**PGBooking: An Online Booking System for Hotels and Resorts**

**In Puerto Galera**

An

Application Development Project

Presented to the Faculty of

**Mindoro State University Calapan Campus**

Masipit, Calapan City

Oriental Mindoro

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Information Technology

by:

**De Guzman, Yuan Gabriel T.**

**Logatoc, Kate Angela**

**Medrano, Dominic E.**

**Rivera, Katherine B.**

**BSIT III-F1**

December 15, 2023

**CHAPTER I**

**INTRODUCTION**

**Project Context**

In the dynamic landscape of travel and tourism, the demand for efficient and user-friendly booking systems is more critical than ever. PGBooking aims to address this need by introducing a specialized online booking system focused on Puerto Galera, a renowned destination in the Philippines. The travel and hospitality industry has seen a significant shift towards online platforms for booking accommodations and activities. Puerto Galera, with its picturesque landscapes and diverse attractions, is a prime location for such a system. PGBooking seeks to streamline the booking process for tourists, offering a centralized platform for reserving accommodations and recreational activities within the Puerto Galera region. As a comprehensive online booking system, PGBooking will serve as a digital repository for various services and experiences offered in Puerto Galera. The system's primary objective is to enhance the overall tourism experience by providing a one-stop solution for travelers to plan and book their entire trip seamlessly. The repository 2 will include information about hotels, resorts, and local activities available in Puerto Galera. Through open standards and user-friendly interfaces, PGBooking ensures that users can easily search, retrieve, and book services, optimizing their travel planning process. PGBooking represents a significant advancement in Puerto Galera's tourism infrastructure, providing a digital repository for a wide array of services. This project not only contributes to the efficiency of tourism management but also enhances the overall experience for tourists, thereby promoting the sustainable growth of Puerto Galera's tourism industry. There have been changes in hotel management systems as we are becoming more and more advanced. In the past bookings were made manually inside the hotel. But now it’s been digitalized and now users can just send forms to reserve a room at a hotel remotely. According to (Zhou, W., & Liu, Z. (2022), due to the improvement of people’s living standards, the traditional hotel management model has been unable to meet the needs of customers. Traditional hotel management model also has the defects of low efficiency. (Nandasiri et al., 2022) and (Ukamaka Betrand et al., 2023) both created an online hotel booking system in which customers are offered various facilities such as check-in, checkout, and entry editing, advance payments, option to cancel reservation, produce receipts, and view hotel rooms. These systems have 3 the same features as “PGBooking” does. But (Madhura et al., 2023) took it a step further by integrating AI to the system and having an AI driven chatbot and concierge, designed to deliver user friendly interface, allowing customers to ask questions. “PGBooking” has a similar function only that it is a real time chat with real hotel staff. The thing these researches have in common is that they only feature a single hotel which greatly limits it due to the fact that users will prefer to look and browse through multiple hotels to compare prices and rooms. They are also lacking in producing useful data that can be used in the future as they are only focusing on bookings of the users and only keeping records. Their websites are very simplistic in nature and users don’t have much freedom in them either as they are only allowed to book and browse in a single hotel. “PGBooking” aims to fill these gaps and help not only the hotels but also PG tourism to provide even better hotel booking experience.

**Objectives of the Study**

The project aims to design, develop, and implement an Online Hotel Booking System tailored to the needs of travelers and tourists. This system is designed to simplify and enhance the hotel and resort reservation process, offering an efficient, user-friendly, and comprehensive platform for booking accommodations. It addresses various challenges travelers encounter when searching for hotels and resorts, making reservations, and managing their bookings, all while ensuring secure and seamless transactions. The ultimate goal is to provide an outstanding user experience, along with fostering a culture of trust and reliability in the online hotel and resort booking industry.

Specifically, this study aims to:

1. Allow tourists to book a hotel or resort hustle free.
2. Allows tourists to choose any hotel and resort in Puerto Galera.
3. Create a statistical data demographics like age group and gender.
4. Provide users with the ability to create and manage their profiles, view booking history, and receive booking confirmations.
5. Allows travelers to leave reviews and ratings for hotels and resorts, aiding others in their decision making process.
6. Develop an administrative interface for hotel and resorts administrators to manage room availability, user data, and address customer concerns.
7. Ensure that the system complies with relevant laws and regulations, including data protection and consumer rights.
8. Develop a system for sending automated notifications to users, including booking confirmations, reminders, and updates on their reservations.
9. Develop a predictive hotel and resort performance forecasting and tourists’ inflation forecasting.
10. Develop a performance data of hotel and resort daily, monthly and yearly.

