



Bilkent University

Department of Computer Engineering

Senior Design Project

Group No: 2332

Project Name: Cyclops

Detailed Design Report

21902348, Kaan Kurçer, kaan.kurcer@ug.bilkent.edu.tr

21902903, Osman Serhat Yılmaz, serhat.yilmaz@ug.bilkent.edu.tr

21903213, Ali Doğaç Urkaya, dogac.urbaya@ug.bilkent.edu.tr

21902358, Özgür Abi, ozgur.abi@ug.bilkent.edu.tr

21902035, Jankat Berslan Dinçer, berslan.dincer@ug.bilkent.edu.tr

Supervisor: Shervin Rahimzadeh Arashloo

Course Instructors: Erhan Dolak, Tağmaç Topal

10.03.2023

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS492.

Table of Contents

Table of Contents	1
1. Introduction	2
1.1. Purpose of the System	2
1.2. Design Goals	2
1.2.1. Usability	2
1.2.2. Performance	2
1.2.3. Security	2
1.2.4. Scalability	2
1.2.5. Reliability	3
1.2.6. Maintainability	3
1.3. Overview	3
2. Current Software Architecture	3
3. Proposed Software Architecture	4
3.1. Overview	4
3.2. Subsystem Decomposition	4
3.3. Persistent Data Management	5
3.4. Access Control and Security	5
4. Subsystem Services	7
4.1. Monitoring System	7
4.2. Server	7
4.2.1. Controller Layer	7
4.2.2. Logic Layer	9
4.2.3. Data Layer	10
4.3. Client	11
4.3.1. View Layer	11
4.3.2. Controller Layer	12
4.3.3. Model Layer	12
5. Test Cases	13
6. Consideration of Various Factors in Engineering Design	31
7. Teamwork Details	31
7.1. Contributing and functioning effectively on the team	31
7.2. Helping creating a collaborative and inclusive environment	31
7.3. Taking lead role and sharing leadership on the team	31
8. References	32

1. Introduction

Restaurants are visited by hundreds of customers every day. Different types of customers have different spending habits, visiting times, and preferences. The traffic of the restaurant also varies greatly for different hours of the day. Although restaurant owners already record the inside of their restaurants for security reasons, they are unable to collect and utilize the data produced by their customers. The information gained from monitoring the customers' count and preferences could be used to enhance both the user experience of the visitors and the profits of the restaurant owners. Cyclops aims to analyze the security feed of the restaurant to collect this information and provide it to the restaurant owner along with tips on how to improve their restaurant using said information.

1.1. Purpose of the System

Our system aims to keep track of customers in the restaurant while also providing detailed reports and statistics about customers. Therefore, the purpose of the system is to accurately track the number of customers while analyzing them at the same time.

The application will be able to track customer info of the restaurant and give precise and detailed information to restaurant owners. Also, since our system is meant to be used by restaurant owners or managers and not by engineers, UI will be easy to manage and navigate.

Another aim of this application is to safely keep the customer data and protect their privacy.

1.2. Design Goals

1.2.1. Usability

- System should be easily understandable by the restaurant staff
- For the customers it should be easy to view restaurant info

1.2.2. Performance

- System should be able to keep track of customers pretty quickly
- Stored information should be accessible without much lag

1.2.3. Security

- Data about customers should not be easily accessible
- Each restaurant's info should only be accessible by that restaurant

1.2.4. Scalability

- The app should work with decent size restaurants
- Program should also be able to work with large number of customer info

1.2.5. Reliability

- The program should track customers with few or no errors

1.2.6. Maintainability

- Program and its components should be easy to keep track of and modify

1.3. Overview

Cyclops is a web/mobile application that aims to help restaurants by tracking and analyzing their customers. Cyclops aims to both provide the customer with information on restaurants seat availability and the restaurant with data about customers. While there already exist some apps that provide information about restaurant availability, these applications only provide limited availability data and not actual numbers, our application will be much more precise. For the restaurants, the app will provide them with useful insights on their customers. For this application computer vision, developed with python, will be utilized with cameras and it will be the base of our customer tracking and analysis systems. Our application will also be able to work even with crowded restaurants. Our program will be accessible with desktop and Android. Desktop version will be used by the restaurant. Clients will use the mobile version.

2. Current Software Architecture

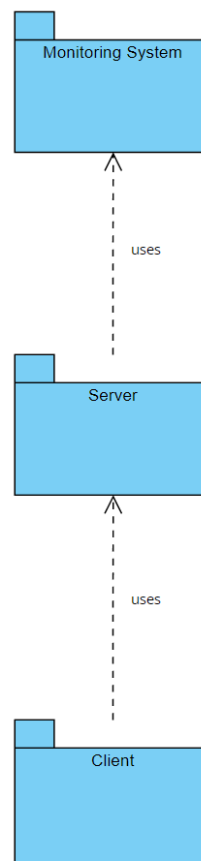
Our main target audience for our application is larger restaurants with many repeating customers that we can recognize and provide personalized experiences for. The restaurants we plan on serving are also places where people stay for a long time on a given visit. We have investigated similar services and found that the majority of customer counting systems aim to analyze queue lengths and waiting times at a given hour for fast food restaurants¹. They aim to scan the ordering areas and the waiting areas to determine how long a customer would wait before receiving their order at a certain time. There are also systems for a restaurant to investigate their customer profiles and manage their restaurants accordingly, but none of them incorporate software and image recognition². A widely used software system that profiles its customers and keeps track of their visitation and ordering habits does not currently exist. However, articles exist on how the restaurant industry can be revolutionized using AI³, and this is a field that is currently in development with a preexisting demand.

