

# Wait-Less Product Scenario

**Group 2: Ammar Idrees, Zaid Patel, Bryan Gutierrez, Maaz Ahmed**

The application that we are creating is to create an efficient program for restaurant owners to implement to further optimize delivery and waiting time for customers. Furthermore, it will allow waiters to be able to serve customers with ease without having to memorize orders, which customers need assistance, and many other tasks that may need attention. The application will feature the layout of a restaurant, with different tables and customers present, routing the most efficient pathway for the waiters, and ensuring all customers are taken care of on a first-come basis, it would also allow waiters to be able to visualize the restaurant to effectively locate customers when needed. The application will be in use for the owners and waiters as they are able to view how many tables are full, which ones have been recently served, and those still in need of service. In terms of this project, we will not have to download any data files to have the layout working, rather we will be using any kind of JS framework for the front end while utilizing Java and another application to complete the MVC model. Furthermore, SQL is what we will be utilizing for the database to be able to store information on each customer (arrival, order, when they are served, and when they leave), as it should also be compatible with MVC.

## **Scenario "Restaurant Experience"**

Once the application is started, the user will be prompted to sign in with their given username and password. What would happen next would depend on whether a waiter or manager logs in. If a waiter logs in, they will be able to view all the tables within the restaurant, and all tasks that require attention. On the other hand, if a manager logs in, they would be able to have an overview screen which would allow the manager to see the statistics of the restaurant and be able to do the same things a waiter can. Statistics that can appear can include average wait times and some other things.

After they have logged in, they will immediately be reminded to check for customers that have arrived and they will be the number one priority because they would be the only customers in the restaurant. Furthermore, employees can ping each other to remind them of certain tasks. If the tasks aren't done in a timely manner, they would go up in urgency for an employee to see.

Once a customer has arrived, the user will select a table based on what tables are open on the app and then set that table as closed in order to avoid this table being selected twice. The user must select the number of people in the table, whether or not there are elderly customers. The application will then automatically set a timer for this table to give them time to decide what to order and will then remind the user to check up on them shortly. Furthermore, there would be an estimated time for each table in case all tables are occupied. That way, a waiter would be able to inform waiting customers what the expected wait time would be.

When there are multiple customers in the restaurant, the application and user will make sure to prioritize the customers that have been waiting for the longest time based on the timer set for each table. When a table has not been visited in some specified time frame, the application will remind the user to check on their table. When the user has finished checking up on that certain table, the timer will then reset.

Often times, customers will need something from their waiter or waitress before the timer has run out(he could be asking for a drink, placing another order, etc.) so the user will override the timer that was placed on that table and reset it counting this as a visit to the table.

The interface would be simple and clean. When the user logs in, it would display to the user all of the tables within the restaurant and a menu that would display tasks that need to be done such as checking up on a table or any pings that may have been sent by other employees. The interface should have a clean and simple design to ensure proper performance as well as make it easy to use. If a manager is logged into the system, additional menu items will be visible such as statistics.

We can also implement the interface using icons so it's visually appealing and makes it easier for the user to know what to click. When you are working in a fast-paced environment, it can be difficult to read words so implementing icons would make it more efficient and ensure that mistakes are being limited.

### **Diagram of the basic functionality for wait-less**