**Scope and Limitations**

This study is centered on the development of an Online Hotel and Resort Booking System for Puerto Galera Tourism, catering to customers, hotels, and resorts as its primary user groups. Customers can effortlessly search for and book resort and hotel rooms, while hotels and resorts can easily manage their webpages and amenities along with seeing all the booking they might receive. The system's design places a strong emphasis on user-friendliness and cross-platform accessibility, accommodating users through web browsers and mobile applications. It integrates essential features for efficient hotel and resort room management, including real-time availability updates, secure payment processing, and automated user notifications. However, it's essential to acknowledge potential limitations related to the rapid evolution of technology and customer preferences may pose challenges in keeping the system up-to-date and competitive. Recognizing these scope and limitations is critical for setting the project's boundaries and objectives.

**Definition of Terms**

* **Hotel** - An establishment that provides temporary lodging, accommodations, and often additional services and amenities for travelers and guests in exchange for payment the process of ordering goods and services over the Internet that allows customers to purchase products or services from a seller using a web browser or a mobile app.
* **Booking -** The process of reserving and securing accommodations or services in advance.
* **Guest** - a person who patronizes a hotel, restaurant, etc., for the lodging, food, or entertainment it provides.
* **Resort** - An establishment that offers a comprehensive vacation experience, providing lodging, recreational activities, and various amenities for guests seeking leisure and relaxation, often situated in scenic or exotic locations.
* **Puerto Galera** - a tourist spot located at Oriental Mindoro, Philippines.
* **Puerto Galera Tourism** - a government group that works and manages the hotels and other activities that promotes tourism in Puerto Galera.
* **Forecasting** - is the process of making predictions or estimates about future events, trends, or outcomes based on past and present data. It involves analyzing historical data and using various methods and models to make informed projections about what may happen in the future. Forecasting is commonly used in business, economics, meteorology, and other fields to aid in decision-making and planning.
* **Amenities** - include in-room extras such as air conditioning, entertainment (TV/music), Wi-Fi, toiletries, robes and more.
* **Review** - a report that gives someone's opinion about the quality of a book, performance, product, etc.
* **Ratings** - a measurement of how good, difficult, efficient a hotel or resort is.

**CHAPTER II**

**REQUIREMENTS SPECIFICATION**

This chapter outlines the specific requirements, features, and functionalities significant to the researchers' system development.

**Hardware Requirements**

Hardware Requirements refer to the representation of the hardware used by the system. Table 1 below presents the hardware requirements to be used by the project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hardware** | **Functions** | **Specifications** | | **Unit** |
| **Minimum** | **Recommended** |
| Computer/  Processor | The specified requirement of an Intel Core i3 11th gen aims to ensure smoother performance in handling the system's data, providing a baseline for efficient operations. However, for an even higher level of performance and smoother operations, it is recommended to use an Intel Core i5 processor or higher. | Intel Core i3 11th gen | i5 and above of version | 1 |
| RAM | The recommendation of 8GB to 16GB RAM serves to facilitate efficient data management, offering a sufficient memory capacity for the system. To achieve an even higher level of performance and responsiveness, it is advised to consider the recommended configuration of 16GB to 32GB RAM or more. | 8GB | 16GB or higher | 2 |
| Storage | The developers will utilize a 256GB to 512GB SSD for efficient storage, access, and management of the system's data. To enhance performance further, it is recommended to consider a larger capacity, such as 512GB to 1TB or more, providing increased responsiveness and accommodating greater data storage requirements. | 256GB | 512GB or higher | 3 |
| Network Connectivity | The modem, with a 35 Mbps download speed, serves as the gateway for the developers to access a quick and reliable network connection for the system. Meeting the recommended download speed of at least 25-50 Mbps ensures faster file uploads and downloads, contributing to an efficient and responsive internet connection within the hardware requirements of our system.  Top of Form | 25mbps | 50mbps or higher | 4 |
| Mobile Devices | IOS and Android mobile devices will be used to test and try out the responsiveness and compatibility of the system in mobile devices enhancing its efficiency and flexibility | IOS11 / Android 7 | Latest | 5 |

**Table 1: Hardware Requirements**

**Software Requirements**

Software Specifications refer to the representation of the software used by the system. Table 2 below presents the software specifications to be used by the project.