3. Proposed Software Architecture

3.1. Overview

In order to achieve an efficient/secure system architecture, interactions and authorizations inside the Cyclops system needs to be defined. To achieve this, decomposition of each subsystem that makes up the Cyclops system should be analyzed. In the following sections the decomposition of each subsystem, how data management is handled and the access control and security protocols will be examined.

3.2. Subsystem Decomposition



The subsystem decomposition of the Cyclops system can be seen on the diagram above. Our system is made up of 3 subsystems: the Client which the customers will be interacting with, the Server which will handle all the processing

and the Monitoring System which will provide information about the customers to the Server via the cameras which will be set up in the restaurant.

The monitoring subsystem uses computer vision and machine learning algorithms to gain information from security cameras placed in the restaurant. It works in real-time and records the customer count and profile from the camera footage, then sends this information to the server to be displayed on the web application.

The server subsystem will store and distribute the data it collects from the monitoring subsystem to the client subsystem. It will be responsible for allowing various different user types to access data they are privileged to view. All data processing will also be done in the server subsystem. The server will receive both customer profile information and customer count information from the monitoring system, and determine favorite foods and visiting habits of profiled customers to be displayed to the restaurant staff.

The client subsystem will be utilized by three groups of people, managers, staff, and customers. All three types of users will connect to the server using the web application. The data on restaurant capacity present in the will be presented to the customers. Managers and staff will also be able to access data on customer profiles in order to provide the aforementioned personalized experiences. Managers will be able to view data on their restaurant's metrics, such as occupancy at various days and hours, and customer profiles.

3.3. Persistent Data Management

We will use MySQL as our database, it is a cloud-native database system. We will use MySQL because it is free to use, has a nice user interface in its desktop application and also most of our group members know it. Also it is simple to use with Spring with the jdbc library. Created entities will be persisted in MySQL databases as tables. Repository interfaces which are inside the Repository Subsystem will be able to manipulate the data in the database, performing operations such as save, delete, get, update etc. These interfaces will extend a parent interface JpaRepository, this interface is imported by Spring Data JPA (Java Persistence API) and helps us to perform these operations without writing manual queries for simple query operations. With true naming of a function in this Repository interface, queries will be automatically created by methods and implemented when needed.

3.4. Access Control and Security

Cyclops will have a security system implemented by using Spring Security library, this library will be used for various security processes. One of them is encrypting user passwords when saving them to the database so any hacker with a key of the password will not be able to know the passwords real value because it will be hashed. Also to decrease the predictability of key values of passwords in the database, id's of every entity will be created using UUID (Universally Unique ID), this id is impossible to be guessed by hackers so it protects the data in the database by securing the primary key of the data. Another security process is controlling the access of users that have different roles. For example only restaurant managers and waiters will be able to see detailed information about customers of the restaurant, whereas the customers will only see the density of restaurant and restaurant information etc. This will be implemented by using roles in Spring Security, these roles will be used in html pages of the program to specify which roles can see specific actions on the web page. Here are the list of the roles of the Cyclops and the functionalities that these roles can do:

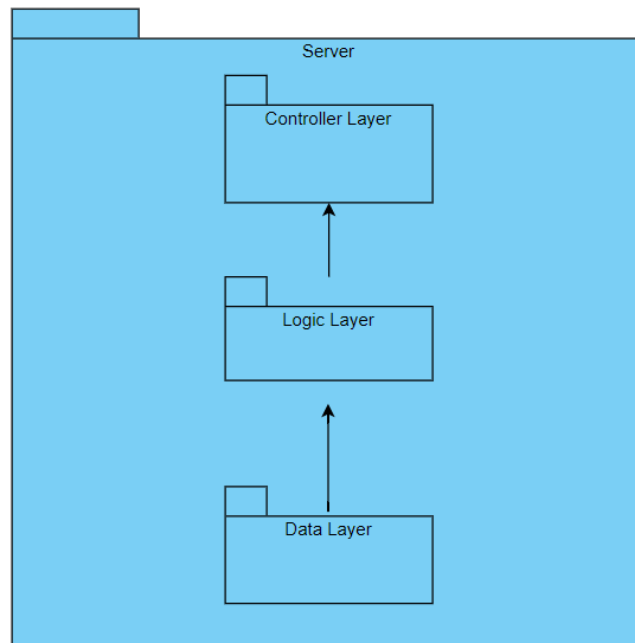
- Restaurant Manager
 1. Login operation
 2. See amount of customers in restaurant
 3. See restaurant density on a given day
 4. See a specific customer's information
- Restaurant Staff
 1. Login operation
 2. See entered customer information momentarily
 3. See a specific customer's information
- Customer
 1. See the density of the restaurant for that moment
 2. See restaurant information

4. Subsystem Services

4.1. Monitoring System

The monitoring system consists of multiple parts. We will utilize security cameras around the restaurant to count and profile customers. The cameras will be recording the restaurant from multiple angles and counting the number of customers at each table. This data will then be sent to the servers to be processed. We will also be connected to the restaurant's own payment systems to receive billing data and customer information if present, to enhance our customers' profiles with favorite orders and visitation information to provide personalized experiences.

