**Deliverable 4 -** **Project Phase 2**

**DineSys: Comprehensive Restaurant Management System for Seamless Dining Experience**

# Group Name:

**TEAM SE's**

# Group Members:

* **Vangala Ruchitha (Team Leader)**
* **Nikhila Polkampally**
* **Nandhini Kasukurthi**
* **Vishal Rachuri**
* **Sai Srinivas Valleti**
* **Chandralekha Gude**
* **Aishwarya Karukonda**
* **Hunny Keshwani**
* **Leela Varadatta Sai Addanki**

## **Executive Summary**

* 1. **Project Overview**

In these days when technology has gone through huge steps some of the everyday life jobs keep being very hard and tiring. One of them is management of a restaurant.

Anyone can think this as a small thing but not for them who spent their life dealing with this. Huge restaurants, with amazing views and menus do have service problems. Waiters try their best but some time are not capable to cover and please every client in time and at the end of day they will be more tired than usually. So owners are obligated to hire more waiters and give more money from what he gains. But the new restaurant management system will make everyone happy. The client will be able to order directly from the table right after sit by choosing the dish through a tablet found there. The order will be received as a message from the chef and after being prepared will be served by the waiters. So waiters will not forget the orders and get only tired, will have less work to do, the chef will not have some pieces of papers and be disorganized, and manager will not have to pay a lot of employers and at the same time will be something very elegant and new for clients too. At the same time this will help a good part of youngers be employed and earn money so they will not be a burden to their families and will get better and professional to this branch of technology. It is also important to mention that this will be a good step to advance more in technology and will be good for a country in development as the situation that we are facing in these moments.

* 1. **Purpose and Scope of this Specification**

The purpose of this new method is to help the restaurant to have a better management and at the same time make clients feel more pleased and come back at the same restaurant. With this method we think that the service will get better. Service getting better the number per day of clients will be higher so the incomes will get better at the same time. At the same time the managers will be able to control the incomes in a better way because everything will be recorded. But these businesses are very effective in the economy of the state will affect in it too. So starting with some of the restaurants as an experiment which is believed to be very successful and helpful at the same time. Of course the priority are known and famous restaurants which will be able to afford this new technology. So will be continued to other restaurants and those who will embrace

this new method. This method at the same time will be a good way for the municipality to control the incomes of the restaurants that will apply this technology.

## **Product/Service Description**

In our country management of a business as restaurant and its service are not helping the situations to get better. This product is an experimental decision taken by a group of researchers who had different meetings with some restaurant managers.

It will be firstly used by the three or four restaurants which will give stable and continuous information according to this new method.

It is thought to be installed and get under control by a group of researches made by economists and managers.

It will help:

* Better service
* Easy management and income control
* Less hand- work
* Not tired employers
  1. **Product Context**

This is a software application designed to manage the activities within the restaurant itself but also to serve as a website for the customers to see the facilities this restaurant provides.

It is supposed that each table is going to have its own tablet from which the customers can select on the menu what they want. After the order is confirmed from the customer, the chefs will know and send a notification in the table’s tablet that the order is being prepared. When the client’s plates are ready, the waiters get a message to deliver them to the destination table. The users of this application which are listed below are going to

communicate with each other through notifications. Each of them will be able to send a notification to the other members.

* 1. **User Characteristics**

This application can be accessed by all the members of the staff:

* Admin
  + Can check notifications
  + Hire/add employee
  + Add/delete tables
  + Add new items in the menu
  + Update the total amount of money
  + See the whole list of products
  + Can see the whole list of employee and edit it
  + See all the orders
* Accountant
  + Can see the list of notifications
  + Can see the list of whole products
  + Add/delete products
  + Can edit the salaries of the employees
  + Can see the list of suppliers
  + Can contact the suppliers
  + Can add and delete suppliers
* Chefs
  + Add new items to the menu
  + Can see the orders which has not been confirmed yet
  + Notifies the waiter when the order is ready
  + Can send notifications
* Waiters
  + Can see the completed orders confirmed by chefs
  + Update the total confirming the payment of the orders
* Other
  + Make check-in when they start their working day and leave it

Tables also will have a login page: each page will have a unique number and a password. For each table, there will be recorded all the orders received.

* 1. **Assumptions**

It is assumed that the administrator profile is added b the creators of the application and then the administrator is responsible for adding, deleting other staffs’ profile.

It is assumed that all the staff is trained to use the application in order to avoid misconceptions.

It is assumed all the clients will be serious and not try to use the tablets for anything else except their primary function: to order the food.

* 1. **Constraints**

The project is constrained by the financial state of the restaurants that are going to use it. Since the application is going to be put in tablets in the tables in the restaurant, it is crucial that the owners are willing to initially invest their capital in some qualitative tablets. The project is constrained by the Internet connection. Since the application fetches data from the database over the Internet, it is crucial that there is stable Internet connection for the application to function.

* 1. **Dependencies**
* The chain starts from the client who does the order. Directly the order goes to the chef.
* Chef prepares the order and automatically products used for the order are subtracted from the database and if there is anything absent any product or tool chef sends notification message to accountant. Also when order is ready notify the waiter.
* Waiter takes the notification and sends the order from the chef to the client.
* Accountant is engaged on employees’ salaries distribution and if gets any notification for tool or products missing make orders to the relevant suppliers companies.
* Above all of the employees, there is Administrator who supervises all their actions.

## **Report Requirements**

* 1. **Functional Requirements**

The requirement numbering follows the scheme - BR\_ ##

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Requirement** | **Comments** | **Priority** | **Date** | **Reviewed/**  **Approved** |
| BR\_01 | The software should have different views for different user levels | The view for the client, chefs, accountant and administrator will be  different. | 1 |  |  |
| BR\_02 | Administrator is responsible for registering all the staff members and tables into the management system, applying the predefined rules by the conventional system of the restaurant. | Usernames will be in the format name.surname and the password generated for each user will be in the format NameSurnameBirthYe ar, for the tables  Table-Number. Users can change their passwords  after.(recommended) | 1 |  |  |
| BR\_03 | No staff member can edit the username. | Editing usernames conflicts with our operational intelligence. | 2 |  |  |
| BR\_04 | In case a staff member leaves the job/is fired, the administrator has to delete his account from the system. | All the personal data and transactions between them and the restaurant will be erased from the  system. | 2 |  |  |
| BR\_05 | The software needs to add  modify and delete users/tables. | CRUD functionalities  possessed by the administrator | 1 |  |  |
| BR\_06 | A user should have a profile page. On the profile page a user can edit his/her information, which includes the password, e- mail address and phone number. The user can  modify the personal information in his/her |  | 1 |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | profile. |  |  |  |  |
| BR\_07 | Staff members' accounts should be secured with passwords. | For ethical and security reasons, passwords will be hashed before being  stored in the database. | 1 |  |  |
| BR\_08 | Tables' accounts should have the menu page displayed as the main page. Clients can easily find their food by clicking  into the favourite category. (meat, pasta, rice, pizza) | This will provide a practical and easy method for the visualization of the dishes. | 1 |  |  |
| BR\_09 | When a costumer clicks "+" symbol, the corresponding dish will be added to the Order List on the right side of the page. | This will give real-time information about the price of your order, as well as help the costumer not forget what he had previously ordered.  The costumer can remove his choices  from the ordering list as well. | 1 |  |  |
| BR\_10 | Given that the costumer confirms the order, a timer will be displayed, showing the remaining estimated time until dish is ready. |  | 3 |  |  |
| BR\_11 | Upon confirmation, the order will be added to the queue list of the chefs,  until one of them takes the order. | This is a quick way to order food and doesn't need an intermediary such as waiters. | 2 |  |  |
| BR\_12 | When the chef clicks "Ready" for the respective dish, the ingredients of the dish and their amounts are subtracted from the total  amounts, stored in the database. | This is a more efficient way of managing food products dynamically, instead of spending a lot of time doing an  inventory by the end of the week. | 1 |  |  |
| BR\_13 | As soon as the customer pays the bill, the waiter/chef confirms the payment and table's status  is set to available. |  | 2 |  |  |
| BR\_14 | When a payment is confirmed, the amount of money is added/subtracted to the current total amount  of money owned by the restaurant. | This avoids stealing or other mistakes made during calculations, as everything is done  automatically. | 1 |  |  |
| BR\_15 | When the payment is confirmed, the software prints the bill. |  | 3 |  |  |
| BR\_16 | The customer can rate the ordered dishes from 1-5 | 1 star means the customer didn't enjoy | 3 |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | stars. | the food, 5 stars means the customer is  fully satisfied with the dish. |  |  |  |
| BR\_17 | Given that the accountant is logged in, he should be provided with the following 3 lists:   1. suppliers list 2. product list 3. worker list | These are among the key sources of data that come to the restaurant. | 1 |  |  |
| BR\_18 | A chef can notify the accountant about missing/broken/needed kitchen gadgets, by sending notifications/requests. | The accountant receives the notification, reads it and orders whatever is needed, by contacting the distributors. | 2 |  |  |
| BR\_19 | The notifications will be in the form of text/description where the request is specified. |  |  |  |  |
| BR\_20 | When the amount of a certain product is below a threshold, an automatic notification is sent to the  accountant. | These automated requests avoid the need for a product inventory. | 1 |  |  |
| BR\_21 | Every transaction managed by the accountant is saved in  pdf/text format or is sent to administrator. | It makes every process be transparent and well- documented. | 2 |  |  |

* 1. **Non-Functional Requirements**

**3.2.1 User Interface Requirements**

A software application is as good as the interface it provides to its users. Appropriate performance, easy navigation, elegant and stylish design, fast response times make the difference to a system’s utility. The user interface for this software is designed to be well suited to any browser as Chrome or Mozilla and can be accessed through portable tablets which are very practical and easy to use.

This software will include: Log In interface, Administrator Interface, Chef’s Interface, Waiter’s interface, Accountant’s Interface.

Log In interface will include:

* Restaurant Logo,
* one box to write the username
* one box to write the password
* and a log in button

Administrator’s interface will include:

* Worker button
* Tables button
* Add on menu button
* Orders button
* Log out button
* All restaurant tables as buttons

Chef’s interface will include:

* Add new dish button
* Log out button
* A table where which row involves: table number, Order, Accept button, Done button.

Waiter’s interface will include:

* Log Out button
* A table where which row involves: table number, Accept button, Payment Confirmation, Print Receipt.

**3.2.2 Usability**

Usability is an important attribute which defines flexibility of the software. This software shall be able to work in a practical and easy way for clients, employees and administrator.

