

Restaurant Food Delivery and Pickup System

Final Report

SUBMITTED TO: DR. HAROLD NELSON

SUBMITTED BY: SAPANA POUDEL, GANDEEV NIRAULA, ABDULLAH GHURAB

| CSC 480 | 05/05/2019

Table of Contents:

***1. Abstract***

***2. Introduction***

***Roles of the Group Member***

***3. Cost of Third-Party Vendor for Online Ordering System***

***Third Party Vendor Vs. Restaurant POS System***

***4. Design Methodology and Design***

***1. Requirement Determination***

***i)Functional Requirement***

***ii)Non-Functional Requirements***

***2. Architecture***

***3. Advantage of 3-Tire Architecture***

***4. Physical System***

***5. Use Case Diagram***

***5. Implementation***

***i. Software Design***

***ii. Tools and Technology***

***iii. Programming language***

***6. User Interface***

***7. Testing***

***i. Software Testing Types***

***ii. Testing Approaches***

***iii. Testing Levels***

***8. Completion Process***

***9. Experimental Result and Analysis***

***10. Conclusion and Future Work***

***11. Reference***

**1. Abstract**

People today visit restaurant at least twice a week because some loves to eat outside while some doesn’t have time to cook, some do dine in while some do carry-out or even get it delivered to their home or offices. That is why, there are lots of restaurants around offices and market places. Big restaurants come up with different schemes to attract customers while the small and single restaurant owner struggle getting customers. So, our main focus was to develop an affordable small budget online ordering system for low budget food business. It can be gamechanger for small quick serve restaurants and food trucks. In small business, like food truck, manual ordering can be resource and time consuming during busy hours and are more prone to errors. Our main goal with this system is to boost the average order sizes and operational efficiency. Also, they require data analytics and statistics to create better strategies and grow their sales, which needs real time customizations. Restaurant single point of sale system is a better as well as long term solution while solving these issues. Developing an effective, affordable, and easily maintainable system for our client who is the food truck owner is our main objective in this project.

**2. Introduction**

Restaurant Food Delivery and Pickup System (RFDPS) is an online food ordering for individual restaurant, which allow users to get their food deliver to their door or takeout from a specific restaurant through the restaurant app. A customer can view the menu, search for the menu, view pricing information, and select the menu. Also, customers have an option to pay online or at the restaurant. Customer can sign-up and login the app and get the benefit of reward system. It allows customer to get it delivered up to 10-mile radius from the restaurant. Online ordering is a game changer for restaurant, which can attract many customers by providing convenient and faster service. With the technology evolving in every sector, online food ordering system has become popular in the present restaurant industry to compete within the market and to serve customers in a better way. And we believe this design will serve the purpose.

This system will work not only for small business but for any size, because it is a scalable application. It can work on either physical or cloud premises as needed. For small businesses, physical hardware and software are recommended, while for big restaurants, cloud could be a better solution, when it comes to deploy this application. It helps small business with reducing payroll expenses, encouraging repeat restaurant customers, increasing profit margin, and saving time for both employees and customers while serving large volume of customers in a specific time.

**Tentative Schedule**: Refer to Excel Spreadsheet attached with the report

**Roles of the Group Members:**

**Sapana Poudel:** My role in this project is system design, building, and testing. In first two weeks, I gathered requirement. I used interviews, questionnaires, and observations techniques to gather the requirements of fully functional web application. I built two web pages; signup and login pages using Microsoft Visual Studio. I have also built a Home page, Sign up page, and Login Page. I have created CSS files for all the page I have created and finished the design. I have created JavaScript files to validate the forms and check the accuracy of the information that user input and connect them with .aspx files. I have also created C# files for the pages above for server-side development and successfully connected the front side HTML content with the database side. So far, user and admin can successfully sign up and log in to the website. I also worked on a part of master page content to create consistency throughout the web pages. I have also worked on menu page, add to cart page, and payment page, their CSS files, JavaScript files and C# files for server side. In order to keep consistency throughout the web pages, master page is created, which displays similar layout in all web pages. I have created database and successfully connected signup page with the database. I also successfully logged in to the page as a user. I have created admin panel and successfully logged in as an administrator. As an administrator, I am able to perform following functions: change menu, change price information, add menu, view order, and view user’s information.