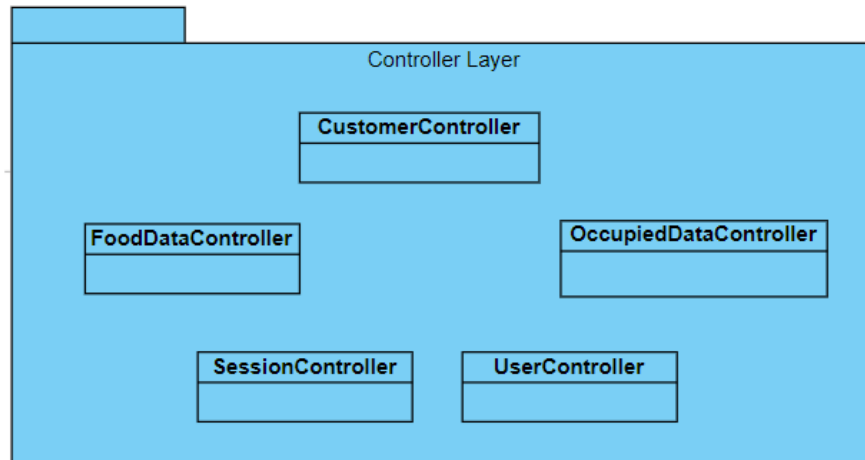
4.2. Server



The server subsystem of the Cyclops consists of three layers as shown above, controller layer, logic layer and data layer.

4.2.1. Controller Layer

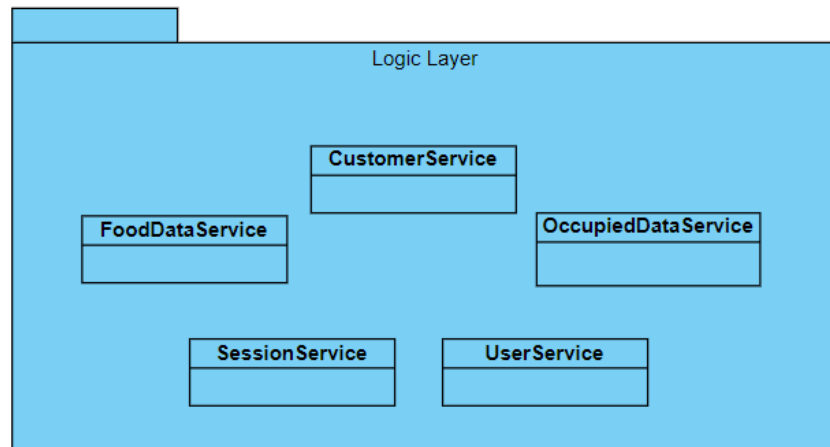
Controller layer of the server subsystem will handle the HTTP requests to exchange information with the client subsystem.



Class	Description
CustomerController	Handles the HTTP requests related to Customer entity.
FoodDataController	Handles the HTTP requests related to FoodData entity.
OccupiedDataController	Handles the HTTP requests related to OccupiedData entity.
SessionController	Handles the HTTP requests related to Session entity.
UserController	Handles the HTTP requests related to User entity.

4.2.2. Logic Layer

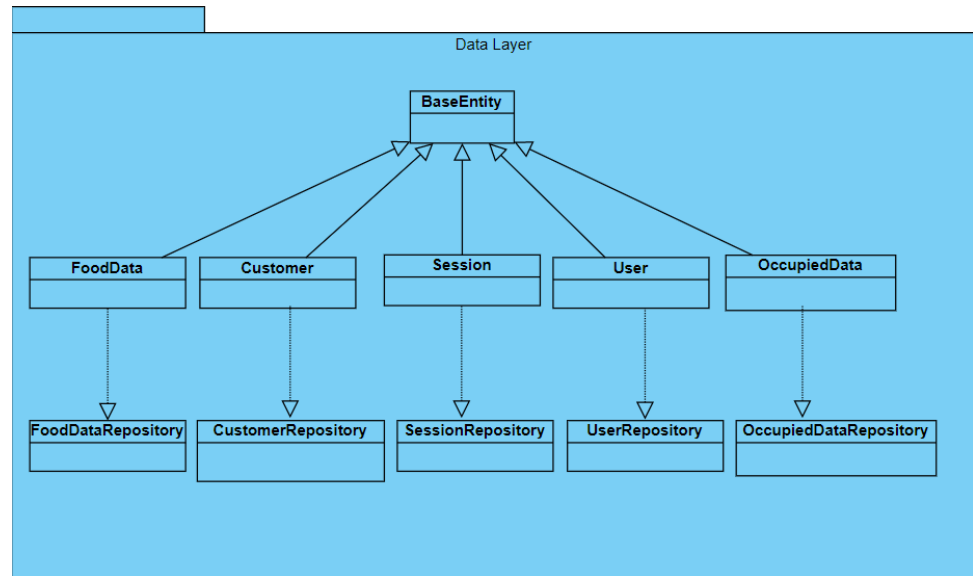
Logic layer of the server subsystem will handle the business logic within the server.