Employees shall find it very comfortable using the software on the tablet during work hour and also clients shall find it not difficult to understand how to use it. In this way all software users can achieve their objective with effectiveness and satisfaction.

**Learnability**

Learnability is the capability of a software product to make as simple as possible for the user to learn its application.

* This software is designed to be easy to learn, easy to use, subjectively pleasing.
* Clients can log in. Employees can log in. This software is designed to be used by all kind of people from all ages. There are instructions and alert messages that will be shown during execution of each task.
* Icons and Menu options facilitate the user to perform their tasks. For example clients find any dishes that they impact - a right click on a button displays a list of specific dishes restaurant offers.
* Provide respond messages that a user command has succeeded or advise of failure. If you try to log in with a wrong username/password it will be shown a feedback message that the log in process failed due to wrong credentials.
* Provide instruction messages for user interface components: dialog boxes, fields that require input and image details. For example, a if you type a password with less than a certain number of characters it shall display :”No less than 6 number of characters is
* allowed”.

**Accessability**

We know that an application have to be functional, but can this application be accessed by everyone? In this application users are staff employees, administrator and client consumers. This software tends to simplify as much as possible accessability of these users through:

* Employees working in the restaurant can access the software through their accounts.
* Clients ordering in the restaurant can access the software when they reserve a table.
* Administrator of the restaurant can access the software through his account.
* Every user level will have access to resources that belong to his/her interest.
* Taking in consideration all the conditions of the client this software is applicable and usable also for the people with disabilities such as :

Hearing-You can look for details of dishes on the portable table instead of asking the waiter, you can order through tablet instead of communicating with the waiter.

Speaking-The ordering can be done automatically,

People who can’t lift and carry anything, walk and use stairs-ordering is done through tablet, waiter bring the order to the table.

**Memorability**

After the client learns how to navigate this software and find what they are looking for, they need to be able to remember how to do it when they come back. Memorability is a measure of how easy software is to remember after a substantial time lapse between visit. Design of the software through icons and instructions encourages the increasement of client memorability.

* + 1. **Performance**

In order to assess the performance of a system the following components must be clearly specified:

* Response Time
* Workload
* Platform

Response Time of the software is another component to be taken in consideration. Log in or ordering on tablet shall be processed in a few milliseconds. Our motto is: “Our client is sacred. Don’t let him/her waiting to order, don’t let him/her waiting for the order.” Workload defines the capability of the system to handle the maximum of the clients interacting with the system. This software shall be able to offer the required service even to the maximum capacity of the clients.

A platform is defined as the combination of both hardware and software which will house the system. So this platform to be able to offer the best software design and the best functionality.

### Capacity

The ONLY measure that is meaningful and relevant when it comes to defining the Capacity needs of an application is the MAXIMUM.

The performance of the web application shall support all the tasks of the restaurant employees, clients and administrator to be performed on the maximum capacity in order to be functional in each possible restaurant population conditions.

Since this application uses a single database, there may come a time when it can become congested and overloaded, affecting the time of data fetching.

### Availability

The restaurant is open from 8 o’clock am to 12 o’clock pm, so the system shall be available 24/7, but its maximum availability is required during the specified time interval. Availability is an essential ethic component that is closely related with good reputation of the restaurant. So this software’s purpose is to offer maximum availability.

### Latency

The latency of RMS will mostly depend on:

* Internet connection speed
* Efficiency of the fetching data from the database algorithms + processing and calculation algorithms.
* Size of the database being used.

# Manageability/Maintainability

* + - 1. **Monitoring**

Monitoring is an approach of defining and checking the performance characteristics of software systems. The purpose of this task is to ensure that this software covers successfully the issues below:

* + - * 1. Whether the application is running.
        2. Unusual tablet/memory/network usage.
        3. Report any unhandled exceptions.
        4. Status of external components (databases, etc.)
        5. Number of pending tasks.

It is administrator’s task to handle and correct the errors, by following specific procedures with the appropriate validations and tests.

* + - 1. **Maintenance**

MySQL is the platform of our database and server type is Apache. In case of a failure, a re-initialization of the program is recommended. If it is not the case, that means that the server may be down, so the user needs to wait for the system administrator to start the server.

If any extension or modification needs to be done, then it will be very easy. Since PHP is object oriented also, we’ve created the appropriate classes with the right functions needed for our software’s functionalities. Therefore if you need to add or remove lines in the code, you just need to find the right class and change what you need to. Modularity provides for a better maintenance.

In case of bugs correction, we will be able to deliver the updated and improved version of our software.

* + - 1. **Operations**

Operational requirements are the basis for system requirements. They define the essential of functionality.

Some of the operations required by the users include:

•login of chef

* login of waiters
* login of accountant
* login of administrator
* ordering for clients
* mediation of dishes with products
* service offered as a chain from chef and then from waiter
* CRUD of employee profiles, tables, products and dishes.
* financial assistance from economist for employees and restaurants products
* communication of clients with staff
* communication of the accountant with the suppliers
* generation of PDF invoices and bills.
  + 1. **System Interface/Integration**

System integration is defined as the process of bringing together the component sub-systems into one system (an aggregation of subsystems cooperating so that the system is able to deliver the overarching functionality) and ensuring that the subsystems function together as a system. Our project is not considered as a big software and that’s why we have a whole system, not organized into sub-system. But it is organized in modules, Accountant, Chef, Tables, Chef and Admin, which communicate with each other using notifications and queries.

* + 1. **Security**
* The surveyor of the software will be administrator so he/she will be the the only one who monitors and accesses the data for everyone on the restaurant.This software offers to every user the highest system of security.
* To meet the security objectives, a certain security components shall be covered such as: Identification, Authentication, Authorization, Privacy Requirements, Physical Protection Requirements, System Maintenance.
* These components shall be achieved through:
* The application shall identify all of its clients, staff employees, administrator before allowing them to access its capabilities.
* Specific authenticated externals shall access specific software component capabilities or information if and only if they have been explicitly authorized to do so by a properly appointed person(s).For example: The account of waiter shall not be accessed by the chef.
* Unauthorized individuals shall not gain access to users credentials.
* The restaurant security staff shall protect its hardware components from physical destruction, theft, or any kind of damage. Despite software security, it’s important and hardware security.
* The application shall not violate its security requirements as a result of the replacement of a data, hardware, or software component.
  + - 1. **Protection**

The software is accessible by a username and an encrypted password. Only by these two credentials the worker or the client can use this utilitary application for their purposes.

* + - * 1. In order to increase the protection of information, encrypting passwords should be applied to each user registered.
        2. If an employee wants to edit its profile, the administrator should be alerted through a notification message.
        3. The password of an employee must be changed anytime,it’s up to the employee’s preferences.
        4. The username and password for administrator might be edited anytime based on administrator preferences.
      1. **Authorization and Authentication**

An authentication requirement is a component of security requirement. It shall verify the identity of its externals before interacting with them.

* + - * 1. The software shall verify the identity of all the users who access it before allowing them to use its capabilities. Employees, clients or administrator can have access to the system through their credentials: username, password.
        2. The software shall verify the identity of all of its users before allowing them to update their user information. An employee can’t edit his/her profile without logging in.
        3. The software shall consider the repeated requested validation failures as fraud. If a client or a worker makes repeated requests to log in to the system with a wrong username/password the client or worker shall not be logged in.
        4. The software shall not allow any employee to access any account information of any other worker.

**3.2.7 Data Management**

This application is based on MySQL language.

There will be some tables which are going to be connected with each other with joint tables. For example: each order will have the id-s of each table, each chef and each waiter. There will be a table for the staff and a table for the tables of the restaurant.

Each user entry or table entry is protected by a password.

* + 1. **Standards Compliance**

This method is also approved by the trade office and will be at the same time under their control. Their function will also be controlled by this office in co-operation with our group. It is important to mention that will not be allowed any use of this method wrongly and will be only one center where every restaurant that wants it should apply. The price of using this method will be according to economical standards of the country and of course by analyzing the conditions of the restaurants. This method policy will also be useful for the financial office to control the incomes of the restaurant better for them not to have frauds.

* + 1. **Portability**

This application can be accessed by everyone who wants to see the facilities this restaurant provides, but only the members of the staff will have a username and a password to login so to collaborate with other members of this online community.

* 1. **Domain Requirements**

This system manages and covers everything everything related to the restaurant, other remaining tasks are performed by the responsible people without the need of computers. Therefore our system doesn’t need to communicate with other systems of the same business. Therefore our project doesn’t have any specific domain requirements.

# SOFTWARE ANALYSIS AND DESIGN

### User Scenarios

These are the user scenarios of RMS:

**USERS:**

1. Admin
2. Accountant
3. Chefs
4. Waiters
5. Customers

**ADMIN**

1. Taps Log In in the main Page
2. Enters his/her credentials
3. Will be redirected to Admin Page, with all the functionalities provided for the admin status such as

* CRUD options for software users
* Table Management
* Add on Menu

**ACCOUNTANT**

1. Taps Log In in the main Page
2. Enters his/her credentials
3. Will be redirected to Accountant Page, with all the functionalities provided for the accountant status such as:

* Communication with food companies (suppliers)
* List of products and their specifications + management
* Management of workers’ salaries
* PDF files documenting every transaction

**CHEFS**

1. Taps Log In in the main Page
2. Enters his/her credentials
3. Will be redirected to Chef Page, with all the functionalities provided for the chef status such as:

* Order Management
* Add new items in the menu
* Make requests about new kitchen equipment

**WAITERS**

1. Taps Log In in the main Page
2. Enters his/her credentials
3. Will be redirected to Waiter Page, with all the functionalities provided for the waiter status such as:

* Table Management (busy/available)
* Order payment confirmation
* Receipt printing

**CUSTOMERS (USING TABLES ACCOUNT)**

1. The tablet on their table displays the menu page.
2. Customer chooses one of the categories (pizza, pasta, meat etc.) and clicks on one of the dishes, in order to add it to the order list.
3. Customer confirms the order, by clicking SEND ORDER.
4. As soon as the order is confirmed, the customer is displayed the home page, stating the order number and content at the top.

*More Detailed User Scenarios*

**Scenario 1 – Successful Login**

1. The user enters his username.
2. The user enters his password.
3. The user attempts to Login after the fields are filled in.
4. If there is a match with an entry in the database, he is logged in to his account.
5. The user is redirected to the home page of his account.