**Gandeev Niraula:** My role in this project is system design, coding various components of the project, database design and implementation. I was involved in coding of Master page which in .Net provides consistent site wide layout throughout all other web pages. I created all the necessary tables in the database required for proper functioning of this web application. The tables created for this project were i) signup ii) order iii) contact us iv) add category v) add cart. I also created multiple stored procedures for our program to perform the repetitive tasks that requires checking, looping, multiple statements and no user interaction. Recently, I have been working on the admin panel of the project. I have been able to create adminpanel.aspx page so far where admin will have the following special privileges i) Add/Update/Delete Category ii) Add products iii) Update products iv) View orders v) View messages vi) List of members vii) Search member. We are still working on List of member and Search member feature of it and plan to complete those in future. I was also involved in testing such as Usability and Functionality testing. In usability testing, we dealt with how users are able to navigate around our site and get from one area to another while functionality testing consisted of testing for data validation, user input testing, URL validation and dynamic content testing.

**Abdullah Ghurab:** My role in this project is the Database section and the implementation of it, help writing the code for different parts and functions of the project. In addition, I was the last team member to be added to this project, so the touch-ups were my thing. Moreover, I have been editing and assisting all along the project. Since I have been going around most of the stuff on the project, I have been able to write the final report, and it was a part of my job to compose the final presentation.

**3. Cost of Third-Party Vendor for Online Ordering System**

According to expert, digital ordering from computers, tablets, and mobile phones is growing 300 percent faster than dine-in traffic. There are lots of online ordering systems that are available in market, but cost of these third-party vendors has to be examined carefully before we decide to go with any of them. Because these vendors charge lots of hidden commission and fees to the restaurants and these commissions and fees mean that the more business a restaurant does online, the higher percentage of its sales go to the vendor. These are some of the popular vendors and their cost:

* GrubHub- It charges an average of 13.5% commission on every order. Lower rates are available but that means lower placement in search results for the restaurant
* UberEats- it charges restaurants 30% for each order with an additional customer side charge
* Eat24- it charges a fixed percent which results in 12.5% of a restaurant’s net online sales
* Door Dash-It charges restaurant 10-25% depending on the restaurant and the market they are operating on

Third Party Vendor Vs. Restaurant Ordering Software System:

Third Party Vendor:

* Online ordering with third-party vendor is an expensive venture with all the commissions and hidden fees
* Data drives business but most vendors don’t provide their restaurant clients access to the customer data they collect as consumers place orders
* Third party vendor attracts new customer but restaurant still has to pay commissions for a guest’s second and their order and so on
* With them, the online ordering menu is hosted separately from the rest of the restaurant assets, and it is difficult and expensive to set up and make changes to menu items
* Restaurant operators are essentially giving up control over their brand as restaurant clients has very little control over how they get represented on vendor’s site

Restaurant Owned POS System:

* Owing own’s system means there is no need to pay commissions and restaurant can keep more revenue from each online sale rather than giving away to third-party
* Restaurant has access to customer’s contact information and detailed order, which help them to better understand the market
* When customers order through restaurant’s site directly, it spared the commission fees
* Online menu changes are as simple to make as they are in store
* A restaurant’s System is built on the quality of the food, service, and overall guest experience

From this study, we can conclude that restaurant owned point of sale system is a better solution than third-party owned online ordering system in terms of its budget and effectiveness.

**4. Design Methodology and Design**

1. Requirements Determination

i) Functional Requirements

• The system must display the menu for the customers.

• The system must provide price information in the menu.

• The system must allow customer to sign up or login the app.

• The system must allow customer to add or remove items from the cart.

• The system must allow proceed to check out and see total bill.

• The system must provide option of pay online or at the store.

• The system must provide option of pickup or delivery for a user’s order.

• The system must display order to the employee.

• The system must manage payment system

• The system must show the map of the store location.

• Using RDPS, the customer must be able to call the store.

ii) Non-Functional Requirements

• The system can run on handheld devices

• The system should be able to run on any Web browser.

• The system should be available for use 24 hours a day.

• The system should include all available safeguards from viruses, worms, Trojan

Horses, etc.

**2. Architecture**

Three tire architecture which is client server architecture is used for this project. Client system handles presentation layer, application server handles and application layers while database server handles database layer. The communication takes place between the client and the server. Client system send requests to the server and server system processes the request and sends back the data to the client system. Here is the more breakdown of what each layer does in the system:

* Presentation Layer: the top-most level of application is the user-interface. The main function of the interface is to translate tasks and results by sending contents to browsers in the form of HTML/CSS/JS
* Logic Layer: this layer coordinates the application, processes commands, make logical decisions, and perform evaluation and calculation. It also processes data between the presentation layer and data layer. It is written in C#.
* Data Layer: Here information is stored and retrieved from database. The data is passed back to the logic layer for processing and back to the user. MSSQL is used in integrated environment in visual studio.