Class	Description
CustomerService	Handles the business logic of Customer entity.
FoodDataService	Handles the business logic of FoodData entity.
OccupiedDataService	Handles the business logic of OccupiedData entity.
SessionService	Handles the business logic of Session entity.
UserService	Handles the business logic of User entity.

4.2.3. Data Layer

Data layer of the server subsystem will handle the database operations such as save, delete, update etc. and persistence of the data.

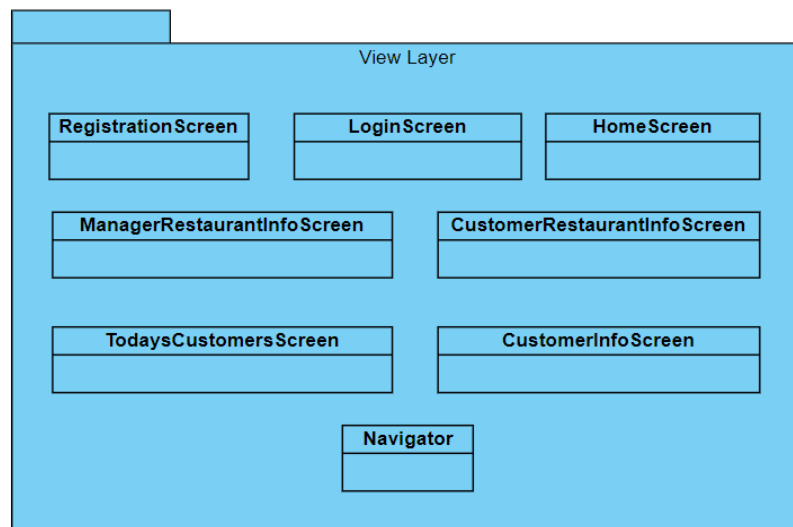


Class	Description
BaseEntity	Parent class for the entity objects in Data Layer.
FoodData	This class provides a model for the FoodData entity in the database.
OccupiedData	This class provides a model for the OccupiedData entity in the database.
Customer	This class provides a model for the Customer entity in the database.
Session	This class provides a model for the Session entity in the database.
User	This class provides a model for the User entity in the database.
<<interface>> FoodDataRepository	Handles the database operations for the FoodData entity, managing customers' orders, determines favorite foods.
<<interface>> OccupiedDataRepository	Handles the database operations for the OccupiedData entity, managing customer count and restaurant occupancy at any given time, charts the number of customers at various hours and days.

<<interface>> CustomerRepository	Handles database operations for the Customer entity.
<<interface>> SessionRepository	Handles database operations for the Session entity, keeping the time any given customer spends at the restaurant.
<<interface>> UserRepository	Handles database operations for the User entity.

4.3. Client

4.3.1. View Layer

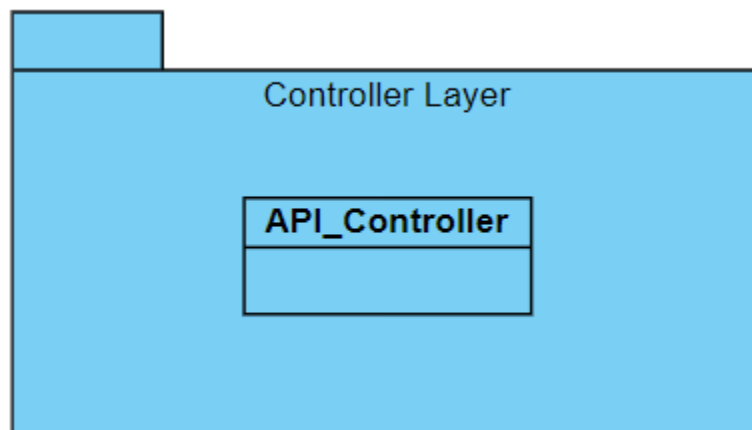


View Layer is the subsystem which contains the UI elements which the user will interact with.

Class	Description
RegistrationScreen	This class provides the registration components to the user.
LoginScreen	This class provides the login components to the user.
HomeScreen	This class provides the home screen components to the user.
ManagerRestaurantInfoScreen	This class provides the specified restaurant info screen components to the user with the manager role.