**Scenario 2 – Unsuccessful Login**

1. The user enters his username.
2. The user enters his password.
3. The user attempts to Login after the fields are filled in.
4. If the credentials are wrong, there won’t be a match in the database, therefore the user will be displayed an error message, saying “Wrong credentials, enter your credentials again.”
5. The Login page will be refreshed, so the user can enter the credentials again.

**Scenario 3 – Admin creates a new user**

1. Administrator of the software is logged in the system.
2. Administrator goes to the Staff Page.
3. Administrator creates a new Staff Member.
4. Administrator is displayed a form, with the basic information needed to be provided.
5. Administrator fills in the information, regarding name, surname, username, initial password, phone number, category and an optional picture.
6. Administrator confirms the procedure after filling in the spaces.
7. Member is created as an entry in the Employee table of the database.
8. Administrator is redirected to the Staff Page.

**Scenario 4 – Admin faces an error while creates a new user**

1. Administrator of the software is logged in the system.
2. Administrator goes to the Staff Page.
3. Administrator creates a new Staff Member.
4. Administrator is displayed a form, with the basic information needed to be provided.
5. Administrator fills in the information, regarding name, surname, username, initial password, phone number, category and an optional picture.
6. Administrator confirms the procedure after filling in the spaces.
7. If the validation functions detect an anomaly, Administrator is displayed an error message, telling him where the problem with the entered data is.
8. Administrator enters the data again, until there are no validation problems.

**Scenario 5 – Admin modifies user information**

1. Administrator of the software is logged in the system.
2. Administrator goes to the Staff Page.
3. Administrator searches through the Staff members list the name of the user he wants to modify.
4. After finding the user, he starts the editing procedure.
5. Administrator is displayed the page containing user’s details, which are editable by the administrator.
6. After editing, administrator saves changes.
7. Administrator is redirected to the Staff page again.

**Scenario 6 – Admin modifies user information**

1. Administrator of the software is logged in the system.
2. Administrator goes to the Staff Page.
3. Administrator searches through the Staff members list the name of the user he wants to modify.
4. After finding the user, he starts the editing procedure.
5. Administrator is displayed the page containing user’s details, which are editable by the administrator.
6. If there was a mistake during the process, the administrator will be displayed an error massage, telling him where the problem with the entered data is.
7. Administrator enters the data again, until there are no validation problems.

**Scenario 7 – Admin inserts a new table in the restaurant.**

1. Administrator of the software is logged in the system.
2. Administrator goes to the Tables Page.
3. Administrator adds a new table in the restaurant.
4. A window will pop-up, asking admin to enter the table number, table password and number of chairs.
5. After clicking “Add”, the table will be added as an entry in the database.

**Scenario 8 – Accountant inserts new products.**

1. Accountant of the software is logged in the system.
2. Accountant goes to the Products section.
3. Accountant wants to add a new product.
4. Accountant will be displayed a form, with the entry information needed to be provided.
5. Accountant fills in the information, regarding the product.
6. Accountant saves the new product entry.
7. Accountant is displayed a confirmation table, indicating success.

**Scenario 9 – Admin inserts new entries on the menu.**

1. Administrator of the software is logged in the system.
2. Administrator goes to the Menu Page.
3. Administrator wants to add a new entry on the restaurant’s menu.
4. Administrator will be displayed a form, with the entry information needed to be provided.
5. Administrator fills in the information, such as the name, price, ingredients and picture.
6. Administrator saves the new entry.
7. Accountant is displayed a confirmation table, indicating success.

PS: Normally, this is a feature of Chefs but since there may be a radical change in the menu, for unknown reasons, it wouldn’t be convenient for the chefs to insert all the dishes, that’s why administrator can do it as well.

**Scenario 10 – Accountant searches a supplier in the Supplier’s list.**

1. Accountant is logged in the system.
2. Accountant goes to the Supplier Page.
3. Accountant is shown a table with all the suppliers of the restaurant and a search icon in the corner.
4. Accountant starts typing the name of the supplier and table gets updated every time a key in pressed, displaying search results.

**Scenario 11 – Accountant adds a new supplier in the Suppliers list.**

1. Accountant is logged in the system.
2. Accountant goes to the Supplier Page.
3. Accountant is shown a table with all the suppliers of the restaurant.
4. When accountant wants to add a new Supplier, a popup window appears, displaying a form where the new supplier’s information is entered.
5. After entering the required information, accountant saves the changes and if the process is successful, he will be redirected to the Supplier’s list.
6. Otherwise, display an error message.

**Scenario 12 – Accountant edits the information of the food products.**

1. Accountant is logged in the system.
2. Accountant goes to the Product section.
3. Accountant is shown a table with all the food products needed for the restaurant.
4. When accountant wants to edit the information of some product, it can change the content of the text fields of quantity and price.
5. Accountant can edit them and then save changes after.
6. Accountant is redirected to the Products page.

**Scenario 13 – Accountant searches a product in the Product’s list.**

1. Accountant is logged in the system.
2. Accountant goes to the Product section.
3. Accountant is shown a table with all the food products of the restaurant and a search icon in the corner.
4. Accountant starts typing the name of the product and table gets updated every time a key in pressed, displaying search results.

**Scenario 14 – Accountant updates the monthly salary for an employee.**

1. Accountant is logged in the system.
2. Accountant goes to the Staff section.
3. Accountant is displayed a list of the employees and their salaries next to their names.
4. If it doesn’t display the salary next to the name, then accountant knows he hasn’t updated the salary of that employee for the current month, so he does it.
5. After entering the salaries, accountant saves the changes.

**Scenario 15 – Accountant generates a PDF document with all the salaries of the employees for that month.**

1. Accountant is logged in the system.
2. Accountant goes to the Staff section.
3. Accountant is displayed a list of the employees and their salaries next to their names.
4. Account selects the print option and a PDF containing all the salaries and total amount is printed, which will be later signed by the accountant.

**Scenario 16 – Accountant orders food products.**

1. Accountant is logged in the system.
2. Accountant goes to the Suppliers Page.
3. Accountant will be displayed the suppliers.
4. Accountant selects the suppliers he wants to order from and checks the products he wants and the quantities.
5. After finishing marking the needed products, he submits the request.
6. Accountant is redirected to home page again.

**Scenario 17 – Accountant generates a PDF, listing the ordered food products for a supplier**.

1. Accountant is logged in the system.
2. Accountant goes to the Suppliers Page.
3. Accountant will be displayed the suppliers.
4. If the accountant has already marked the food products for a supplier, a “Generate Bill” option will appear.
5. Accountant clicks on this button and the PDF will generated.
6. Accountant saves it to his computer, in order to send it later to the supplier.

**Scenario 18 – Chef claims an order.**

1. Chef is logged in the system.
2. Chef is displayed a table of orders and he has to choose one of them.
3. If he wants to take any of the orders, he accepts the order.

**Scenario 19 – Chef confirms an order.**

1. Chef is logged in the system.
2. Chef is displayed a table of orders.
3. If he has already claimed an order and has finished cooking, he can confirm the order.

**Scenario 20 – Chef adds a new dish on the menu.**

1. Chef of the software is logged in the system.
2. Chef goes to the Menu Page.
3. Chef adds a new entry on the menu.
4. Chef will be displayed a form, with the entry information needed to be provided.
5. Chef fills in the information, such as the name, price, ingredients and picture.
6. Chef saves the new dish by confirming it.

**Scenario 21 – Chef makes requests about kitchen materials that got broken or went missing.**

1. Chef of the software is logged in the system.
2. Chef goes to the Requests Page.
3. Chef writes in a text field what he needs in the kitchen and then sends it to the accountant.

**Scenario 22 – Waiter assigns the status to a table.**

1. Waiter is logged in the system.
2. Tables are displayed as tabs and the waiter can choose the status for each of them. (available/busy)

**Scenario 23 – Waiter confirms payment.**

1. Waiter is logged in the system.
2. As soon as the client pays, the waiter can confirm it.

**Scenario 24 – Waiter prints the bill.**

1. Waiter is logged in the system.
2. As soon as the client pays, after payment confirmation by the waiter, he also prints the generated bill.
   * 1. **Use Cases**

|  |  |
| --- | --- |
| **Name** | Login |
| **Summary** | The user enters the system by typing correct username and password |
| **Actor** | Admin, Chef, Waiter, Accountant, Others |
| **Description** | To enter the system, the user must type the correct username and correct password |
| **Precondition** | Every user must have an existing account |
| **Alternatives** | There are no alternative options |
| **Post Condition** | Each one to enter his/her page |

|  |  |
| --- | --- |
| **Name** | Admin adds a new worker |
| **Summary** | The admin can add a new worker by entering the workers information, a photo of the worker and the category |
| **Actor** | Admin |
| **Description** | All fields of the form must be filled so a new worker to be added |
| **Precondition** | Admin must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | An alert notifies if the worker had been added and this employee can enter the system immediately |

|  |  |
| --- | --- |
| **Name** | Admin see tables |
| **Summary** | The admin can see the tables of the whole restaurant. Can see which of these are available or not and also open the table page. |
| **Actor** | Admin |
| **Description** | Every table is identified by its own table number and has a password viewable only from the admin. |
| **Precondition** | Admin must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | Here it can be seen the number of tables and their availability |

|  |  |
| --- | --- |
| **Name** | Admin edit the worker page |
| **Summary** | Admin can open the worker’s profile and can add the password, phone number or category |
| **Actor** | Admin |
| **Description** | Admin can click on the edit button next to the password, phone number or category and make the changes |
| **Precondition** | Admin must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | An alert notifies the admin about the change and the employees profile has been changed. |

|  |  |
| --- | --- |
| **Name** | Admin deletes the worker |
| **Summary** | Admin deletes the worker from the database |
| **Actor** | Admin |
| **Description** | When a profile page is opened you can delete the worker by pressing the ‘delete’ button |
| **Precondition** | Admin must be logged in and must have opened profile page of the worker |
| **Alternatives** | There are no alternative options |
| **Post Condition** | Admin is redirected to his page |

|  |  |
| --- | --- |
| **Name** | Admin adds a new table |
| **Summary** | The admin can add a new table by its entering its table number, table password and the number of people |
| **Actor** | Admin |
| **Description** | All fields of the form must be filled so a new table to be added |
| **Precondition** | Admin must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | An alert notifies the admin that a new table is available and the table can be accessed immediately |

