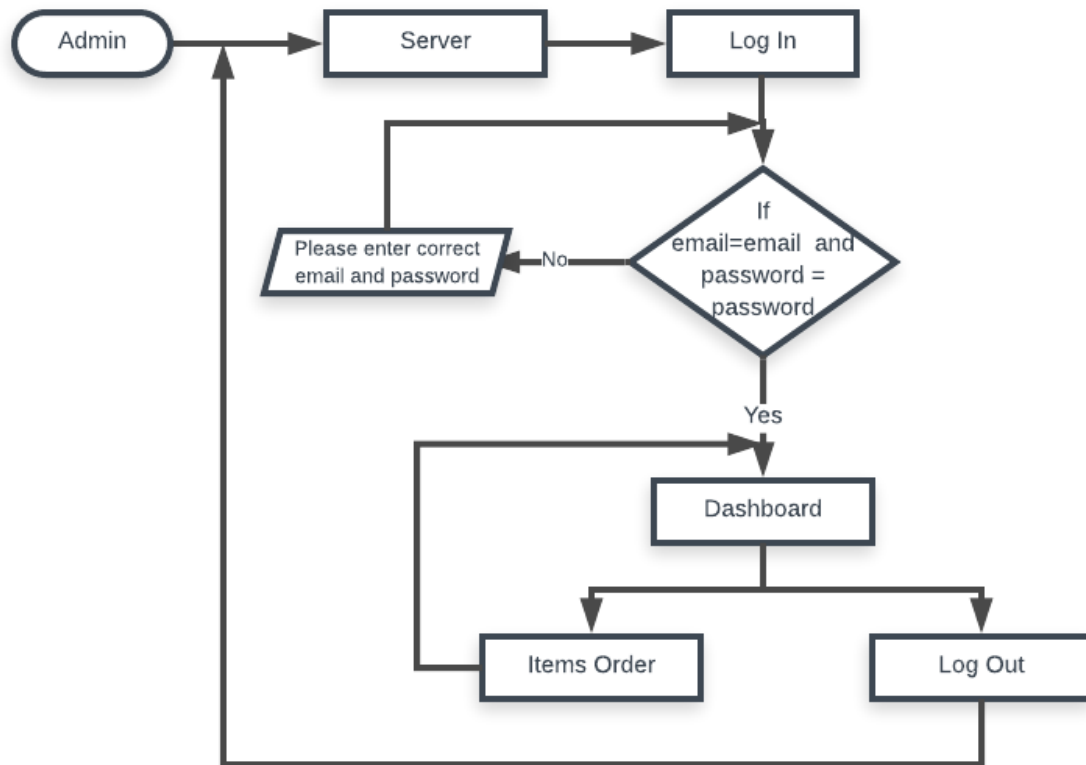


Figure 3.2.2.1: Flowchart (Admin View)

Chapter 4: Discussion on the Achievements.

4.1 Challenges Faced.

There were several problems which were hard to tackle, some of them are:

- This project was a web based application so we had to code differently than console programming.
- It takes to much time for design homepage as we had design two home page one for customer one for staff.
- We spent a lot of time for learning back end programming language like MYSQL, ExpressJS, Node.Js etc. So time management was challenging for us.

4.2 Features.

The following are the features of “SMART RESTAURENT”:

1. It make live communication between customers and staffs.
3. It helps for providing faster services and saves time.
4. It offers interactive interface.

Chapter 5: Time Allocation.

The following is the time allocation Gantt chart in the process of completion of our project as schedule from initial to final phase.