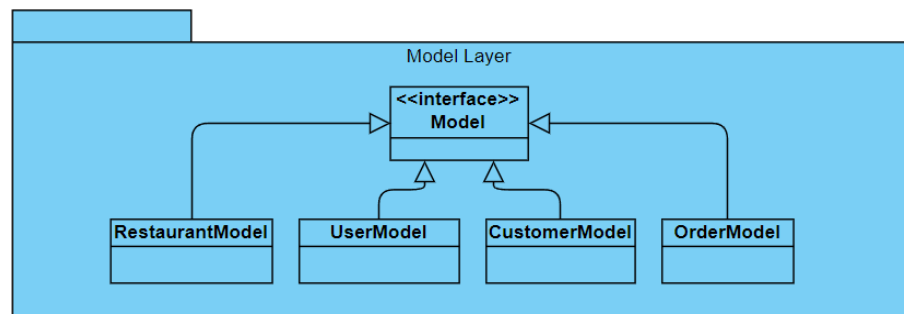
CustomerRestaurantInfoScreen	This class provides the restaurant info screen components to the user with the customer role.
TodaysCustomersScreen	This class provides the today's customers info screen components to the user with the manager role.
CustomerInfoScreen	This class provides specific customer info screen components to the user with the manager role.
Navigator	This class handles the navigation between different screens of the app.

4.3.2. Controller Layer



Controller Layer is the subsystem that contains the API Controller class which will handle the HTTP requests and the server connection of UI elements.

4.3.3. Model Layer



Model Layer is the subsystem which contains the data elements that are needed for the client side.

Class	Description
<<interface>> Model	This interface provides the get and set methods.
RestaurantModel ~Model	This class provides a model for the Restaurant object.
UserModel ~Model	This class provides a model for the User object.
CustomerModel ~Model	This class provides a model for the Customer object.
OrderModel ~Model	This class provides a model for the Order object.

5. Test Cases

Test ID	1
Test Type/Category	Functional, component, usability.
Summary/Title/Objective	Check if the login and sign up buttons correctly redirect the user to their respective pages.
Procedure of testing steps	Try the buttons on multiple platforms and verify that they successfully redirect users to the correct pages.
Expected results/Outcome	The login and sign up buttons correctly redirect the user to their respective pages.
Priority/Severity	Critical

Test ID	2
Test Type/Category	Security, component.
Summary/Title/Objective	Check if the password is hidden while entering.
Procedure of testing steps	Try entering the password in the login screen.
Expected results/Outcome	The password is hidden while entering, and the letters are replaced by stars (*).
Priority/Severity	Major

Test ID	3
Test Type/Category	Security, installation, compatibility.
Summary/Title/Objective	Make sure the username does not contain invalid characters.
Procedure of testing steps	Try entering invalid characters into the username box while creating an account.
Expected results/Outcome	The user is blocked from choosing a username that contains invalid characters.
Priority/Severity	Critical

Test ID	4
Test Type/Category	Security, installation.
Summary/Title/Objective	Make sure the user cannot have a password that does not fit predefined criteria.
Procedure of testing steps	Try entering a password that does not fit predefined criteria into the password box while creating an account.
Expected results/Outcome	The user is blocked from choosing a password that does not fit predefined criteria.
Priority/Severity	Critical

Test ID	5
Test Type/Category	Usability, component.
Summary/Title/Objective	Check if closed restaurants are listed after open restaurants in the drop down list.
Procedure of testing steps	Check if the restaurants are listed correctly.
Expected results/Outcome	The restaurants that are currently closed show up after the restaurants that are currently open in the drop down list.
Priority/Severity	Minor

Test ID	6
Test Type/Category	Usability, non-functional, component.
Summary/Title/Objective	Check if restaurants are displayed with the correct images.
Procedure of testing steps	Check different restaurants from the drop down list and see if their images are correct.
Expected results/Outcome	Each restaurant displays its logo or image next to its name in the drop down list.
Priority/Severity	Minor

Test ID	7
Test Type/Category	Usability, functional, component.
Summary/Title/Objective	Check if the drop down list disappears and the user is correctly redirected to the chosen restaurant when a restaurant is selected from the list.
Procedure of testing steps	Check if the user is correctly redirected to the chosen restaurant.
Expected results/Outcome	The user is taken to the page of the restaurant of their choice.
Priority/Severity	Critical

Test ID	8
Test Type/Category	Component, functional, usability, compatibility.
Summary/Title/Objective	Check if the drop down list functions correctly at different window sizes.
Procedure of testing steps	Resize the window and check if the drop down list works.
Expected results/Outcome	The drop down list works at different window shapes and sizes.
Priority/Severity	Minor

Test ID	9
Test Type/Category	Component, functional, usability..
Summary/Title/Objective	Check if the forgot password text is clickable.
Procedure of testing steps	Try clicking the forgot password text.
Expected results/Outcome	The forgot password text is clickable and redirects users to the password reset page.
Priority/Severity	Major