|  |  |
| --- | --- |
| Software Used | Description |
| Operating System | Windows 10 and Windows 11 Are the versions of the Windows operating system that will be used for the developers’ system. |
| Laragon | A powerful development environment and server stack for building and managing the researchers’ system. Version 7 to 8.2 is required |
| Visual Studio Code | A streamlined code editor with support for development operations like debugging, task running, and version control. The required version is 1.83.1 or higher. |
| CodeIgniter 4 | An open-source PHP framework. Version 4 is the required version to develop the researchers’ system. |
| phpMyAdmin | The latest version (currently 5.2.1) is required to be used for supporting a wide range of operations and manage the system’s MySQL databases. |
| Vue JS | A progressive JavaScript Framework used for developing mobile apps simultaneously with a single codebase. Version 2.7.14 or higher is required. |
| MySQL | Version 8.0.30 or higher is required for storing and managing the system’s structured data. |
| User Interface Templates | A designer-made web template based on Bootstrap 5, HTML and CSS will be used by developers along with other optional JavaScript plugins for easy customization of the user interface. |
| Web browsers | Samples are Google chrome, Firefox, and Microsoft edge, etc. These are used to access and display the necessary web-based contents and manage the system’s communication with online services and resources. |

**Table 2: Software Requirements**

**Functional Requirements**

This part enumerates the operations and activities that the system must perform. Table 3 represents the descriptions of data requirements, process requirements, and output requirements.

|  |  |
| --- | --- |
| **ID NO** | **Requirement Description** |
| **FR1** | **Data Requirements** |
| **Admin Account** | |
| 1.1 | The admin must input a username and password to access the system. |
| 1.2 | The admin can send notifications to the client with all transactions. |
| 1.3 | The admin can be able to add/modify/view products as well as they can see also all transactions, such as product orders, delivery, and voided transactions |
| 1.4 | The admin must provide a menu management |
| **User Account** | |
| 1.5 | The user must input a username and password to access the webpage |
| 1.6 | The user can add to the cart all the products they want to buy |
| 1.7 | The user can reserve their table slots |
| 1.8 | The user can give suggestions through feedback |
| **FR2** | **Process Requirements** |
| **Admin Account** | |
| 1.9 | The system must accept the registration of the new user. |
| 2.0 | The system must accept all the orders from customers and process them immediately. |
| 2.1 | The system must send notifications email to the client with all transactions. |
| 2.2 | The system must send a confirmation email whenever an order is placed. |
| **User Account** | |
| 2.3 | The system must receive an email notification of the previous transaction. |
| 2.4 | The system can view and track their orders in real-time. |
| **FR3** | **Output Requirements** |
| **Admin Account** | |
| 2.5 | The system must be able to create an inventory management |
| 2.6 | The system must provide Sales Analytics. |
|  |  |
| **User Account** | |
| 2.7 | The system must print all history transaction |
|  |  |

**Table 3: Functional Requirements**

**Non-Functional Requirements**

These are requirements that pertain to behavior properties that a system must have. It defines how a system is supposed to be or its system properties. It contains the following:

**Operational Requirement**

Table 4 represents the requirement description that will specify the operating environment(s) in which the system must perform and how these might change over time.

|  |  |
| --- | --- |
| ID No. | Requirement Description |
| 1.1 | The system is compatible with any operating system |

**Table 4: Operational Requirement**

**Performance Requirement**

Table 4 represents the requirement description that will emphasize the response time, capacity, and reliability of the system.

|  |  |
| --- | --- |
| ID No. | Requirement Description |
| 1.2 | Each web page must load within 2 seconds |
| 1.3 | The user can track their orders 98% of the time without failure. |
| 1.4 | Any user can order and buy the products in the system. |

**Table 5: Performance Requirement**

**Security Requirement**

Table 5 represents the requirement description that will address issues with security, such as who has access to the system's data and must have the ability to protect data from disruption or data loss.

|  |  |
| --- | --- |
| ID No. | Requirement Description |
| 1.5 | The user’s information must be encrypted |
| 1.6 | Only the admin can view the inventory and sales statistics |
| 1.7 | The user’s must be logged in before they can order or can access the homepage of QualiTEApp. |

**Table 6: Security Requirement**

**Chapter III**

**DESIGN AND DEVELOPMENT METHODOLOGIES**

**System Design**

The project developers created a comprehensive plan in order to implement the QualiTEApp system. This way customers can use the QualiTEApp Shop Socorro to place orders and gain easier access to information. Additionally, transactions will be processed more quickly and efficiently. This system is hosted on a website which requires an internet connection for the administrator and customers to use it. Customers who have already registered can access the system through the website.