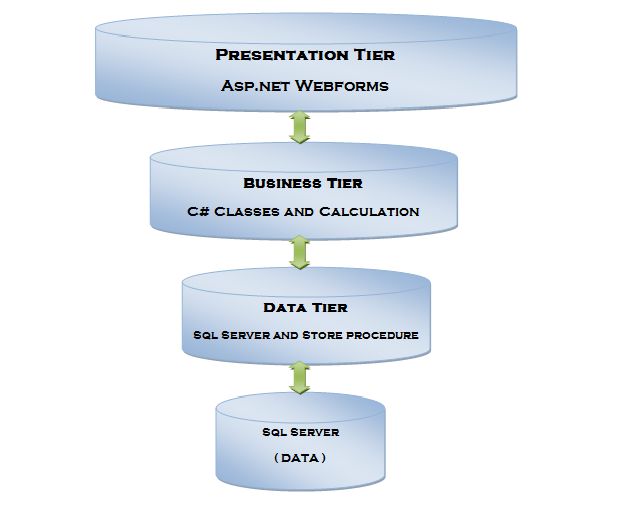


Figure 1: 3-TIre Architecture

**3. Advantage of 3-Tire Architecture**

* It gives the ability to update the technology stack of one tire, without impacting other areas of the application
* It will make it easy to scale the application
* It adds reliability and more independence of the underlying servers and services
* As an owner, you have the ability to utilize new technologies as they become available and provides opportunity to redesign application and actually looks the need of future

**4. Physical System**

Hardware/Software Interface: This section lists the minimum hardware and software requirements needed to run the system efficiently.

Technology and Framework: Visual Studio 2017 and .Net Framework

Front End: .ASPX (HTML, CSS, JavaScript)

Back End Server: C#

Database: SQL Server Integration Studio

Host: Windows Server (Minimum 2GB RAM)

Client: Any machine that can access a web page

Development: Windows Machine with IIS server installed (Minimum 1GB RAM)