Test ID	10
Test Type/Category	Component, non-functional, security.
Summary/Title/Objective	Make sure an account cannot be created without having pressed the "I agree to the Terms of Use and Privacy Policy" button.
Procedure of testing steps	Try and create an account without pressing the button.
Expected results/Outcome	Users that try to create accounts without pressing the aforementioned button are denied.
Priority/Severity	Major

Test ID	11
Test Type/Category	Component, functional
Summary/Title/Objective	Check if the correct restaurant is displayed when the customer selects a restaurant.
Procedure of testing steps	Try to choose a restaurant from the restaurant list.
Expected results/Outcome	The correct restaurant is displayed when the customer selects a restaurant.
Priority/Severity	Major

Test ID	12
Test Type/Category	Component, functional
Summary/Title/Objective	Check if the restaurant list is correctly displayed to the customer.
Procedure of testing steps	Navigate to the restaurant list screen to see the restaurant list.
Expected results/Outcome	The restaurant list is correctly displayed to the customer.
Priority/Severity	Major

Test ID	13
Test Type/Category	Component, Functional
Summary/Title/Objective	Check if the current customer count is correctly displayed in the restaurant info page after choosing a restaurant.
Procedure of testing steps	Choose a restaurant and check if the current customer count matches with the data given to the backend via the monitoring system.
Expected results/Outcome	The current customer count is correctly displayed in the restaurant info page after choosing a restaurant.
Priority/Severity	Major

Test ID	14
Test Type/Category	Component, Functional
Summary/Title/Objective	Check if the maximum capacity of a restaurant is correctly displayed in the restaurant info page after choosing a restaurant.
Procedure of testing steps	Choose a restaurant and check if the maximum capacity is displayed correctly in reference to the data in the database.
Expected results/Outcome	The maximum capacity of a restaurant is correctly displayed in the restaurant info page after choosing a restaurant.
Priority/Severity	Minor

Test ID	15
Test Type/Category	Component, Functional
Summary/Title/Objective	Check if the date selection menu under the Restaurant Density tab is displaying the dates correctly.
Procedure of testing steps	Click on the date selection menu and check if the dates are correctly ordered and displayed.
Expected results/Outcome	The date selection menu under the Restaurant Density tab is displaying the dates correctly.
Priority/Severity	Major

Test ID	16
Test Type/Category	Component, Functional
Summary/Title/Objective	Check if the correct hour intervals are displayed under the Restaurant Density tab.
Procedure of testing steps	Select a day from the date selection menu and check if the bar chart displays the hour intervals correctly.
Expected results/Outcome	The correct hour intervals are displayed under the Restaurant Density tab.
Priority/Severity	Major

Test ID	17
Test Type/Category	Component, Functional
Summary/Title/Objective	Check if the date selection menu under the restaurant density tab allows the user to change the year, month and day.
Procedure of testing steps	Try to change the year, month and day in the date selection menu.
Expected results/Outcome	The date selection menu under the restaurant density tab allows the user to change the year, month and day.
Priority/Severity	Major

Test ID	18
Test Type/Category	Component, Non-functional
Summary/Title/Objective	Check if the bar charts are distinct enough from each other for different densities.
Procedure of testing steps	In the restaurant density tab check if the size of the chart is big enough to distinguish the bars from each other.
Expected results/Outcome	The bar charts are distinct enough from each other for different densities.
Priority/Severity	Minor

Test ID	19
Test Type/Category	Component, Integration, Functional
Summary/Title/Objective	Check if changing the date changes the bar chart density representation.
Procedure of testing steps	Change the date from the date selection tab and check if the bar chart changes accordingly.
Expected results/Outcome	Changing the date changes the bar chart density representation.
Priority/Severity	Major

Test ID	20
Test Type/Category	Component, functional
Summary/Title/Objective	Make sure the user can only enter a date (in the form of DD/MM/YYYY) in the date selection tab.
Procedure of testing steps	Try to enter text into the date selection tab to check if it allows text into the input field.
Expected results/Outcome	The user can only enter a date (in the form of DD/MM/YYYY) in the date selection tab.
Priority/Severity	Major

Test ID	21
Test Type/Category	Functional, Integration
Summary/Title/Objective	Check if the prediction of human count in restaurant is correct
Procedure of testing steps	Compare the predicted customers by human counting system and compare it with the real results.
Expected results/Outcome	Error rate is expected to be below %10
Priority/Severity	Critical

Test ID	22
Test Type/Category	Integration, Usability
Summary/Title/Objective	Check if the prediction of the human counting system is displayed correctly on the UI.
Procedure of testing steps	Compare the value in the database with the value in the UI.
Expected results/Outcome	Both values should be the same.
Priority/Severity	Major

Test ID	23
Test Type/Category	Integration
Summary/Title/Objective	Check if the past human counts is saved
Procedure of testing steps	Compare the prediction of human counting system for a past time footage with the value saved to the database for that time.
Expected results/Outcome	Both values should be the same.
Priority/Severity	Major

Test ID	24
Test Type/Category	Integration, Compatibility
Summary/Title/Objective	Check if the graph of density of restaurant is correct
Procedure of testing steps	Compare the values displayed in the graph with the real values saved in database.
Expected results/Outcome	Values should be the same.
Priority/Severity	Major