**Database Design**

The design of the database that includes the many entities on the QualiTEApp is an important part of application development. This gives users information about existing organizations and outlines how to utilize the application.

This shows the tables, including the fields utilized, similar data types, and a description. It gathered, organized, and validated the meaning of specified data phrases. This also demonstrates the link between the two entities.

For the database design, the developers use MySQL for the RDBMS. MySQL allows us to have a relation for tables that needs to be connected to each other. The database design will be for QualiTEApp.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Default** | **Description** |
| id | int | 11 | Not Null | User ID |
| name | Text |  | Not Null | User Name |
| password | Text |  | Not Null | User Password |
| email | Text |  | Not Null | User Email |
| usertype | varchar | 11 | Not Null | User Usertype |
| status | varchar | 20 | Not Null | User Status |
| created\_at | datetime |  | Current\_timestamp | Time Created |
| Updated\_at | datetime |  | Current\_timestamp | Time Updated |

**Table 7. Fields for User**

Table 7 above contains the field name, data type, size, default, and description of the field in the User table. Here, the id is the Primary Key (PK).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Default** | **Description** |
| id | int | 11 | Not Null | Products ID |
| name | text |  | Not Null | Products Name |
| description | text |  | Not Null | Products Description |
| price | float |  | Not Null | Products Price |
| quantity | int | 11 | Not Null | Products Quantity |
| image | text |  | Not Null | Products Image |
| category | varchar | 30 | Not Null | Products Category |
| created\_at | datetime |  | current\_timestamp | Time Created |
| Updated\_at | datetime |  | current\_timestamp | Time Updated |

**Table 8. Fields for Products**

Table 8 above contains the field name, data type, size, default and description of the field in the Products table. Here, the id is the Primary Key (PK).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Default** | **Description** |
| id | int | 11 | Not Null | Cart id |
| userid | int | 11 | Not Null | User id (FK) |
| menuid | int | 11 | Not Null | Menu id (FK) |
| order\_count | int | 11 | Not Null | Cart order\_count |
| size | varchar | 20 | Not Null | Cart Size |
| total | float |  | Not Null | Cart Total |

**Table 9. Fields of Cart**

Table 9 above contains the field name, data type, size, default and description of the field in the Cart table. Here, the id is the Primary Key (PK), while the userid and menuid is the Foreign Key (FK).

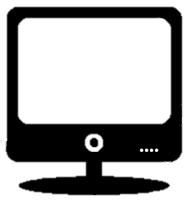
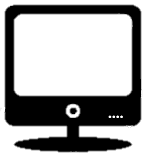
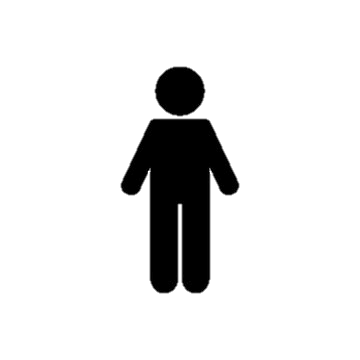
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Default** | **Description** |
| id | Int | 11 | Not Null | Menu ID |
| name | Text |  | Not Null | Menu Name |
| prod\_name | Text |  | Not Null | Menu Prod\_name |
| description | Varchar | 50 | None | Menu Description |
| category | Varchar | 30 | None | Menu Category |
| prices | Float |  | None | Menu Prices |
| discount | Int | 11 | None | Menu Discount |
| image | Varchar | 30 | None | Menu Image |
| stocks | Int | 11 | None | Menu Stocks |
| status | Varchar | 30 | None | Menu Status |

**Table 10. Fields of Menu**

Table 10 above contains the field name, data type, size, default and description of the field in the Menu table. Here, the id is the Primary Key (PK).

**Architectural Diagram/ Block Diagram**

In this section, system architecture was designed to define the flow and behavior of the system’s functionalities to execute its high-quality performance. This covers the formal illustration and description of the project structure.



**Internet**

**Admin**

**Customer Login**

**QualiTEApp Application**

**Transactions**

**Figure 1. System Architecture of QualiTEApp Website**

Figure 1 shows the system architecture of the development of the QualiTEApp Application. It displays the flow and how the system work. The researchers show that the internet is needed in order to access the website of both client and admin to proceed with the intended transactions.

**Data Flow Diagram Level 0**

This section shows the Data Flow Diagram Level 0 which is commonly known as an exploded view of the context diagram that shows the detailed process of how the project works.