**5. Use Case Diagram**

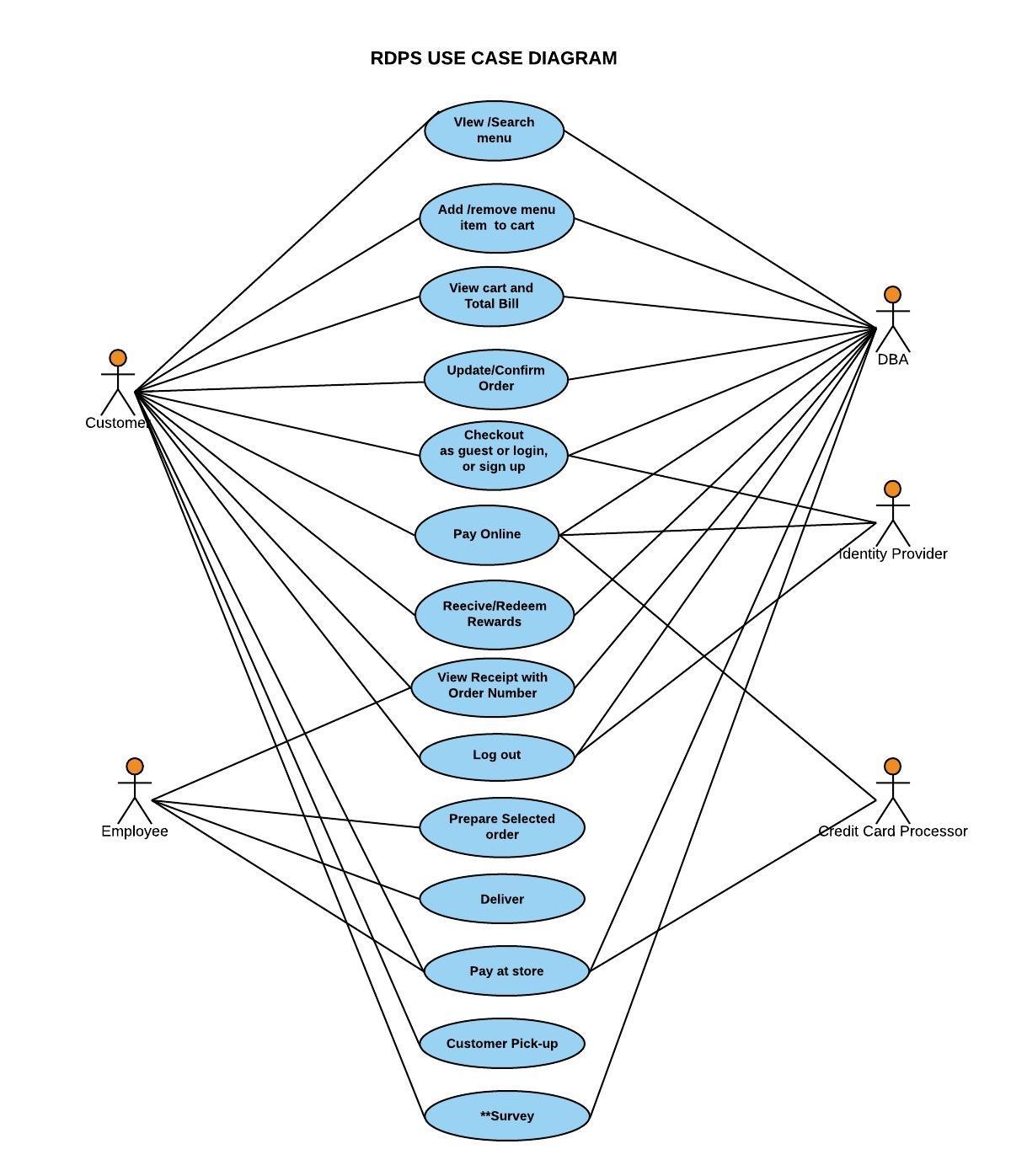


Figure 2: Use Case Diagram for Online Ordering Syste

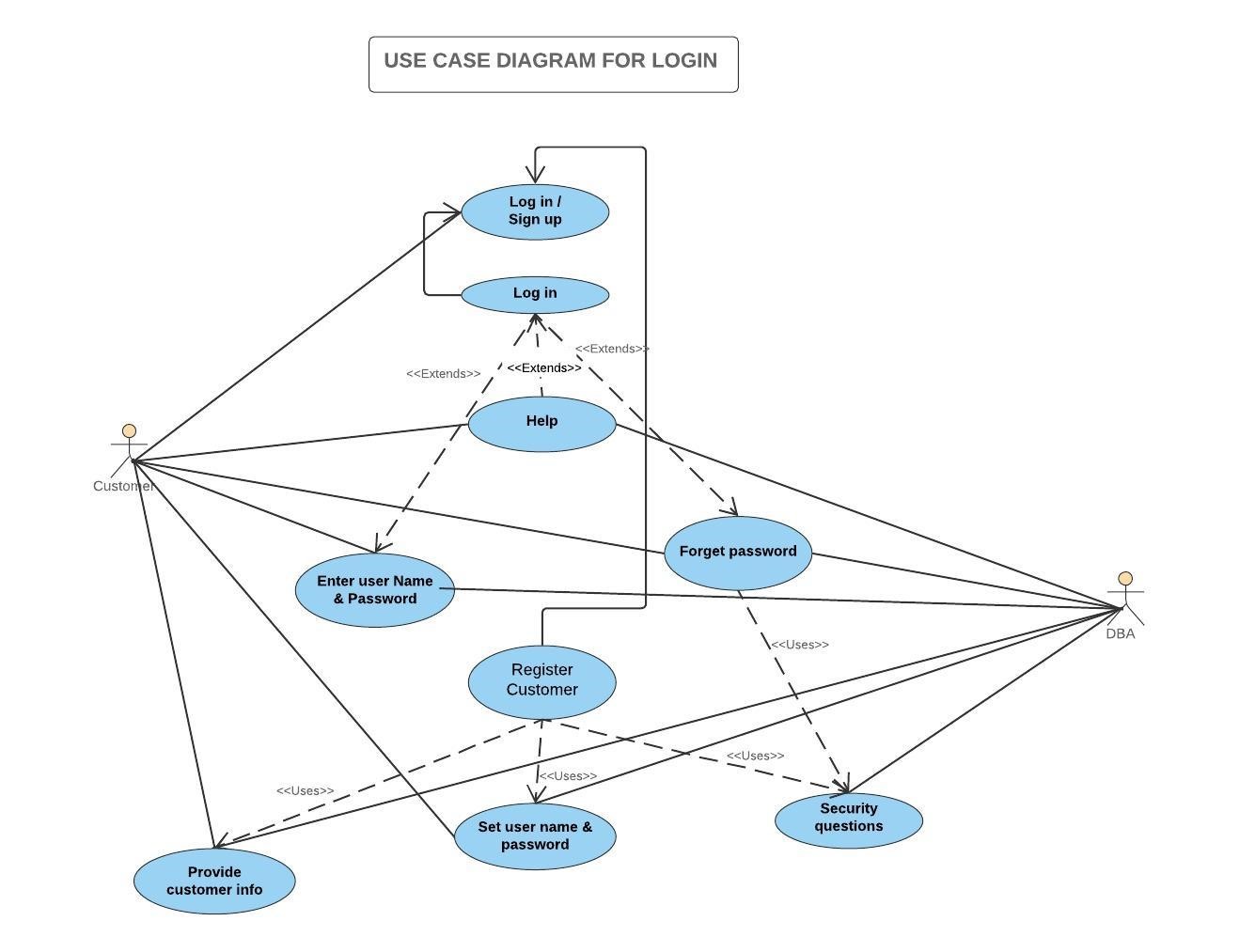


Figure 3: Use Case Diagram for Checkout:

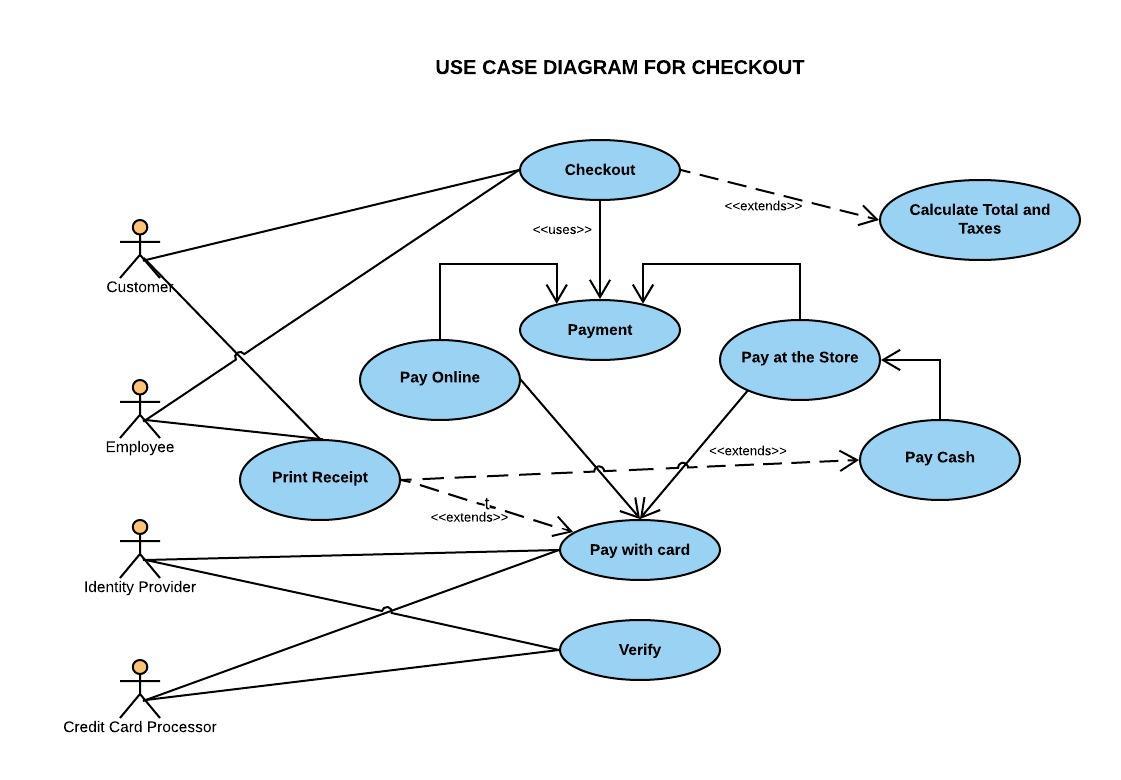
****

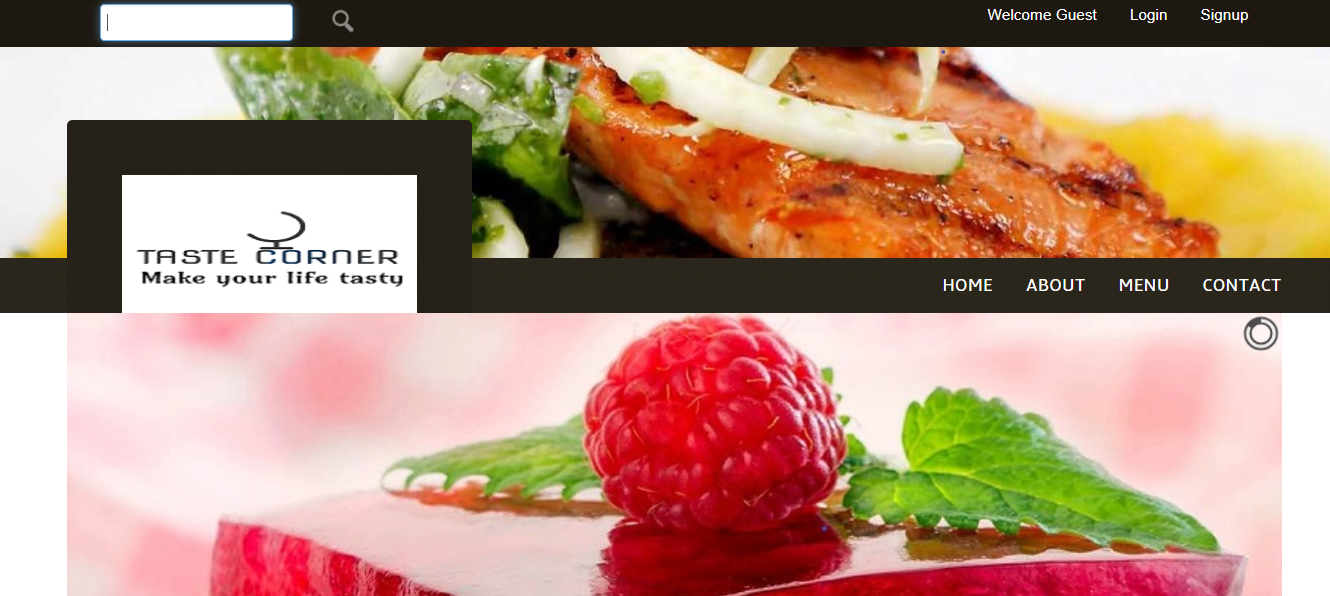
Figure 4: Use Case Diagram for Checkout

**5. Implementation**

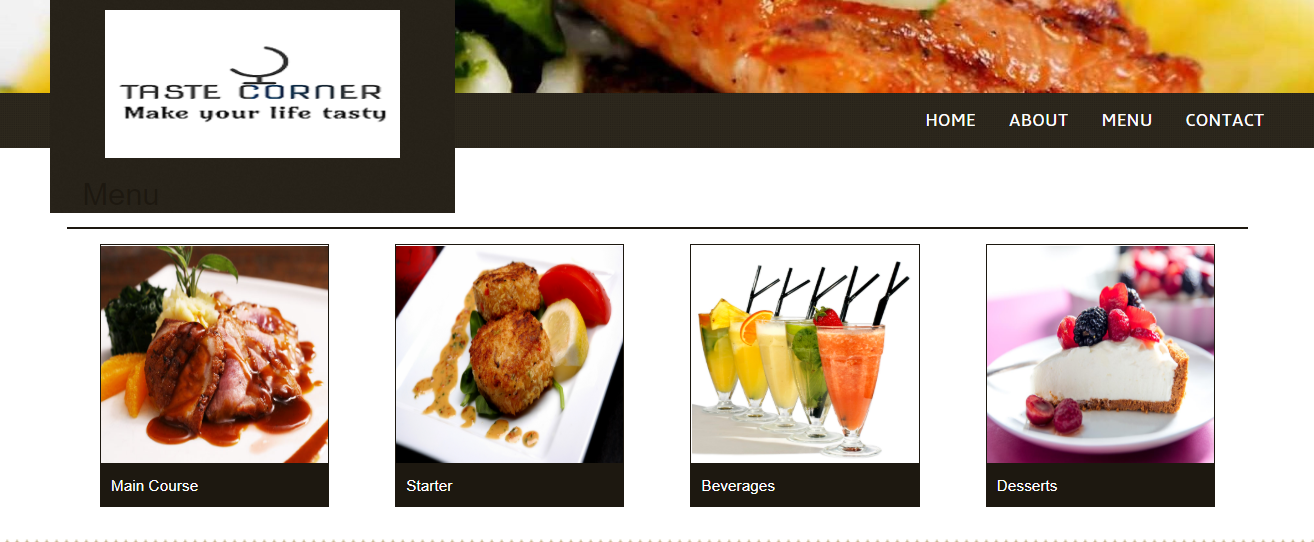
1. Software Design: We used waterfall model for software development methodology for our system design where we laid out detail planning and have a complete forecast of the exact task and features to be delivered in schedule time.
2. Tools: Visual Studio 2017, .Net Framework, IIS Server in Windows, Integrated SQL Server Database tool in Visual Studio
3. Programming Language: HTML, CSS, JavaScript, C#