Test ID	25
Test Type/Category	Integration, Compatibility
Summary/Title/Objective	Check if the favorite meal information of customer is correct
Procedure of testing steps	Compare the most ordered food by customer and favorite meal of customer in the database.
Expected results/Outcome	Both values should be the same
Priority/Severity	Major

Test ID	26
Test Type/Category	Security
Summary/Title/Objective	Check if logout is made successfully
Procedure of testing steps	Login to see if session started and then logout and check the session information to see whether it is finished or not.
Expected results/Outcome	Session should be finished after logout is made.
Priority/Severity	Critical

Test ID	27
Test Type/Category	Integration, Functional
Summary/Title/Objective	Check if the table numbers are unique for every table
Procedure of testing steps	Check the database to see if there are 2 tables that have the same id.
Expected results/Outcome	Every table should have different table number.
Priority/Severity	Major

Test ID	28
Test Type/Category	Integration, Functional
Summary/Title/Objective	Check if the same table is not shown occupied at the same time by different customers.
Procedure of testing steps	Check the session time for tables and see if 2 different sessions intersect.
Expected results/Outcome	Sessions should not intersect for the same table.
Priority/Severity	Major

Test ID	29
Test Type/Category	Usability, Functional
Summary/Title/Objective	Check if users can see empty tables at that moment correctly.
Procedure of testing steps	Compare the state of tables at the UI with the real state of tables.
Expected results/Outcome	Both values of empty tables should be the same.
Priority/Severity	Major

Test ID	30
Test Type/Category	Functional
Summary/Title/Objective	Check if the session duration of the customer is predicted correctly.
Procedure of testing steps	Compare the value recorded by human counting system with the real duration of session of customer.
Expected results/Outcome	Values should not have more difference than 5 minutes.
Priority/Severity	Major

Test ID	31
Test Type/Category	Functional
Summary/Title/Objective	Check if the transition between restaurant information and customer information works correctly
Procedure of testing steps	Click the buttons to check if transition happens
Expected results/Outcome	The app should be able to switch from one tab to the other
Priority/Severity	Major

Test ID	32
Test Type/Category	Functional
Summary/Title/Objective	Check if today's customers menu displays all the customers
Procedure of testing steps	Check the todays customers menu and compare it to actual customers in a location
Expected results/Outcome	Menu lists all the customers
Priority/Severity	Major

Test ID	33
Test Type/Category	Functional
Summary/Title/Objective	Check if the pictures of today's customers are listed correctly
Procedure of testing steps	Look at the displayed pictures and compare them to real pictures
Expected results/Outcome	Pictures are displayed correctly
Priority/Severity	Major

Test ID	34
Test Type/Category	Functional
Summary/Title/Objective	Check if registered customers are recognized properly
Procedure of testing steps	See if the system can recognize a registered customer
Expected results/Outcome	System properly recognizes registered customers
Priority/Severity	Major

Test ID	35
Test Type/Category	Functional
Summary/Title/Objective	Check if entry time of customers are tracked correctly
Procedure of testing steps	Check in real life if the customers that enter are tracked properly
Expected results/Outcome	Customer entry times are correct
Priority/Severity	Major

Test ID	36
Test Type/Category	Functional
Summary/Title/Objective	Check if departure time of customers are tracked correctly
Procedure of testing steps	Check in real life if the customers that leave are tracked properly
Expected results/Outcome	Customer departure times are correct
Priority/Severity	Minor

Test ID	37
Test Type/Category	Functional
Summary/Title/Objective	Check if the table no is properly listed
Procedure of testing steps	Check if the listed table no and real table no are matching
Expected results/Outcome	Table no is tracked correctly
Priority/Severity	Major

Test ID	38
Test Type/Category	Functional
Summary/Title/Objective	Check if the customer no is correct
Procedure of testing steps	Check to see if when a customer enters their no is properly assigned
Expected results/Outcome	Customer no is correctly tracked
Priority/Severity	Minor

Test ID	39
Test Type/Category	Functional
Summary/Title/Objective	Check if customer visits are correct
Procedure of testing steps	Check if a customer is recorded when they visit
Expected results/Outcome	Customer visits are listed correctly
Priority/Severity	Major

Test ID	40
Test Type/Category	Functional
Summary/Title/Objective	Check if customer purchases are correct
Procedure of testing steps	Check if purchases can be entered and stored correctly
Expected results/Outcome	Purchases are valid
Priority/Severity	Major

Test ID	41
Test Type/Category	Functional
Summary/Title/Objective	Check if favorite meals are correct
Procedure of testing steps	Check the prior purchases and compare them to the favorite meal
Expected results/Outcome	Favorite meal is listed correctly
Priority/Severity	Minor

Test ID	42
Test Type/Category	Functional
Summary/Title/Objective	Check if current customer count of the table is displayed correctly
Procedure of testing steps	Check the record in the database and compare it with the UI
Expected results/Outcome	Current customer count of the table is displayed correctly
Priority/Severity	Major

