



POLITECNICO DI MILANO

Master's Degree in Automation and Control Engineering

PROJECT DOCUMENTATION

CLup: Customers Line-up
For Software Engineering - (2020-2021)

Guided by: Prof. Matteo Rossi

The Project presented by:
Mullaiventhan Anna durai 883406
Shahrukh Ahmed Irfani 874143

Table of Contents

List of Figures	3
1. Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Process Requirements	4
1.4 Environmental Constraints	4
1.5 Project Restrictions	4
1.5 Project License	4
2. Project Description	5
2.1 Software Requirements	6
2.1.1 Functional Requirements	6
2.1.2 Process Requirements	7
2.2 Scenario and Story	7
3. UML Class Diagram	10
4. Sequence Diagrams	13
4.1 Sequence diagram login:	13
4.2 Sequence diagram admin:	14
4.3 Sequence diagram for booking token:	15
5. Flow Chart Diagram	16
5.1 Flow chart diagram admin:	16
5.2 Flow chart diagram customer:	17
6. Design and Implementation	18
6.1. User data base:	18
6.2. Data base for storing token:	19
7. Testing	20
7.1. Customer sign up page:	20
7.2. Customer log in page:	21
7.3. Customer home page:	22
7.4. QR code generation page:	23
7.5. Token with QR code:	24
7.6. Logout:	25
7.7. Second user:	26
7.8. QR code page for second user rose:	27
7.9. Token for second user rose:	28
7.10 Third user:	29
7.11. QR code page for third user:	30
7.12.Token for third user:	31

8. Future Scope	32
9. Appendix	33
9.1. Code - Customer Lineup Homepage	33
9.2. Code - Connection	33
9.3. Code - Functions	33
9.4. Code – Signup	34
9.5. Code - Login	36
9.6. Code – QR Code	37
9.7. Code – Logout	39
10. Project Reference	39

List of Figures

Figure 1 Class diagram - CLup(Customers Line-up) software.....	7
Figure 2 UML diagram - CLup(Customers Line-up) software	10
Figure 3 Sequence diagram user login.....	13
Figure 4 Sequence diagram Admin login	14
Figure 5 Sequence diagram for booking Token.....	15
Figure 6 Flow chart diagram Admin.....	16
Figure 7 Flow chart diagram Customer	17
Figure 8 User Database	18
Figure 9 Database for storing Token	19
Figure 10 Testing customer sign-up	20
Figure 11 Testing customer login	21
Figure 12 Testing customer homepage	22
Figure 13 Testing QR code generation	23
Figure 14 Testing token generation with QR code	24
Figure 15 Testing customer logout	25
Figure 16 Testing second user login	26
Figure 17 Testing second user homepage.....	26
Figure 18 Testing second user QR code generation	27
Figure 19 Testing second user Token generation and QR code	28
Figure 20 Testing third user login.....	29
Figure 21 Testing third user homepage	29
Figure 22 Testing third user QR code generation.....	30
Figure 23 Testing third user Token generation and QR code.....	31

1. Introduction

1.1 Purpose

This SRS describes the software functional and non-functional requirements for release 1.0 of the CLup: Customers Line-up. This software allows store managers to regulate the influx of people in the building and, on the other side saves people from having to line up and stand outside of stores for hours on end, all requirements specified here are high priority and committed for release 1.0.

1.2 Scope

The Clup: Customers Line-up software consists of following major functions:

- (1) To allow the store managers to regulate the influx of people in the building.*
- (2) Users who intend to visit the store can generate tokens that gives their position in the queue.*
- (3) To generate QR codes for the user that would be scanned upon entering the store.*
- (4) The software should display list of available time-slots in a day/time range to the user.*
- (5) To allow users to book visiting time-slots and expected duration of the visit.*
- (6) To allow users to indicate list and categories of items to purchase.*
- (7) The software should be able to suggest users for alternative time-slots(same day or different day), changes to visit other stores(same chain or chain-independent)in case of time-slots clashes with other users or maxing out the pre-defined capacity of the stores.*

1.3 Process Requirements

There are no constraints nor requirements on the process to be followed in the application.

1.4 Environmental Constraints

There are no environmental constraints, ither than the fact that the application should be as widely available as possible.

1.5 Project Restrictions

There are no specific restrictions.

1.5 Project License

There are no constraints on the licenses to be used for the developed team. A GPI FLOSS license(or a LGPL license) is recommended.