**6. User Interface**

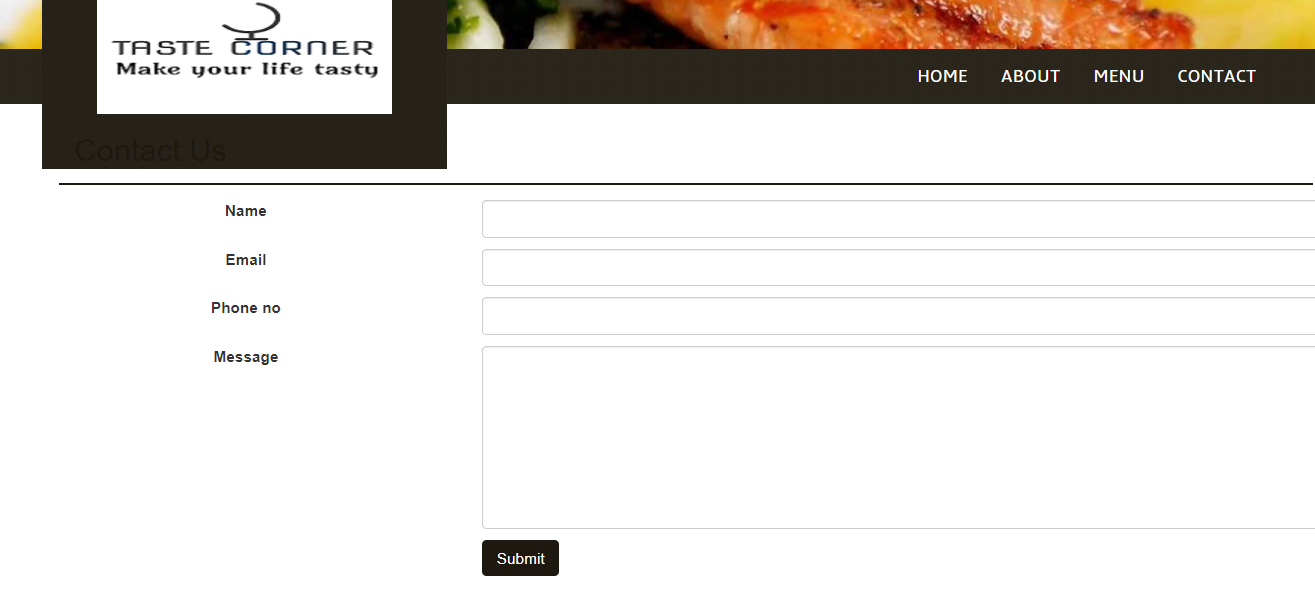
1. **Home Page**

****

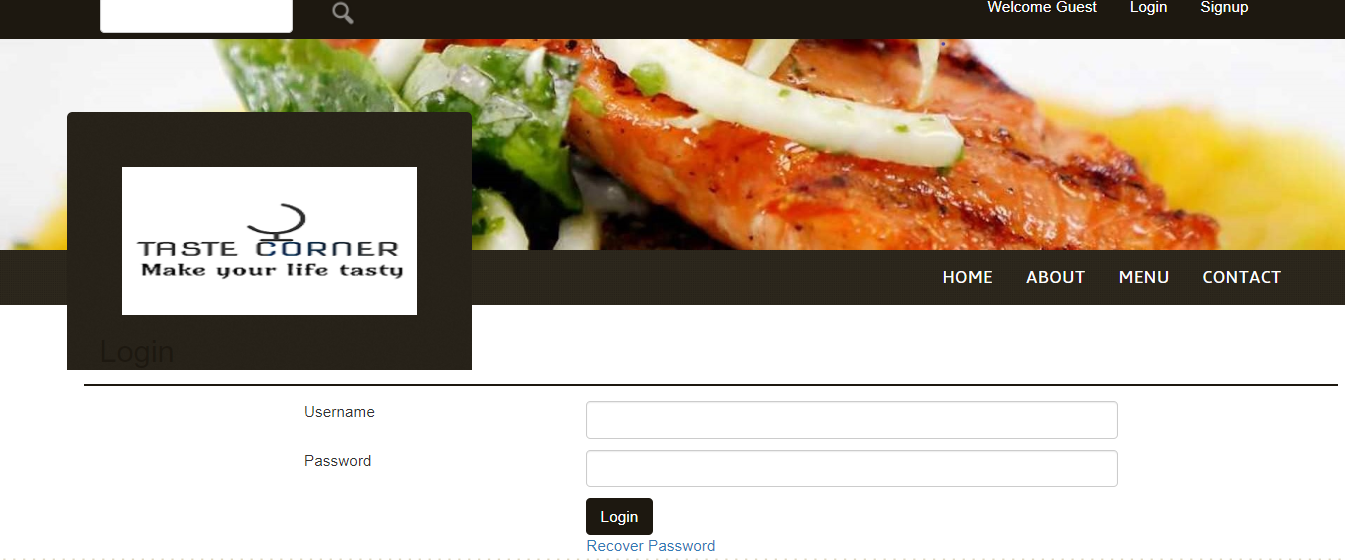
1. About Page
2. Menu Page



1. Contact Page



1. Login Page



1. Signup Page
2. Welcome Page

**7. Database**

We have created 7 tables in the database portion of our project. The tables that we have created are as follows: i) addcart ii) addcart1 iii) addcat (for category) iv) addproduct v) contact Us vi) order table vii) signup. The signup table stores the data that customer uses in order to sign up to the webpage. The order table store details of all the orders made by every customer. Contact us table stores all the messages sent by the customers. Add product and add category table could only be controlled by the admin to add/update product or add/update menu categories. Add cart table shows all the list of items added to the cart.