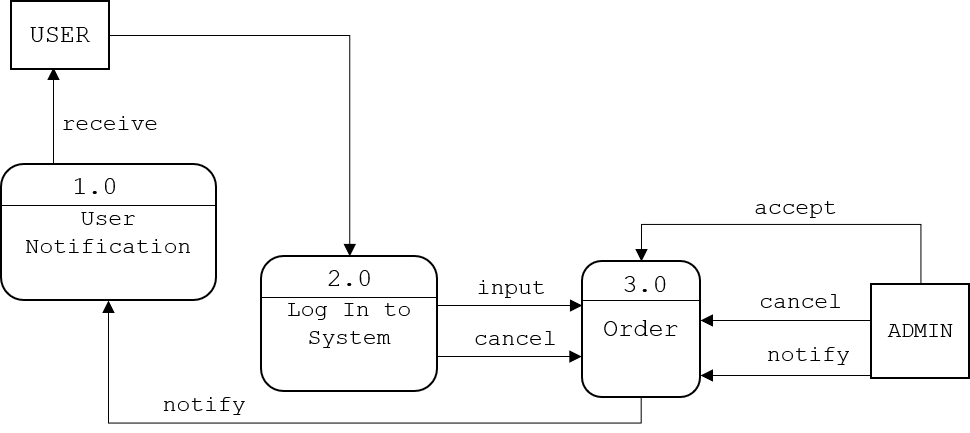
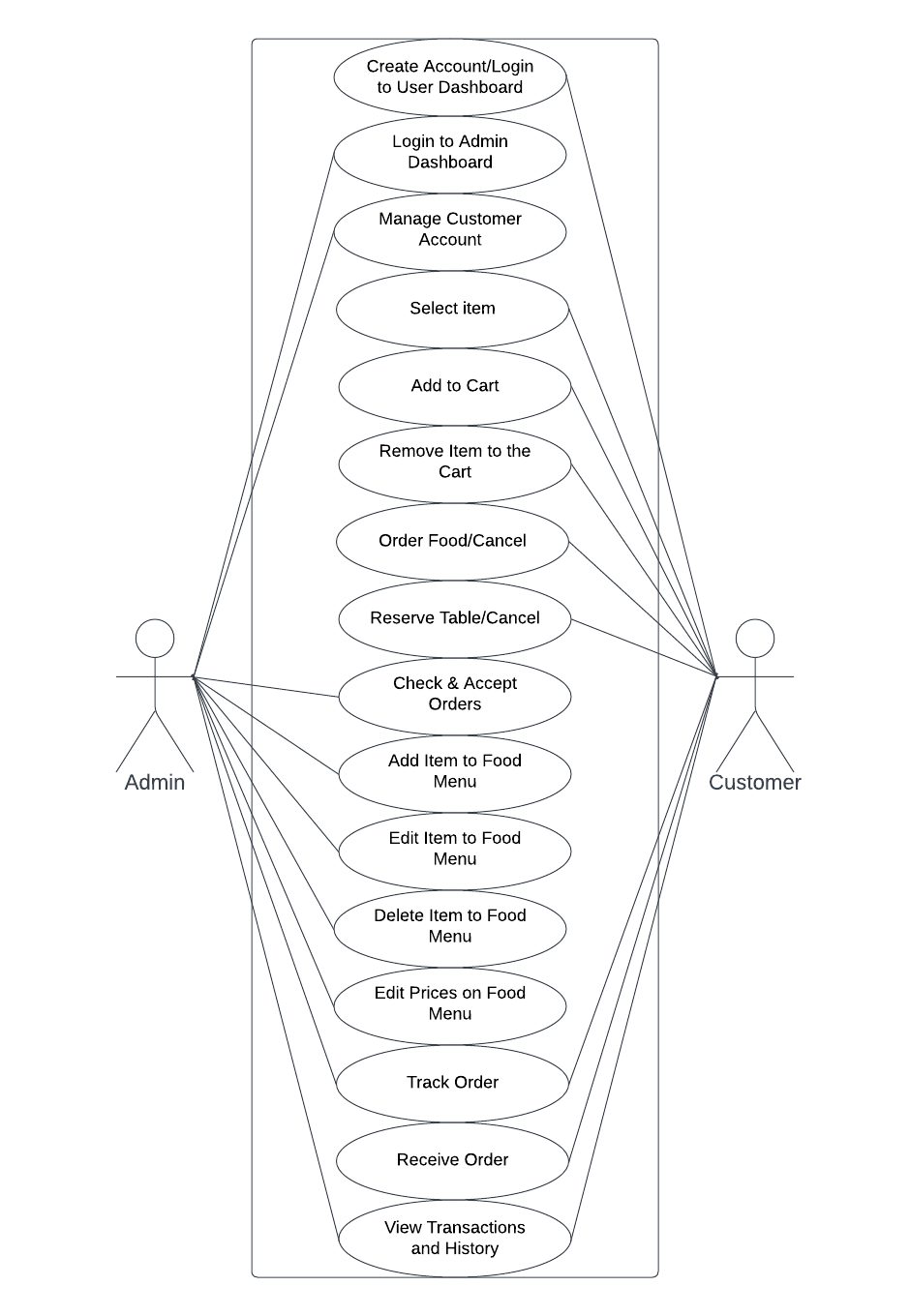
**Figure 2. Data Flow Diagram Level 0**