2. Project Description

The coronavirus emergency has put a strain on society on many levels, due to many countries imposing lockdowns that allow people to exit their homes only for essential needs, and enforcing strict rules even when people are justified in going out (such as limiting the number of accesses to buildings and keeping a distance of at least one meter between people). In particular, grocery shopping---a most essential need---can become a challenge in the presence of such strict rules. Indeed, supermarkets need to restrict access to their stores to avoid having crowds inside, which typically results in long lines forming outside, which are themselves a source of hazards. In these trying times, people turn to technology, and in particular to software applications, to help navigate the challenges created by the imposed restrictions.

The goal of this project is to develop an easy-to-use application that, on the one side, allows store managers to regulate the influx of people in the building and, on the other side, saves people from having to line up and stand outside of stores for hours on end.

The application would work as a digital counterpart to the common situation where people who are in line for a service retrieve a number that gives their position in the queue. Naturally, physically retrieving a number forces people to first approach the building, and then wait in close proximity (though not in a line) until their number is called, which is a less than ideal situation in a lockdown situation. A software application, instead, could provide many improvements to the situation described above.

For example, it would allow customers to “line up” (i.e., retrieve a number) from their home, and then wait until their number is called (or is close to being called) to approach the store. In addition, the application could be used to generate QR codes that would be scanned upon entering the store, thus allowing store managers to monitor entrances. For the application to effectively work in practice, all customers should use it to access the store, which has a number of consequences, including the following ones:

- *The usability of the application should be very high, as the range of users include all demographics (everyone needs to do grocery shopping).*
- *Fallback options should be available for people who do not have access to the required technology; for example, stores should also have the possibility to hand out “tickets” on the spot, thus acting as proxies for the customers.*

In addition to managing lines in real-time, the application could also allow customers to “book” a visit to the supermarket. This feature would be similar to the booking of a slot for visiting, say, a museum/exhibition, but with important differences. In particular, whereas one can expect that the time that it takes to visit a museum is fairly uniform (and people would typically want to visit the whole museum/exhibition), the same is not true for visits to the supermarket. Hence, upon booking a visit, a customer might indicate also the approximate expected duration of the visit. The application might also allow users to indicate, if not the exact list of items that they intend to purchase, the categories of items that they intend to buy. This would allow the application to plan visits in a finer way, for example allowing more people in the store, if it knows that they are going to buy different things, hence they will occupy different spaces in the store when they visit (thus respecting the requirement that people keep enough distance between them).

Other features that the application might have include a suggestion of alternative slots (in the same day, or in different days) for visiting the store, to balance out the number of people in the store, the suggestion of different stores of the same chain (or even of different chains, if the application is chain-independent) if the preferred one is not available, or the periodic notification of available slots in a day/time range (these are other important differences with respect to museums/exhibitions, which are unique, and which are usually visited only once).

2.1 Software Requirements

2.1.1 Functional Requirements

REQUIREMENTS	SOFTWARE FUNCTIONALITY
[REQ #1]	The software allows store manager to input the login credentials
[REQ #2]	The software allows store manager to set the timeslots.
[REQ #3]	The software allows store manager to modify the timeslots.
[REQ #4]	The software allows store manager to add the duration of visit.
[REQ #5]	The software allows the store manager to modify the duration of visit.
[REQ #6]	The software allows the store manager to set the store capacity.
[REQ #7]	The software allows the store manager to modify the store capacity.
[REQ #8]	The software allows the store manager to send the store information to the system.
[REQ #9]	The software allows the store manager to regulate the influx of people in the building.
[REQ #10]	The software allows the user to insert the user id.
[REQ #11]	The software allows the user to enter the user position manually.
[REQ #12]	The software allows the user to grant access to device location coordinates.
[REQ #13]	The software displays a list of stores based on user's position.
[REQ #14]	The software allows the user to select a store based on his preference.
[REQ #15]	The software checks with the system and validates the user preferred store choice.
[REQ #16]	The software allows user to select the preferred timeslot for that store.
[REQ #17]	The software checks with the system and validates the user preferred timeslots for that store.
[REQ #18]	The software suggests user to change timeslot for that store if the store is already booked with max capacity.
[REQ #19]	The software suggests user to change the store preference if the initial store is running at max-capacity or if it does not have free timeslots.
[REQ #20]	The software allows the user to enter the desired visit duration for the preferred store.

[REQ #21]	The software checks with the system and validates the user preferred visit duration.
[REQ #22]	The software allows the user to select a list of shopping items based on their categories and their quantities.
[REQ #23]	The software allows the user to modify the list of shopping items and their quantities.
[REQ #24]	The software allows the user to request a token before visiting the store.
[REQ #25]	The software sends the token generation request to the system.
[REQ #26]	The software receives a unique token number and a unique token QR code from the system and displays it to the user.
[REQ #27]	The software allows the user to request the current queue position for the preferred store.
[REQ #28]	The software allows user to scan the token QR code at the validation machine before entering the store.