|  |  |
| --- | --- |
| **Name** | Admin retrieves and edits information of a table |
| **Summary** | The admin can change the table password and can change the table number of people.  Also admin can see all orders received from that table. There are two buttons which reserve or delete the table |
| **Actor** | Admin |
| **Description** | To edit the table credentials edit button must be clicked. By clicking ‘Orders’ button, the list of orders are shown below. Here the table can be reserved and deleted |
| **Precondition** | Admin must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | An alert notifies the admin about the changes done |

|  |  |
| --- | --- |
| **Name** | Admin adds a new item in the menu |
| **Summary** | After the item name is types, then there can be added new products and their quantities. There is also an option to add a picture of the item |
| **Actor** | Admin |
| **Description** | A new item must have at least two products |
| **Precondition** | Admin must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | An alert notifies the admin that a new item is added on the menu and that item can be accessed from the customer immediately |

|  |  |
| --- | --- |
| **Name** | Accountant retrieves/adds/deletes the suppliers from the supplier list |
| **Summary** | The accountant can see the list of suppliers, add or delete the supplier |
| **Actor** | Accountant |
| **Description** | There is a table filled with the credentials of the supplier |
| **Precondition** | Accountant must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | An alert notifies the accountant for the changes |

|  |  |
| --- | --- |
| **Name** | Accountant sends email to the supplier specifying products |
| **Summary** | The accountant can send an email to the supplier to order the product the restaurant wants |
| **Actor** | Accountant |
| **Description** | Communication with the supplier is done with emails. Accountant sends emails to the suppliers. |
| **Precondition** | Accountant must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | Accountant is notified for the changes |

|  |  |
| --- | --- |
| **Name** | Accountant checks the notifications received |
| **Summary** | The accountant checks the notifications and marks them as read |
| **Actor** | Accountant |
| **Description** | Communication within staff is done with `notifications`. Accountant can see the messages directed to his username. |
| **Precondition** | Accountant must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | Accountant marks as `read` the read messages |

|  |  |
| --- | --- |
| **Name** | Accountant creates transaction |
| **Summary** | Enters info related to amount, status and description |
| **Actor** | Accountant |
| **Description** | Accountant can create transactions and store them in the database about the purchases. |
| **Precondition** | Accountant must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | No post condition |

|  |  |
| --- | --- |
| **Name** | Accountant searches for a transaction and generates the report in PDF |
| **Summary** | Accountant can search for a transaction and then can generate a pdf format for this transaction |
| **Actor** | Accountant |
| **Description** | Accountant can view the transactions and then can generate a pdf format of this transaction that can be printed |
| **Precondition** | Accountant must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | Pdf format can be printed |

|  |  |
| --- | --- |
| **Name** | Accountant/chef/waiter edit their profile/personal information. |
| **Summary** | When these actors enter their profiles, there can edit their personal information |
| **Actor** | Accountant, Chef, Waiter |
| **Description** | Actors can change their only their password, phone number |
| **Precondition** | These actors must be logged on |
| **Alternatives** | There are no alternative options |
| **Post Condition** | An alert notifies about the changes |

|  |  |
| --- | --- |
| **Name** | Accountant inserts staff salaries. |
| **Summary** | Accountant is responsible to edit and add the salaries for the staff |
| **Actor** | Accountant |
| **Description** | On the main page of the accountant there will be the list of all the salaries where only accountant can edit them |
| **Precondition** | Accountant must be logged on |
| **Alternatives** | There are no alternative options |
| **Post Condition** | An alert notifies about salaries |

|  |  |
| --- | --- |
| **Name** | Accountant modifies product information |
| **Summary** | Accountant can edit the information about the product |
| **Actor** | Accountant |
| **Description** | On the main page of the accountant there will be the list of all the products where only accountant can edit the quantity or the price |
| **Precondition** | Accountant must be logged on |
| **Alternatives** | There are no alternative options |
| **Post Condition** | An alert notifies about products |

|  |  |
| --- | --- |
| **Name** | Chef retrieves all the unprocessed orders from the tables |
| **Summary** | The chef can see the list of orders available |
| **Actor** | Chef |
| **Description** | The chef can see the orders in table. Each row has the table number, a list of orders from the menu, a description and a button `Accept`. If one admin accepts one order, it means others cannot see it anymore. |
| **Precondition** | Chef must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | When the order is accepted, a new page opens where the chef selects the products which are in the dishes of the order. These products are subtracted from the database. |

|  |  |
| --- | --- |
| **Name** | Chef retrieves all the processed orders by him |
| **Summary** | After the chef has accepted the orders, he can see all the orders which are being cooked by him. |
| **Actor** | Chef |
| **Description** | The chef by clicking on the `processing orders` button can generate the table of all orders which are being cooked by him. Each row contains id of the order, the list of dishes, description and a button `Done`. There he can press `Done` which means the order is cooked. |
| **Precondition** | Chef must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | When the order is cooked, the waiter can see the order the order. |

|  |  |
| --- | --- |
| **Name** | Chef sends notification |
| **Summary** | Chef can send notification to the other staff members |
| **Actor** | Chef |
| **Description** | The chef can press the button `create notification` and a pop up window will open. In this pop up window the chef can write the username of the staff he wants to send the message and the message he wants to send. (For example: chef wants to send the notification to the accountant for a new utensil) |
| **Precondition** | Chef must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | A new notification record is stored in the database. |

|  |  |
| --- | --- |
| **Name** | Waiter manages the processed orders |
| **Summary** | Waiter is responsible for the cooked orders |
| **Actor** | Waiter |
| **Description** | The waiter can see the finished orders by the chef and accept them. If one waiter accepts the order, the order cannot be seen by other waiters online. The waiters also is responsible for the payment confirmation and printing the receipt |
| **Precondition** | Waiter must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | When the payment confirmation is done, a new transaction is added on the database |

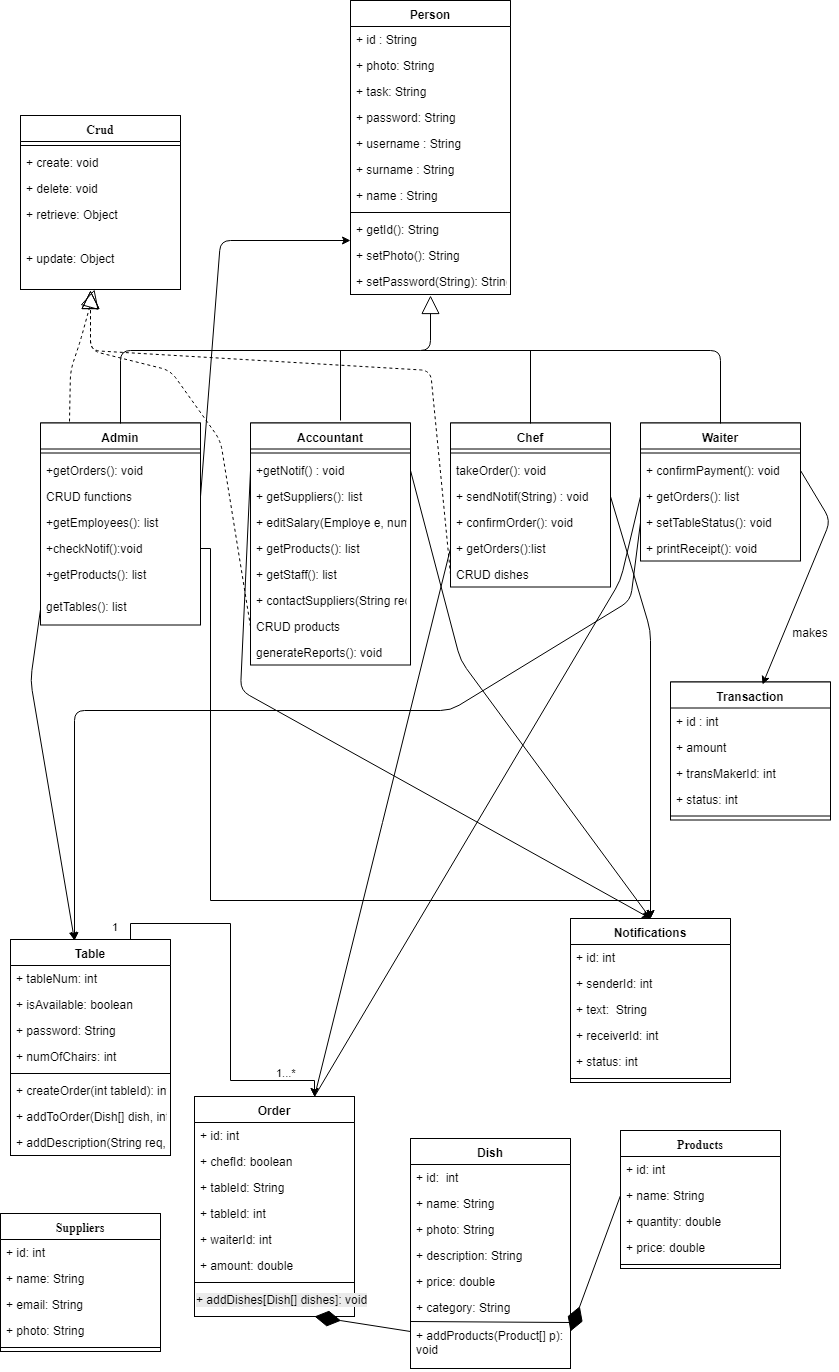
|  |  |
| --- | --- |
| **Name** | Waiter manages the tables of the restaurant |
| **Summary** | Waiter sets the tables as busy or available |
| **Actor** | Waiter |
| **Description** | Waiter can see the state of all the tables in the restaurant. He can change the states of the tables as busy or available. |
| **Precondition** | Waiter must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | No post condition |