Test ID	43
Test Type/Category	Functional
Summary/Title/Objective	Check if customer count of the table at the selected time interval is displayed correctly
Procedure of testing steps	Check the record in the database and compare it with the UI
Expected results/Outcome	Customer count of the table at the selected time interval is displayed correctly
Priority/Severity	Minor

Test ID	44
Test Type/Category	Functional
Summary/Title/Objective	Check if all tables are listed on the restaurant density page
Procedure of testing steps	Check the record in the database and compare it with the UI
Expected results/Outcome	All tables are listed on the restaurant density page
Priority/Severity	Major

Test ID	45
Test Type/Category	Functional
Summary/Title/Objective	Check if customers at the table are identified correctly
Procedure of testing steps	Check the records in the database and compare it with the UI
Expected results/Outcome	Customers at the table are identified correctly
Priority/Severity	Minor

Test ID	46
Test Type/Category	Functional
Summary/Title/Objective	Check if the most preferred time interval for the table is displayed correctly
Procedure of testing steps	Check the record in the database and compare it with the UI
Expected results/Outcome	The most preferred time interval for the table is displayed correctly
Priority/Severity	Minor

Test ID	47
Test Type/Category	Functional
Summary/Title/Objective	Check if there are no duplicate records in the customer count of tables database
Procedure of testing steps	Check the record in the database and compare it with the UI
Expected results/Outcome	There are no duplicate records in the customer count of tables database
Priority/Severity	Major

Test ID	48
Test Type/Category	Functional
Summary/Title/Objective	Check if the most preferred tables of the restaurant are displayed in correct order
Procedure of testing steps	Check the record in the database and compare it with the UI
Expected results/Outcome	The most preferred tables of the restaurant are displayed in correct order
Priority/Severity	Minor

Test ID	49
Test Type/Category	Functional
Summary/Title/Objective	Check if average customer count of the table is displayed correctly
Procedure of testing steps	Check the record in the database and compare it with the UI
Expected results/Outcome	Average customer count of the table is displayed correctly
Priority/Severity	Minor

Test ID	50
Test Type/Category	Functional
Summary/Title/Objective	Check if average customer count of the restaurant is displayed correctly
Procedure of testing steps	Check the record in the database and compare it with the UI
Expected results/Outcome	Average customer count of the restaurant is displayed correctly
Priority/Severity	Minor

Test ID	51
Test Type/Category	Functional
Summary/Title/Objective	Check if there are no unnecessary records in the customer count of tables database
Procedure of testing steps	Check the record in the database and compare it with the UI
Expected results/Outcome	There are no unnecessary records in the customer count of tables database
Priority/Severity	Major

6. Consideration of Various Factors in Engineering Design

We took a comprehensive approach to engineering design in our software project, taking into consideration various factors that were crucial to its success. Firstly, we prioritized the user experience and ensured that our software was user-friendly, intuitive and accessible to both restaurant owners and customers. Additionally, we placed a strong emphasis on the software's reliability and scalability, ensuring that it could handle large amounts of data and traffic and work with different sizes of restaurants. Security was also a major concern, and we implemented robust security measures to protect sensitive user data and prevent unauthorized access. Finally, we ensured that our software was compatible with a wide range of devices and operating systems, allowing for maximum flexibility and accessibility for our users.

7. Teamwork Details

7.1. Contributing and functioning effectively on the team

We were to satisfy the requirements for contributing and functioning effectively on the team by prioritizing communication and teamwork. We made sure to communicate effectively with each other, actively listening to each other's ideas and feedback. We also worked together closely to achieve our common goals, and we were always accountable and responsible for our own tasks and deadlines. In addition, we were always willing to step up and help out when a team member needed assistance.

7.2. Helping creating a collaborative and inclusive environment

To help create a collaborative and inclusive environment, we actively sought out diverse perspectives and ideas, and encouraged each other to share our thoughts and experiences. We fostered open communication and feedback, creating a space where everyone felt valued and supported. We were always respectful and considerate of each other's unique backgrounds, experiences, and perspectives.

7.3. Taking lead role and sharing leadership on the team

In taking a lead role and sharing leadership on the team, we set clear goals and expectations for the team and communicated them effectively. We led by example, being accountable and responsible for our own tasks, demonstrating a strong work ethic and commitment to the project. Additionally, we shared leadership on the team, empowering and encouraging each other to take on leadership roles as well, and actively seeking out each other's input and feedback. By doing so, we were able to create a more collaborative and inclusive environment and ensure that everyone felt valued and supported.

8. References

[1]:<https://www.footfallcam.com/en/Industries/FastFoodRestaurant/Overview>

[2]:<https://www.therail.media/stories/2022/4/11/a-restaurant-owners-guide-to-customer-profiles-with-examples>

[3]:<https://www.forbes.com/sites/forbestechcouncil/2022/09/22/10-ways-ai-is-transforming-the-restaurant-industry/?sh=5a801e1b6ec8>