Table 1 CLup(Customers Line-up) software requirements

2.1.2 Process Requirements

There are no constraints nor requirements on the process to be followed in developing the application

2.2 Scenario and Story

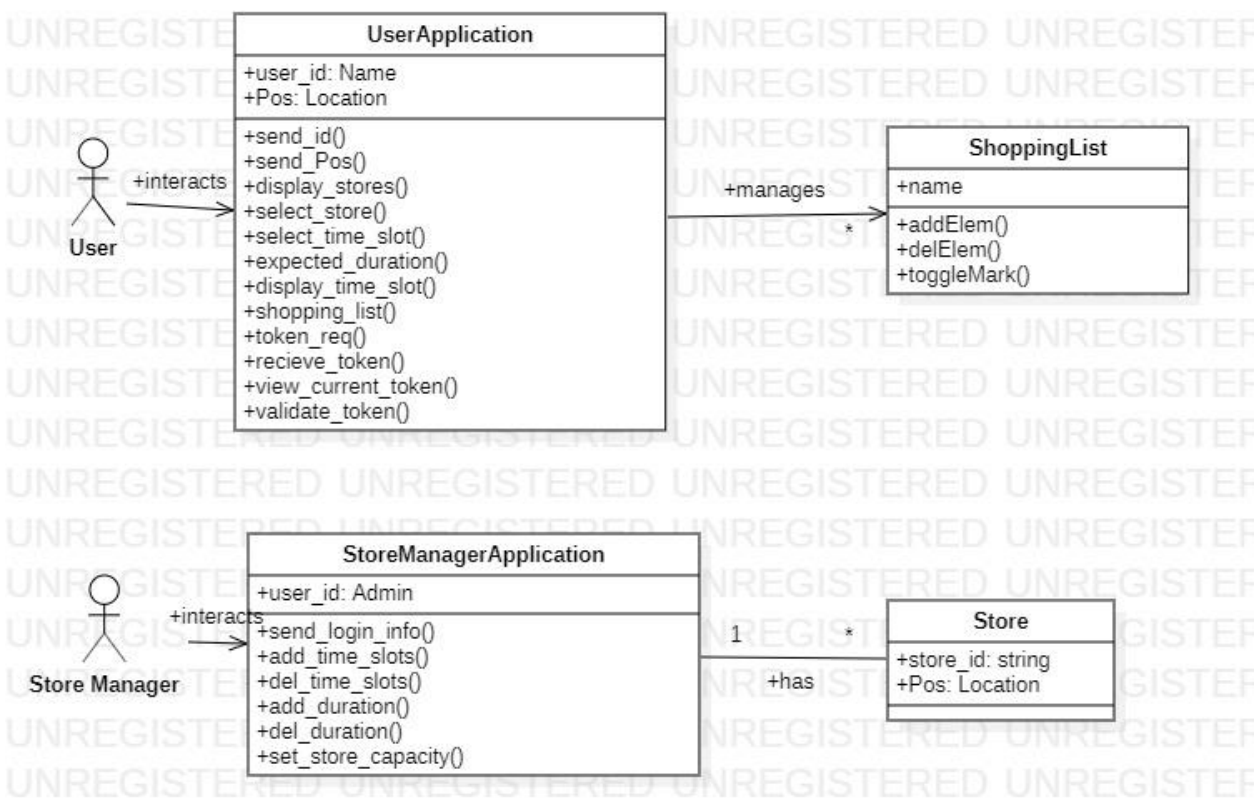


Figure 1 Class diagram - CLup(Customers Line-up) software

Scenario	Story
<u>1.</u> Store Manager enters the login credentials in the app.	The store manager inserts the login credentials in the application. The login credentials are unique username and password given to the store manager by the system administrator and is unique to each store. The application sends the admin login credentials to the system for authentication. If the login credentials are invalid, the admin is prompted a message stating that the login credentials are invalid and suggest for re-entry of login credentials. Upon successful authentication through the system, the admin receives a message from the system to the store manager application stating that the login credentials are valid.
<u>2.</u> Store Manager manages the timeslots	The store manager uses the application to authenticate the login credentials. The admin adds time-slots, add expected visit duration using the application. Similarly, the admin modifies the time-slots, expected visit duration using the same application.
<u>3.</u> Store Manager manages the store capacity	The store manager changes the store capacity using the application in real-time. The store manager can later modify the store capacity. Finally, the store manager sends the entire store info(store id, store location, time-slots, expected visit duration and store capacity) to the system and receives a reply message from the system when the details are updated.
<u>4.</u> User enter the login credentials	The user enters the user id(a unique name/guest name) into the user application. The user id is sent to the system for validation (Example of admissible id's – andrea21, george672, Example of incorrect id's - %4@\$*-+). If a user enters name Andrea, it may or may not be accepted by the system because the system accepts a unique name different than the other user's name stored in the database. Similarly, if the user enters symbols in the name, it gets flagged automatically as invalid name in the system. If the user id is invalid, the user is prompted to re-enter a new name id until the conditions of a valid name id are satisfied.