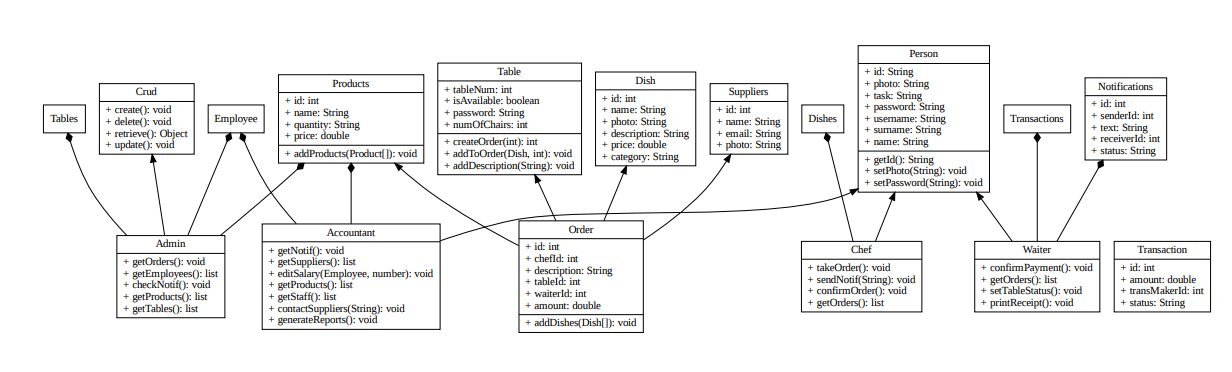
|  |  |
| --- | --- |
| **Name** | Table login |
| **Summary** | Each table can login from the login page. To log in the table number and the password must be written correctly |
| **Actor** | Table |
| **Description** | Each table is unique and there must be entered the right credentials so the customers can order the items |
| **Precondition** | Each table must have been created before from the admin |
| **Alternatives** | There are no alternative options |
| **Post Condition** | When logged in, the menu is displayed and the customer can select the order |

|  |  |
| --- | --- |
| **Name** | Order making by table |
| **Summary** | Here the customer can select the items from the categories drinks, dishes, pasta, pizza, deserts.  When all selected the customer can confirm the order by pressing the confirm button |
| **Actor** | Customer |
| **Description** | There is a list of whole orders available in the restaurant. Also the customer can add a description in the order |
| **Precondition** | Table must be logged in |
| **Alternatives** | There are no alternative options |
| **Post Condition** | The confirmed order can now be seen from the chef. When the chef accepts the order, the customer gets a notification. |

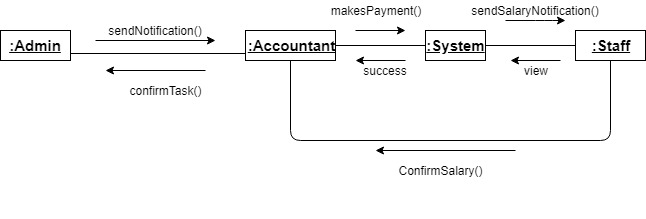
**UML DIAGRAMS**

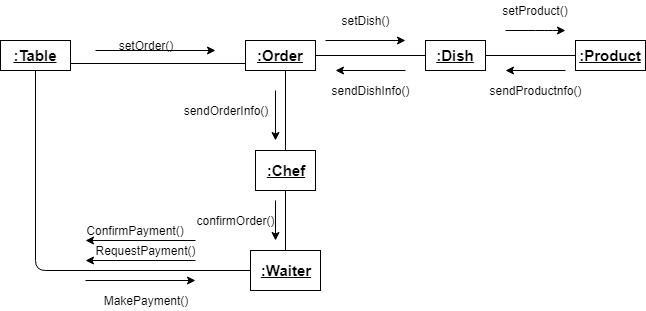
**Class Diagram**

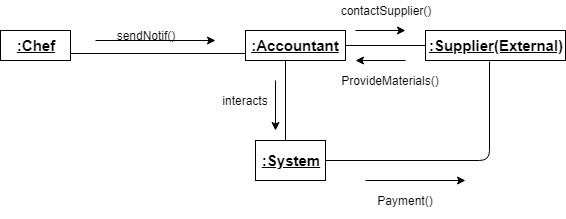




# Collaboration Diagrams

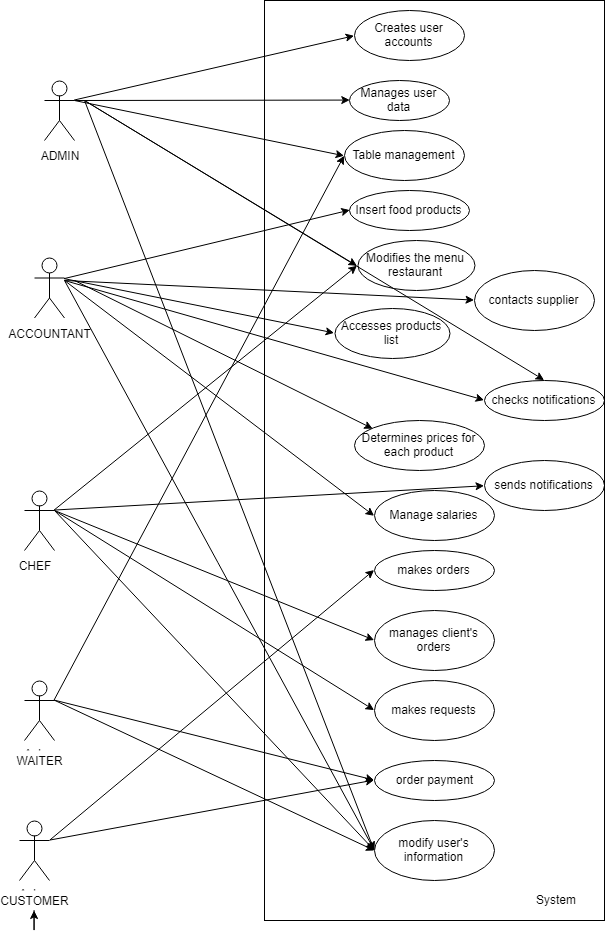


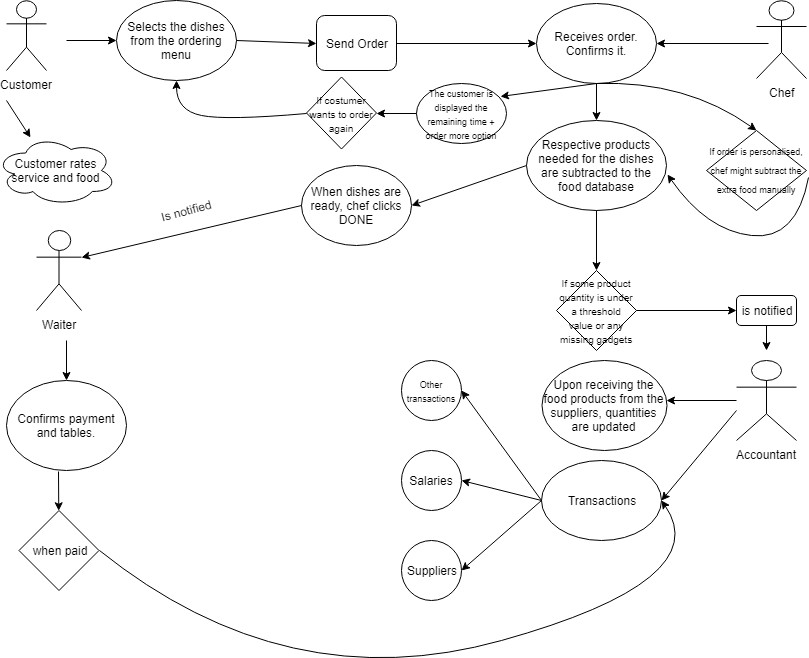


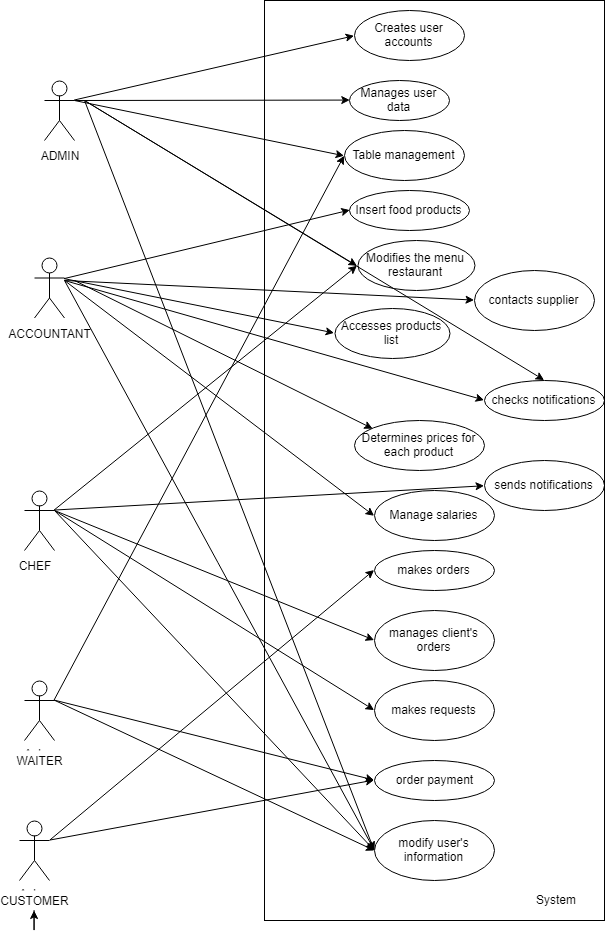


**USE-CASE DIAGRAM**

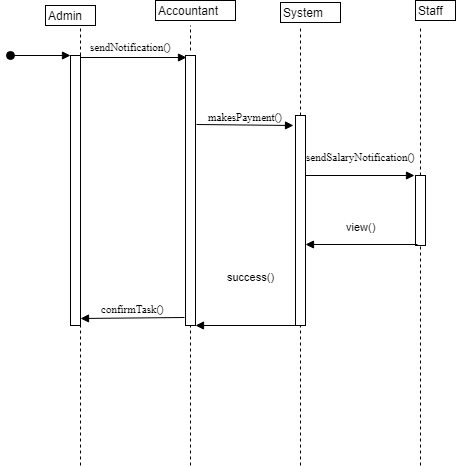
**WORKFLOW**

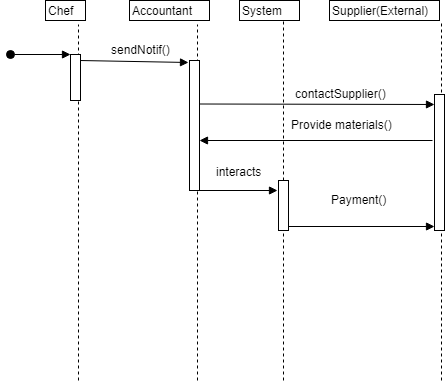
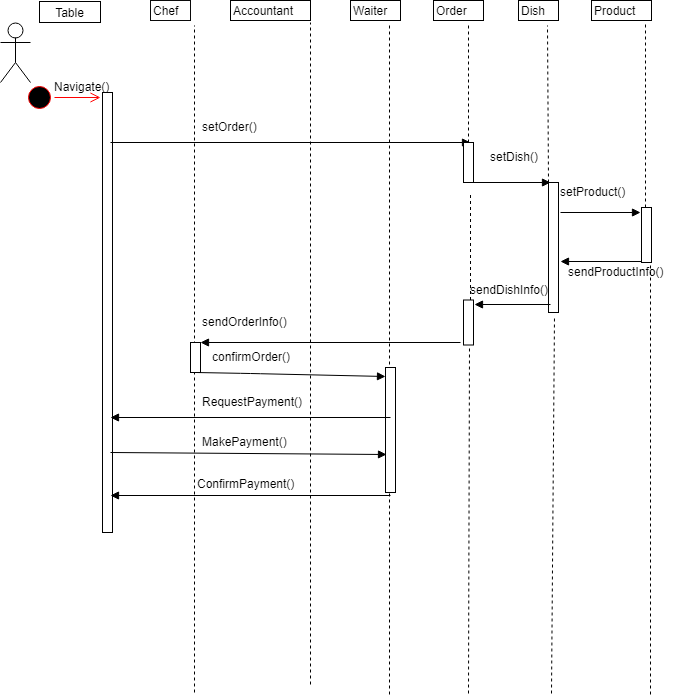
**