**8. Testing**

i. Software Testing Types:

1. Manual Testing: Manual testing is the process of testing the software manually to find the defects. Tester should have the perspective of end users and to ensure all the features are working as mentioned in the requirement document. We have tested our application and run all the pages. We have login and signup, connected with the database and login as an administrator to the admin page.
2. Automation Testing: Automation testing is the process of testing the software using an automation tool to find the defects. In this process, testers execute the test scripts and generate the results automatically by using automation tools. Some of the famous automation testing tools for functional testing are QTP/UFT and Selenium.
3. Testing Approaches
4. White box testing: white Box Testing is based on applications internal code structure. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. This testing is usually done at the unit level.
5. Black Box testing: black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure.
6. Gray box texting: Grey box is the combination of both White Box and Black Box Testing. The tester who works on this type of testing needs to have access to design documents. This helps to create better test cases in this process.
7. Testing Levels
8. Unit testing
9. Integration testing
10. System testing
11. Acceptance testing

**8. Completion Process**

We are still working on various functionalities for the admin panel. Also, once we achieve all the functional and non-functional requirements, we need to do Integration Testing on Main Page, configure QA environment and Quality Assurance Test before the deployment.

**9. Experimental Result and Analysis**

**-** We plan to check for OWASP top 10 security vulnerabilities in our web application by using penetration testing tool like Skip fish in Kali Linux. Based on the report generated by the tool, we will implement necessary step to mitigate the threat. The other testing that we need to perform on our web application are as follows:

I) Functional Testing: This testing is used for checking all the links of the web pages, form testing, cookie testing and database connection.

II) Usability Testing: This testing checks the navigation and user friendliness of the webpages.

III) Interface Testing: This check if web server and application server interface, application server and database sever interface have proper connection or not. It ensures that users do not see any error messages.

IV) Compatibility Testing: This testing is very important as it checks browser compatibility, OS compatibility, mobile browsing and printing options.

V) Performance Testing: It includes web load and web stress testing.  Web load testing technique checks if many users can access the same page at the same time and whether a web page can handle heavy load on any specific page. Web stress testing is done on the site to see that how will the site react and recover during the stress time.

**10. Conclusion and Future Work**

The main objective of this project was to find the simple and scalable solution for small restaurant. Small food service startups like a takeout restaurant and food trucks operate on a low budget and receive a high volume of order during the peak hours which could be either breakfast, lunch or dinner depending on their business hours and kind of food they serve. Therefore, keeping these small businesses in mind; our project is a .NET based monolithic web application which allows their potential customers to place an order online, make a payment through their card and receive an estimated time for their order to be ready. This application will aid small food serving businesses. like takeout and delivery restaurant and food trucks to increase their sales by receiving order in real time and saves the hassle for customers by facilitating order placement without making a phone call and saves time by obliterating the need to visit the brick and mortar site itself to place an order.

We have able to describe the concept of software engineering, apply various UML diagrams to develop abstract model of a system, implement a project plan, most importantly develop a spirit of working in a team and by doing this project we were able to meet the goal of this course (CSC 480).

Since this was a big system, we were not able to complete everything. We still have few functionalities we have to complete, which are:

• System shows the map of the store location and provides contact information.

• System secure the personal information of the customer compliance with the Data Protection Act.

• Testing system.

• Deploying system

**References**

**1. ProgrammingKnowledge. “ASP.NET Tutorial 4- How to Create a Login Website - Connecting Database to Registration Page.” *YouTube*, YouTube, 14 Dec. 2013, www.youtube.com/watch?v=RCnRmoa8r0A&index=4&list=PLS1QulWo1RIaM8 -S7kTHgWd\_pGNu-CyQS.**

**2. ProgrammingKnowledge. “ASP.NET Tutorial 1- Introduction and Creating Your First ASP.NET Web Site.” *YouTube*, YouTube, 8 Dec. 2013, www.youtube.com/watch?v=KVlXccl-XBA.**

****tutorialspoint.com. “ASP.NET Tutorial.” *Www.tutorialspoint.com*, Tutorials Point, www.tutorialspoint.com/asp.net/.