<u>5.</u> User chooses time slots, preferred location and selects store	The user enters the user id in the application. If the user id is valid, the user is requested to provide his location by manually entering it or by access grant through the user's device gps coordinates. After receiving the user's location, the systems scans the store info in the database which was received from the store manager previously. The system compares the user location with the stores location and send a list of stores for selection by the user in the user's location perimeter.
<u>6.</u> User modifies time slots, selects visit duration or changes store	The user enters the preferred store, time-slots and expected duration through the user application. The

	<p>application send the user inputs to the system, the system check the user inputs and matches it with the store info in the database, validates the choices and sends the store confirmation to the user. If the user's inputs doesn't match with those of the store info in the database , the system suggests for changing the store or changing the preferred slots and visit duration.</p>
<p><u>7.</u> User creates a shopping list by adding, deleting or selecting items.</p>	<p>Once the store is confirmed, the user selects the ShoppingList interface through the user application. The user can add list of products and mark their quantities based on their categories. The user can also modify his preferences through the same shopping list interface by adding or deleting products.</p>
<p><u>8.</u> User request generation of Token and checks current queue position</p>	<p>The user uses the user application to request the system to generate the Token(a numerical number corresponding to the current queue position for that particular store) and a unique Token QR code. The user requests the current queue position through the user application and the request is passed to the system. The system then replies to the user regarding the current queue position to the user through the user application.</p>
<p><u>9.</u> User scans and validates Token QR code at Validation Machine</p>	<p>The user scans the QR code through the user application at store's validation machine for entering the store. If the user violates the current queue position by scanning the QR code before the current queue position, the validation machine will notify the user to wait until his/her position in the queue.If the QR code is valid, the validation machine interacts with the gates to open for entering the store.</p>