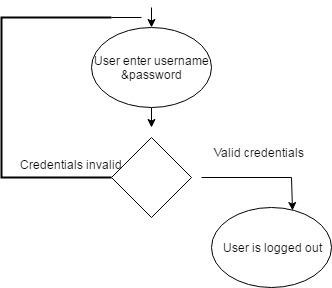
**Sequence Diagram**



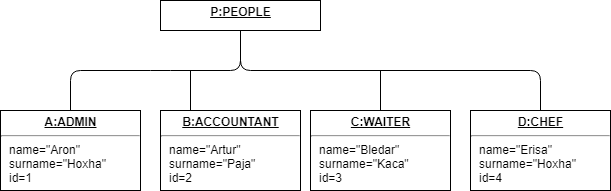


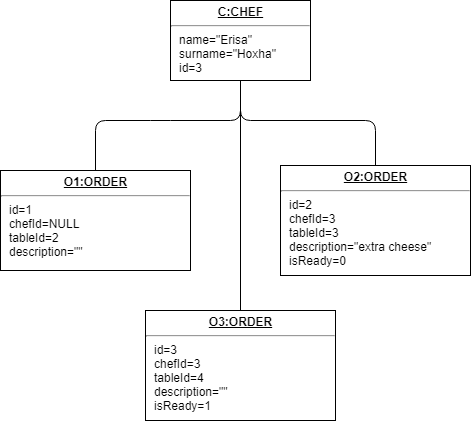
**Activity diagrams**

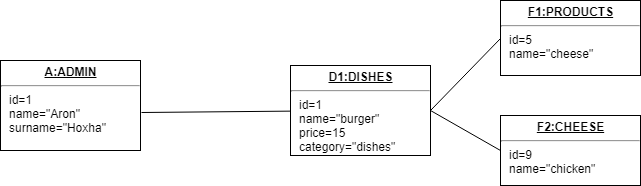
## Scenario 1-2

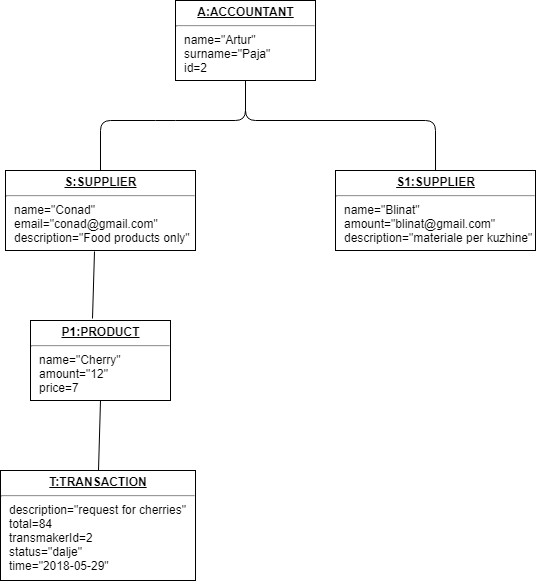


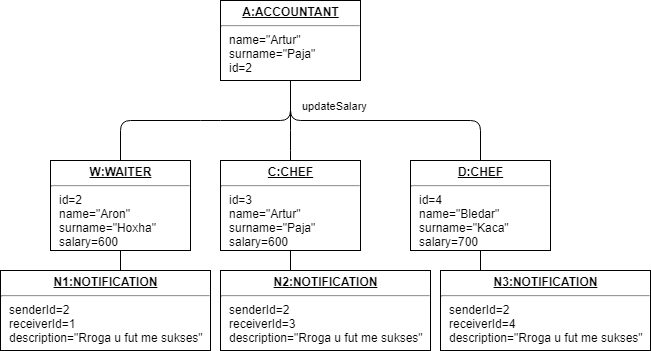
# Object Diagrams

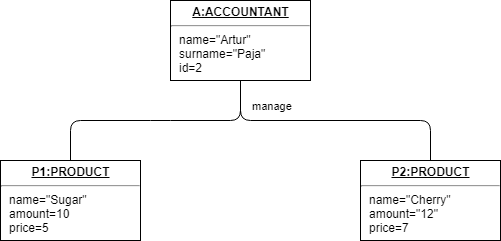


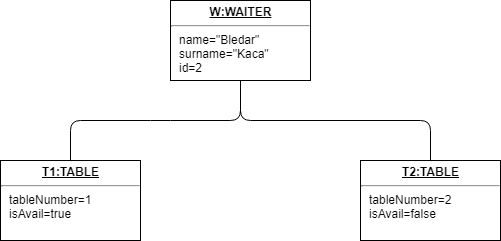


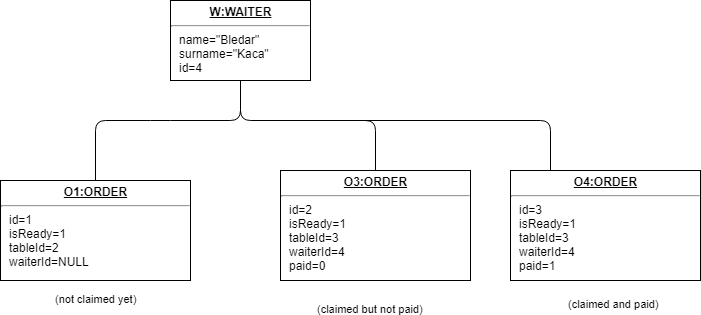


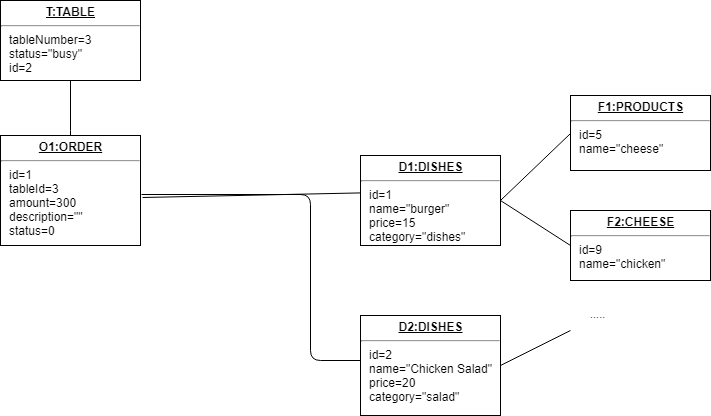












# Test Cases (Unit Tests) for Phase 2

**Preconditions:**

* User is logged in as staff.
* The inventory is pre-loaded with 50 units of "CheeseBurger" and 30 units of "Veggie Delight."

**Test Steps:**

1. Navigate to the order management interface.
2. Place an order for 5 "CheeseBurger" and 3 "Veggie Delight."
3. Confirm the order and complete the transaction.

**Expected Result:**

* The inventory for "CheeseBurger" should show 45 units.
* The inventory for "Veggie Delight" should show 27 units.

**Actual Result:**

* Inventory updated correctly: "CheeseBurger" shows 45 units and "Veggie Delight" shows 27 units.

**Status (Pass/Fail):** Pass

**Comments:** The inventory system updates promptly and accurately upon order confirmation.

Test Case ID: DS0022

**Title:** Profile Update with Email Change

**Objective:** To ensure the user profile is updated with a new email address and the system reflects changes immediately.

**Preconditions:**

* User has an existing profile.
* Test email "[old-email@example.com](mailto:old-email@example.com)" is changed to "[new-email@example.com](mailto:new-email@example.com)."

**Test Steps:**

1. Log in to the user account with current credentials.
2. Go to profile settings.
3. Update the email to "[new-email@example.com](mailto:new-email@example.com)" and submit the change.
4. Log out and attempt to log back in with the new email address.

**Expected Result:**

* User profile shows the updated email address.
* Login with old email address fails.
* Login with new email address succeeds.

**Actual Result:**

* Email update is successful. Login with "[old-email@example.com](mailto:old-email@example.com)" fails, and login with "[new-email@example.com](mailto:new-email@example.com)" succeeds.

**Status (Pass/Fail):** Pass

**Comments:** Email change functionality works as expected. The user is forced to log out and re-authenticate with the new email.

Test Case ID: DS0023

**Title:** Error Handling for Out-of-Stock Items during Ordering

**Objective:** To test if the system correctly handles orders for items that are out of stock.

**Preconditions:**

* User is logged into the system.
* Inventory level for "Grilled Chicken Sandwich" is 0.

**Test Steps:**

1. Navigate to the order page.
2. Attempt to add 1 "Grilled Chicken Sandwich" to the order.
3. Observe the system's response and error message.

**Expected Result:**

* The system prevents adding the item to the order.
* The system displays an out-of-stock message for "Grilled Chicken Sandwich."

**Actual Result:**

* The system allowed the addition of "Grilled Chicken Sandwich" to the cart but displayed an out-of-stock message at checkout.

**Status (Pass/Fail):** Fail

**Comments:** The error was handled at the final step of the ordering process rather than at the initial addition to the cart. Requires front-end validation to prevent out-of-stock items from being added to the cart.

Test Case ID: DS0024

**Title:** Distributor Order Placement Functionality

**Objective:** To verify that orders to distributors are placed successfully when inventory levels fall below the reorder threshold.

**Preconditions:**

* "Tomato" inventory level is at the reorder threshold of 10 units.
* Distributor details are correctly set up in the system.