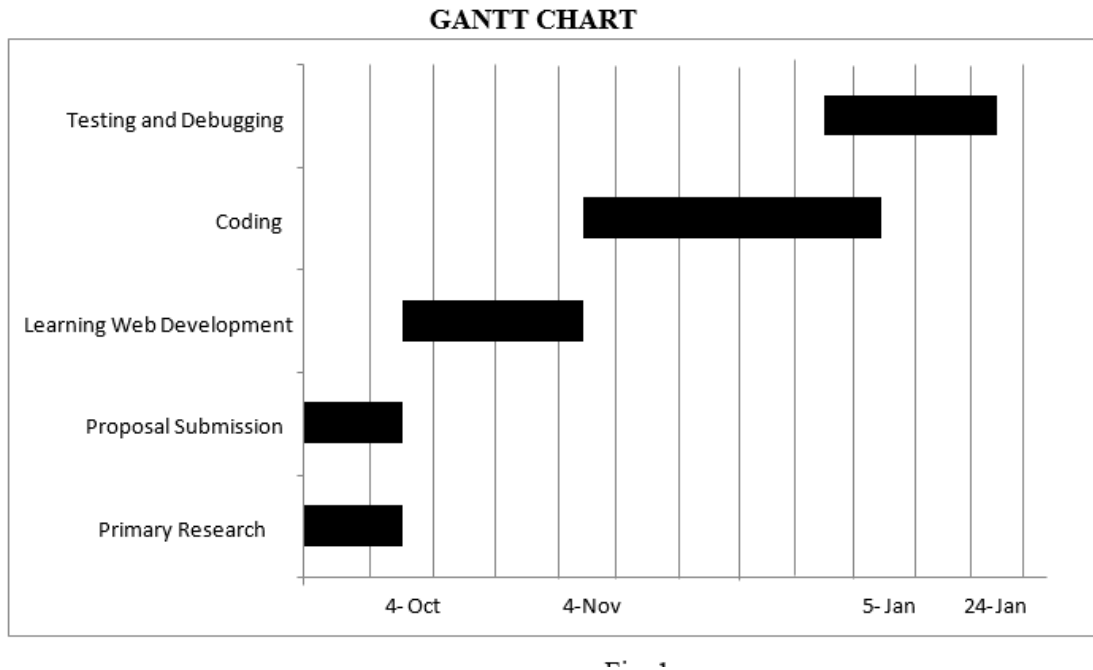


Figure 5.1: Gantt Chart

Chapter 6: Conclusion.

The smart restaurant is a web based interface that allows the customers in a restaurant to make live contact with the staff and place their orders.

In general, whenever we go to any restaurant we have to wait for waiter to place our orders. It takes more time respective from both sides. So, this web interface allow user to place their orders from their respective smart phone or laptop.

Though the project tried to includes maximum scope it doesn't include online payment system.

It supports wide future extension including live online payment.

References.

1. Ninja, T. N. (2016, May 25). Node JS Tutorial for Beginners #1 - Introduction. Retrieved from <https://www.youtube.com/watch?v=w-7RQ46RgxU&list=PL4cUxeGkcC9gcy9lrMJ75z9maRw4byYp>
2. Flanagan, D. (2011). JavaScript: The definitive guide. Sebastopol, CA: O'Reilly.
3. Tahaghoghi, S. M., & Williams, H. E. (2007). Learning MySQL. Sebastopol, CA: O'Reilly.
4. ProgrammingKnowledge. (2012, December 25). Beginners MYSQL Database Tutorial 1 # Download , Install MYSQL and first SQL query. Retrieved from <https://www.youtube.com/watch?v=iP1wOSsKjW8&list=PLS1QulWo1RIahIYDqHWZb81qsKgEvPiHn>

Appendices.

Save Selected Orders to Database

```
exports.getMyItems = function (req, res) {
  var id = req.params.id;
  pool.query("SELECT items.id,items.name,items.category,items.rate FROM items", [id],
function (err, data) {
  if (err) throw err;

  res.render("dashboard/myItems", {
    items: data
  });
});
}
```

Authorized Admin Login.

```
exports.loginAdmin = function(req, res){
  var email = req.body.email;
  var password1 = req.body.password1;

  // password1 = md5(password1 + email + "d800a6eca2c8d8e4f63d577a6b483a62");

  pool.query("SELECT COUNT(*) as num FROM admin WHERE email=? AND
password1=?", [email,password1], function(err, data){
    if (err) throw(err);

    if (data[0].num > 0){
      let age = (30 * 24 * 86400 * 1000);
      let token = md5(randomString());

      res.cookie('email', email, {
        maxAge: age
      });
      res.cookie('token', token, {
        maxAge: age
      });
      pool.query("INSERT INTO sessions (email, token, date_time) VALUES (?,?/?)",
[email, token, date_time], function(err, data){
        if (err){
          res.status(400).send("error")
        };
        res.render("dashboard/view", {
          sessions: data
        });
      });
    }
  });
}
```



```
        });  
    });  
    } else {  
        sendError(res, 'Email and password did not match.');
```

```
    }  
    });  
}
```