Figure 2 diagram 0 shows the sequence of the system on the admin side. The system indicates all the functions that the admin can do, It includes user notification, logging in to the system, and order. Admin can select any of the following bases on how they need it and use the system.

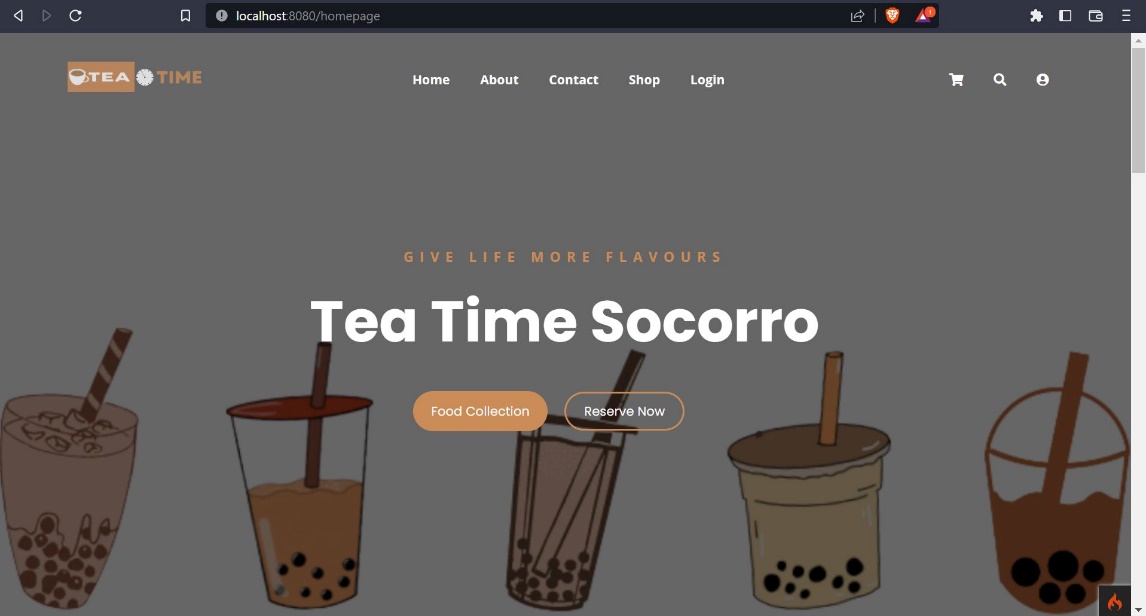
**UML Use-case Diagram**

In this section, the use case diagram summarizes the high-level functions and a graphic representation that depicts the relationship between the system, admin, and users, it is how the system works within interactions of the actors. The diagram will help the researchers to identify and organize the functionality of the entire project.