**Test Steps:**

1. Simulate the sale that brings "Tomato" below the reorder threshold.
2. Check the system for an auto-generated order to the distributor for "Tomato."
3. Confirm that the order details are correct and include the correct quantities based on the pre-set reorder amount.

**Expected Result:**

* An automatic order for "Tomato" is generated.
* The order contains the correct quantities and distributor information.

**Actual Result:**

* An order for "Tomato" was generated but with incorrect quantities—50 units instead of the expected 20 units based on reorder settings.

**Status (Pass/Fail):** Fail

**Comments:** While the automatic order was generated, there appears to be an issue with the quantity calculation logic. This will require a fix and a re-test.

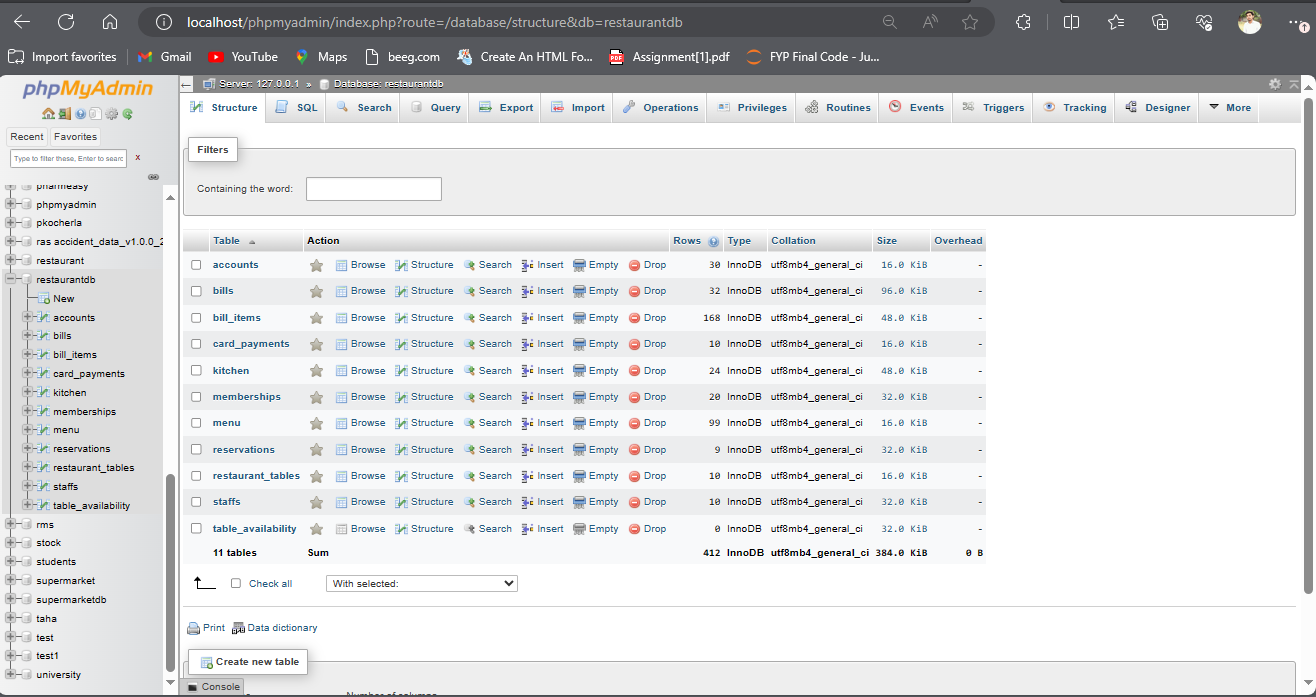
# Introduction to the User Manual

This user manual is intended to guide both new and existing users through the process of navigating and using our PHP-based system. From installation to daily tasks, this guide aims to ensure a smooth experience for all users.

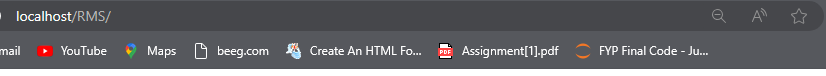
## **Installation Guide:**

Download Project: Start by downloading the PHP project zip from the official repository.

* Extract Project: Once downloaded, extract the project files into your local server environment. For instance, if you're using XAMPP, extract it to the htdocs directory.
* Database Setup: Launch your server's control panel (e.g., XAMPP Control Panel) and start the Apache and MySQL services. Then, open your web browser and navigate to PHPMyAdmin (usually localhost/phpmyadmin). Here, you will import the provided SQL file to establish the required database and tables.



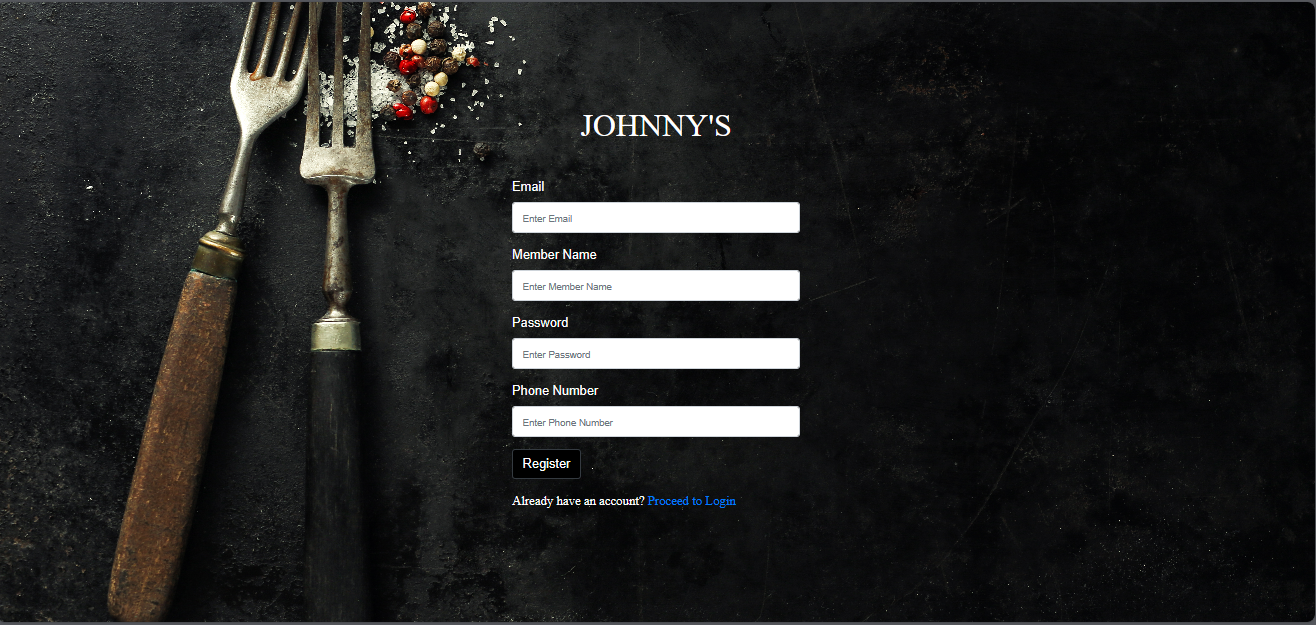
* Accessing the Application: With the database set up, navigate to localhost/your-project-folder-name on your web browser. This is where you can start interacting with the application.



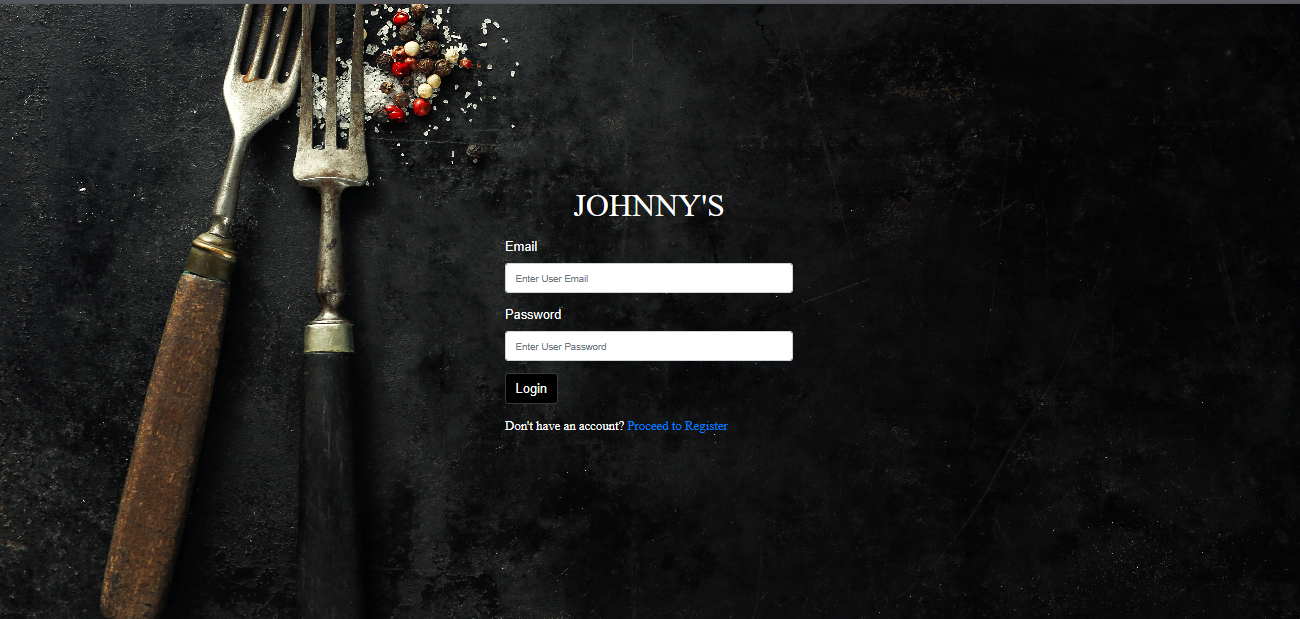
## **User Interface Guide:**

### Registration and Login:

* Registration: If you're accessing the system for the first time, you'll need to register. Click on the 'Register' link or button on the homepage. This will take you to the registration form. Here, you will provide your name, a valid email address (which will also serve as your username), and a password. Once completed, click 'Submit' to create your account.



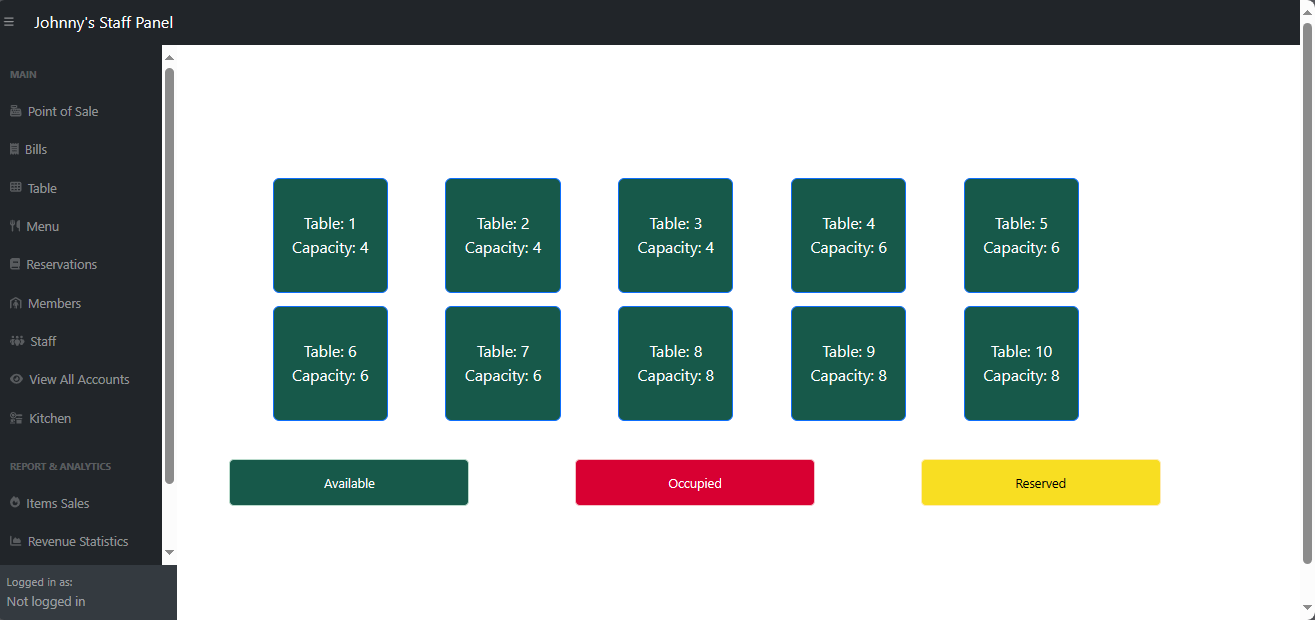
* Login: If you're a returning user, click on the 'Login' link or button on the homepage. Enter your registered email address and password. If the credentials are correct, you will be taken to your dashboard.



### 

### Dashboard and Profile Management:

* Dashboard Overview: After logging in, you'll be taken to your personal dashboard. This page provides an overview of recent activities, user statistics, and other pertinent information. You can navigate to other sections of the system using the navigation bar or menu.
* Profile Management: To update your profile details, navigate to the 'Profile' or 'Account Settings' section. Here, you can modify your name, email address, password, and other personal details. Always remember to save any changes you make.



### 

### Troubleshooting:

* Forgotten Password: If you forget your password, click on the 'Forgot Password' link on the login page. Enter your registered email address, and you will receive instructions on how to reset your password.
* Data Not Loading: If certain data isn't displaying, ensure your server services (Apache and MySQL) are running. If the problem persists, contact your system administrator.

# Compilation Instructions

**Prerequisites:**

* PHP 7.4 or higher installed.
* MySQL Server running.
* Composer dependency manager installed.

**Steps:**

1. **Clone the Repository:** Clone the project repository from GitHub to your local machine.
2. **Install Dependencies:** Navigate to the project directory in your terminal and run **composer install** to install all necessary dependencies.
3. **Database Configuration:** Import the provided SQL file into your MySQL server to set up the required database tables.
4. **Edit Configuration Files:** In the **.env** file, update the database connection details to match your MySQL server configuration.
5. **Start the Server:** Navigate to the project's root directory in your terminal and run **php -S localhost: 8000** to start the PHP server.
6. **Access the Application:** Open your web browser and go to **http://localhost:8000** to access the application.

# Report code inspection Feedback:

**Scope of Review:**

* **Code Quality:** Reviewing the readability, maintainability, and adherence to standard coding practices.
* **Functionality:** Verification of the new features implemented in phase 2, including advanced inventory management, user profile enhancement, and distributor module integration.
* **Security:** Examination of any potential security vulnerabilities introduced or existing in the new code.
* **Testing:** Evaluation of the updated test cases to ensure comprehensive coverage and effectiveness in finding bugs.

**Review Findings:**

1. **Code Clarity and Organization:**
   * The code is generally well-organized with a clear modular structure.
   * There are sections where commenting could be more descriptive to explain complex logic.
2. **Functionality Checks:**
   * All new functionalities introduced in phase 2 are present and operational.
   * The distributor module integrates well with inventory management, although there were minor issues with updating the inventory in real-time.
3. **Security Assessment:**
   * The new user profile enhancement features respect privacy and are secured against SQL injection attacks.
   * Some API endpoints used for the distributor interactions need better access control checks to prevent unauthorized use.
4. **Testing Adequacy:**
   * Test cases developed for phase 2 cover the primary user paths and error conditions.
   * Suggestion for additional test cases to cover unexpected user input and stress conditions.

**Recommendations:**

* **Code Documentation:** Improve inline documentation, especially around the new APIs introduced for distributor management.
* **Real-time Updates:** Address the lag in inventory updates post distributor order fulfillment.
* **Security Measures:** Implement rigorous access controls and authentication checks for all new API endpoints.
* **Test Expansion:** Develop additional test cases as recommended to ensure the robustness of the system under edge cases.

# Reflection

Reflecting on Deliverable 4 of the DineSys project, our team has made commendable progress in bringing the vision for a seamless restaurant management experience closer to reality. The introduction of advanced inventory management has been a highlight, significantly improving the way staff interact with stock data.

**Successes:**

* **Feature Implementation:** We successfully delivered the advanced inventory system, which was a critical component of this phase.
* **User Experience:** The enhancements to the user profiles and dashboard received positive feedback for improving the user journey within the system.
* **Collaboration:** Our team collaboration has been robust, with each member contributing effectively to the project's advancement.

**Challenges and Areas for Improvement:**

* **Real-time Inventory Update:** We observed a slight delay in inventory updates during high-volume transactions, which could lead to potential discrepancies.
* **Security Rigor:** The introduction of new features has increased our attack surface, necessitating a stronger emphasis on security, particularly for API endpoints.
* **Testing Coverage:** While our unit tests covered the primary functionalities, we recognized the need for a broader range of tests, including stress testing and performance testing.

**Future Focus:**

* **Performance Tuning:** As we continue to add features, we must keep an eye on the system's performance, ensuring it scales well and remains responsive.
* **Comprehensive Security Strategy:** We plan to incorporate a security audit in our next phase to proactively identify and remediate vulnerabilities.
* **Feedback Loop:** Strengthening our feedback loop by incorporating more frequent user testing sessions will help us remain aligned with user needs and expectations.

**Concluding Thoughts:**

The progress we've made is substantial, yet there's always room for growth. The reflections from this phase are not merely critiques but stepping stones for future development. The team is motivated to address the highlighted issues and improve upon our current successes for the subsequent deliverable.

## **Improvements for Future Phases**

In the aftermath of Deliverable 3, as we took a moment to ponder on our journey, several key insights stood out that could pave the way for improvements in our subsequent phases. While we achieved a substantial amount of our set objectives, we also identified areas where our approach could be refined to enhance efficiency and output quality.

### 1. Communication:

Even though our team maintained a consistent communication flow, there were instances where certain decisions took longer due to awaiting feedback or validations. We need to adopt a more synchronous mode of communication, especially during crucial decision-making junctures.

### 2. Documentation:

While our documentation met the requirements, it became evident that we could have organized and structured it more effectively. A clearer distinction between different sections, utilization of tables, graphs, and visuals, and ensuring that all technical jargon is adequately explained can be areas of improvement.

### 3. Task Prioritization:

Although we implemented an incremental approach, there were scenarios where non-critical tasks consumed significant time. Going forward, we need to reassess our task prioritization strategies and ensure that critical tasks are always at the forefront.

### 4. User Interface and Experience:

From the feedback received, there are certain elements in our user interface that users found to be not very intuitive. We will be focusing on collecting more user feedback and refining the UI/UX to ensure a seamless experience for the end-users.

### 5. Testing:

While our test cases covered the core functionalities, there were edge cases that we missed. An expansion of our test suite and the integration of continuous testing tools can provide a more robust testing framework.

### 6. Peer Review Feedback Incorporation:

While we took into account most of the feedback from our peer reviews, a more structured approach to categorizing and systematically addressing this feedback would accelerate our iteration process.

### 7. Resource Allocation:

There were instances where certain team members were overloaded while others had bandwidth. A more dynamic approach to resource allocation, based on individual strengths, weaknesses, and current workloads, could enhance our team's efficiency.

### 8. Training:

Certain challenges faced during this phase were due to a lack of expertise in specific tools or technologies. Investing time in short training or knowledge-sharing sessions can mitigate such challenges in the future.

In conclusion, while we're proud of our accomplishments in Deliverable 3, the insights gained from this phase are invaluable. They've provided a clear roadmap for areas of improvement. Embracing these lessons will not only streamline our future endeavors but also enhance the quality of our deliverables. As a team, we are committed to continuous learning and improvement, and we believe that these reflections will be the catalyst for our enhanced performance in the upcoming phases

# Team Member Contribution Table

|  |  |  |  |
| --- | --- | --- | --- |
| Member name | Contribution description | Overall Contribution (%) | Note |
| Vangala Ruchitha | Team Leadership, Client Interactions | 15 |  |
| Nikhila Polkampally | Backend Development, Database Management | 15 |  |
| Nandhini Kasukurthi | Frontend Development, UI/UX Design | 10 |  |
| Vishal Rachuri | Testing, Quality Assurance | 10 |  |
| Sai Srinivas Valleti | System Analysis, Documentation | 10 |  |
| Chandralekha Gude | Project Coordination, Feedback Analysis | 10 |  |
| Aishwarya Karukonda | HR and Staff Training Coordination | 10 |  |
| Hunny Keshwani | Backend Development | 10 |  |
| Leela Varadatta Sai Addanki | Frontend Development, UI/UX Assistance | 10 |  |