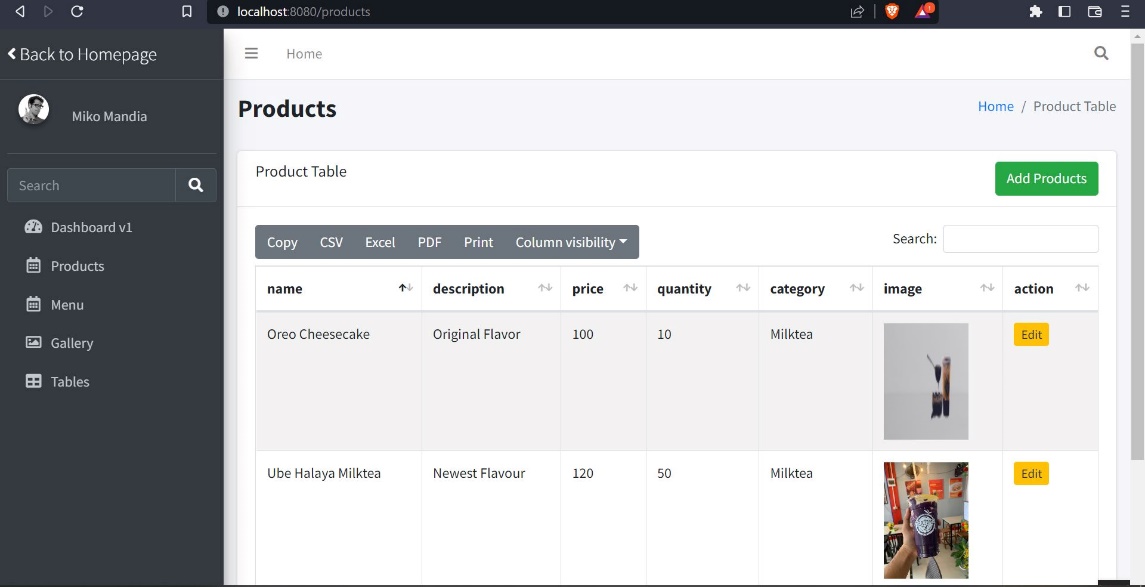
**Figure 3. UML Use-case Diagram**

Figure 3 shows the roles of the Administrator and the customer to be executed in the whole process of the system.

**Sample Mock-up**

A sample mock-up is a visual representation of a website after it is built. It consists of visuals that show how the website should look and its function. It is used to refine the design, identify potential problems, and ensure that the system meets the user's needs and expectations. Below are the system users and admin interface.

**Figure 4. User Interface**

****

**Figure 5. Admin Interface**

**Development Method**

The “QualiTEApp: An Ordering Management System For Teatime Shop Socorro” will employ the System Development Life Cycle (SDLC) to ensure the production of an efficient and high-quality system. This project will use an Iterative and Incremental Model, which allows for improvements to be made throughout the development process. The SDLC is divided into four phases: collecting requirements, design and development, testing, and implementation. These steps will be followed in order to create a successful system.

**Requirements Gathering**

**Testing**

**Implementation**

**Design & Development**

**Testing**

**Implementation**

**Design & Development**

Build 1

Build 2

**Testing**

**Implementation**

**Design & Development**

Build 3

**Planning**

**Maintenance**

**Figure 6. SDLC Iterative and Incremental Model**

The research followed the following phase.

1. **Planning.** In this phase, researchers conduct interviews and observations according to the problem that needed to be resolved. Researchers developed the objectives and plans of the project that serves as the foundation to produce the desired output of the system at the end of the study.
2. **Requirements Gathering.** In this phase, researchers will gather more information and data requirements needed for the development of the system. Also, determine the functional and non-functional requirements of the project.
3. **Design.** During this phase, researchers will begin to design the hardware and software of the system according to the requirements of the users. The developer will thoroughly plan and implement trial and error to be able to meet the desired output of the user interface.
4. **Development.** During this phase, researchers will start the coding process for the functionality of the system through Visual Studio Text Editor Application and PhpMyAdmin as administrator tool for the database, wherein database will serve as data storage of the system and CodeIgniter 4 as the framework of the proposed system.
5. **Testing.** In this phase, researchers will execute the pre-deployment of the system for trial and error just to see if the device is properly functioning.
6. **Implementation.** In this phase, researchers will create the iteration of the project after the testing process in order to analyze and improve the design and functionality which needs more improvements to meet the project objectives.
7. **Maintenance.** In this phase, researchers must maintain the functionality of the system during deployment to maintain its performance. Also, researchers will determine and fix the errors that may occur over a period of time.

**Gantt Chart**

In this section, Gantt Chart is presented to show the plans and schedules of the project timeline. All the development stages up to the completion of the project were documented in this chart. This helps the researchers to know the deadlines needed to accomplish and show breakthroughs in various tasks.

**Table 11. Gantt Chart**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task Name** | **Task Date** | | | | | | | | | | | | | | | | | | | |
| Oct | | | | Nov | | | | Dec | | | | Jan | | | | Feb | | | |
| Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | Week 2 | Week 3 | Week 4 | Week 1 | Week 2 | Week 3 | Week 4 |
| **1.Planning** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.1 Conduct an interview |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.2 Define project objectives |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.3 Define project plan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.4 Approval of project plan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2.Requirements Gathering** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1 Data Collection |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.2 Functional |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.3 Non-Functional |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3.Design** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1 Frontend software design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **4.Development** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.1 Back-end coding |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **5.Testing** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5.1 Functionality testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5.2 User interface testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **6.Implementation** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **7.Maintenance** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7.1 Project monitoring |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7.2 Resolve system errors |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Legend:** - Completed/ Done

Table 11 shows the whole process of developing QualiTEApp. It displayed the various tasks and marks as completed oats certain date. Researchers will be kept informed of the progress of the development which will help them not to miss out on steps and differentiate tasks from the amount of time took to complete them.

**Testing and Evaluation**

This section is referring to the testing and evaluation phase of the software development process. During this phase, developers are required to conduct tests on their system to determine its capabilities and limitations. This will allow them to identify any issues or potential problems prior to the production and deployment stages. The tests should include all the requirements outlined in the Requirements Phase, such as design, performance, supportability, etc. The results of these tests will be evaluated to assess the progress of the system and ensure it meets the requirements of the project. The developers considered the following:

1. **Unit Testing**- a type of software testing where individual units or components of a software system are tested. This type of testing is usually done by the developers, as it requires detailed knowledge of the internal structure of the system. Unit Testing is designed to test individual functions, modules, and features of the system, to ensure that they all behave as expected
2. **Component Testing**- involves testing the individual components of the system (modules, classes, objects, and programs) in isolation, without integrating them with other components. This helps to identify any defects or bugs in the individual components before they are integrated into the system. It also helps to identify any unexpected interactions or dependencies between components that could cause problems or errors in the system.
3. **System Testing**- a type of software testing that evaluates the entire system or application and its components to verify that all individual modules are working properly and that data is transferred accurately between modules and the entire system. System testing is meant to ensure that the system meets its requirements, performs as expected, and functions correctly in its intended environment. It is an overall test of the system and its components, and it is typically done after unit and integration testing.
4. **Unit Acceptance Testing**- a process to validate if the unit (which could be a software, product, or service) meets the requirements of the end-users and clients. During unit acceptance testing, end-users and clients interact with the unit and provide feedback on its features, usability, and performance. Based on their feedback, they either accept or reject the unit. It is an important step in the development process as it ensures that the unit meets the expectations of the use

**REFERENCES**

Aini, Q., & Putri, D. M. (2019). Effectiveness of Installment Payment Management Using Recurring Scheduling to Cashier Performance. Aptisi Transactions On Management, 3(1), 13-21.

Aini, Q., Dhaniarti, I., & Khoirunisa, A. (2019). Effects of ILearning Media on Student Learning Motivation. Aptisi Transactions On Management, 3(1), 1-12.

R. Tamilarasi and N. Elamathi, “E-Commerce- BusinessTechnology- Society,” vol. 4, pp. 33–41, 2018.

Huseynov, F., & Özkan Yıldırım, S. (2019). Online Consumer Typologies and Their Shopping Behaviors in B2C E-Commerce Platforms. SAGE Open, 9(2).

Alalwan, A. A. (2020). Mobile food ordering apps: An empirical study of the factors affecting customer e-satisfaction and continued intention to reuse. International Journal of Information Management, 50, 28-44.

Lim, S., & Noroña, M. I. A Study of Stakeholders Perception of Factors Affecting Online Food Delivery Service Industry in the Philippines.

Gamilla, C. O. (2021). Revenue boom of Online Food delivery industry amidst Pandemic: Evidence from the Philippines. International Journal of Progressive Research in Science and Engineering, 2(4), 24-26.

Doctora, A. L. S., Perez, W. D. D., Sanchez, F. N. A., Vasquez, A. B., & Borres, R. D. Application of Analytical Hierarchy Process in the Comparison of Jollibee, McDonald’s, and KFC In-House Online Delivery System.

Guerrero, V. G. G., Iii, R. J. T. B., Borden, H. R. N., Iii, M. L. G. B., Gonzales, X. K. A., Yamagishi, K. D., & Ocampo, L. A. (2023). Modelling the barriers of online shopping in the Philippines using the ISM-MICMAC approach. International Journal of Business and Systems Research, 17(1), 75-96.

Castillo, A. M. M., Salonga, L. J. L., Sia, J. A. L., & Young, M. N. Improving Fast-Food Restaurants’ Method of Operation: Automated Drive-Through Ordering System.