Table 2 Scenario and Story

3. UML Class Diagram

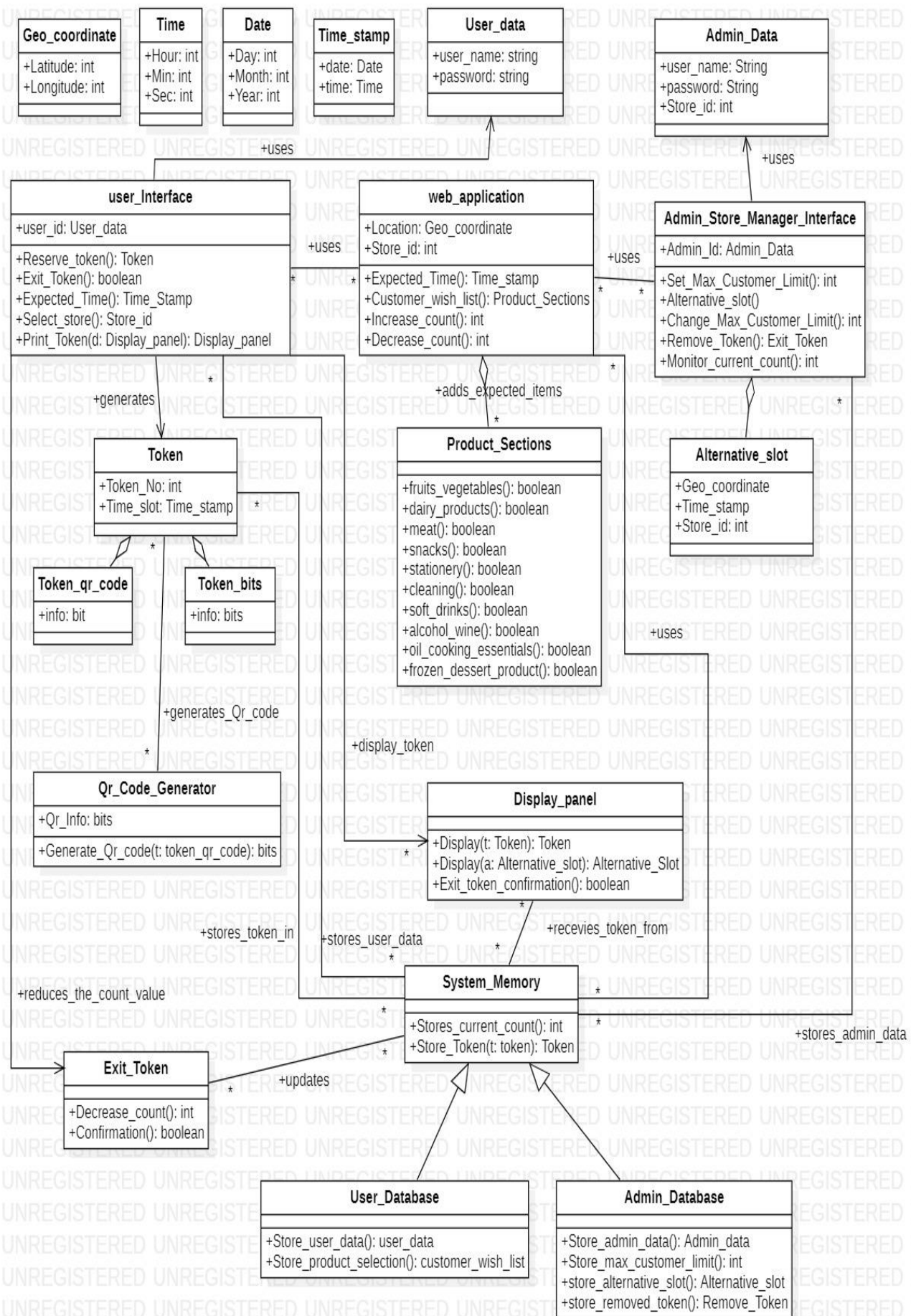


Figure 2 UML diagram - CLup(Customers Line-up) software

Components of class diagram:

- *User*
- *User interface*
- *Supermarket sales application*
 - *Product sections*
- *Admin store manager interface*
 - *Alternative time slot*
- *Token*
 - *QR code generator*
- *System memory*
 - *User database*
 - *Admin database*
- *Display panel*
- *Exit token*

Bullets representation:

- *Main class*
- *Sub-class*

User:

- *In this application customer is a user. Who uses this application to form a virtual queue.*

User interface:

- *the user uses the user interface for doing specific operations. Such as (select store, enter expected time, enter customer product wish list, reserve token, print token).*

Supermarket sales application:

- *it has two attributes (store id, geometrical coordinates)*
- *four operations (customer wish list, expected time, increase count, decrease count).*
- *Counter helps to keep track of customer traffic inside the store.*

Product selection:

- *This class is the subclass for the supermarket sales application.*
- *It has operations that contain product sections, user can able to add expected items.*

Admin store manager interface:

- *Admin is the store manager, it contains admin id in the attributes.*
- *Admin can able to do the following operations.*
- *Can set the maximum customer limit allowable inside the store in a particular time slot.*
- *Can monitor the customer's count inside the store with the help of a counter.*
- *Can change the maximum customer limit, according to the influx of customers in the queue.*
- *Can allocate an alternative time slot if necessary.*
- *Can remove the token in the case of malfunctions, this operation will be discussed in the exit token.*

Alternative time slot:

- *It is the subclass to the admin store manager interface.*
- *It contains geometrical coordinates of the store, alternative time slot, store id.*

Token:

- *The token contains token no, a time slot in the attribute.*
- *It invokes QR code from QR code generator and stores it in system memory.*

QR code generator:

- *This class generates the QR code which is combined with the token number while printing the token.*

System memory:

- *System memory is the internal memory that is used by the system to store the data of the application.*
- *It stores (current count, token).*
- *It has two subclasses such as (user database, admin database).*
- *User database stores user data.*
- *Admin database stores admin data.*

Display panel:

- *The display panel performs three operations.*
- *It can print and display tokens regarding the request of the user interface.*
- *It can display the alternative available time slot to the user.*
- *It can display the confirmation message of an exit token.*

Exit token:

- *This class is used to monitor the real-time customer present inside the supermarket.*
- *When the customer performs the exit token, the counter can allow the next customer inside the store.*
- *This sequence helps the admin to monitor the customer's maximum capacity inside the store.*
- *As a precaution we have included the confirmation message in the display, to prevent the accidental performance of exit token.*
- *If the customer forgets to perform the exit action, the admin will remove that particular token by performing the remove token action. This operation is included for safety purposes*

4. Sequence Diagrams

4.1 Sequence diagram login:

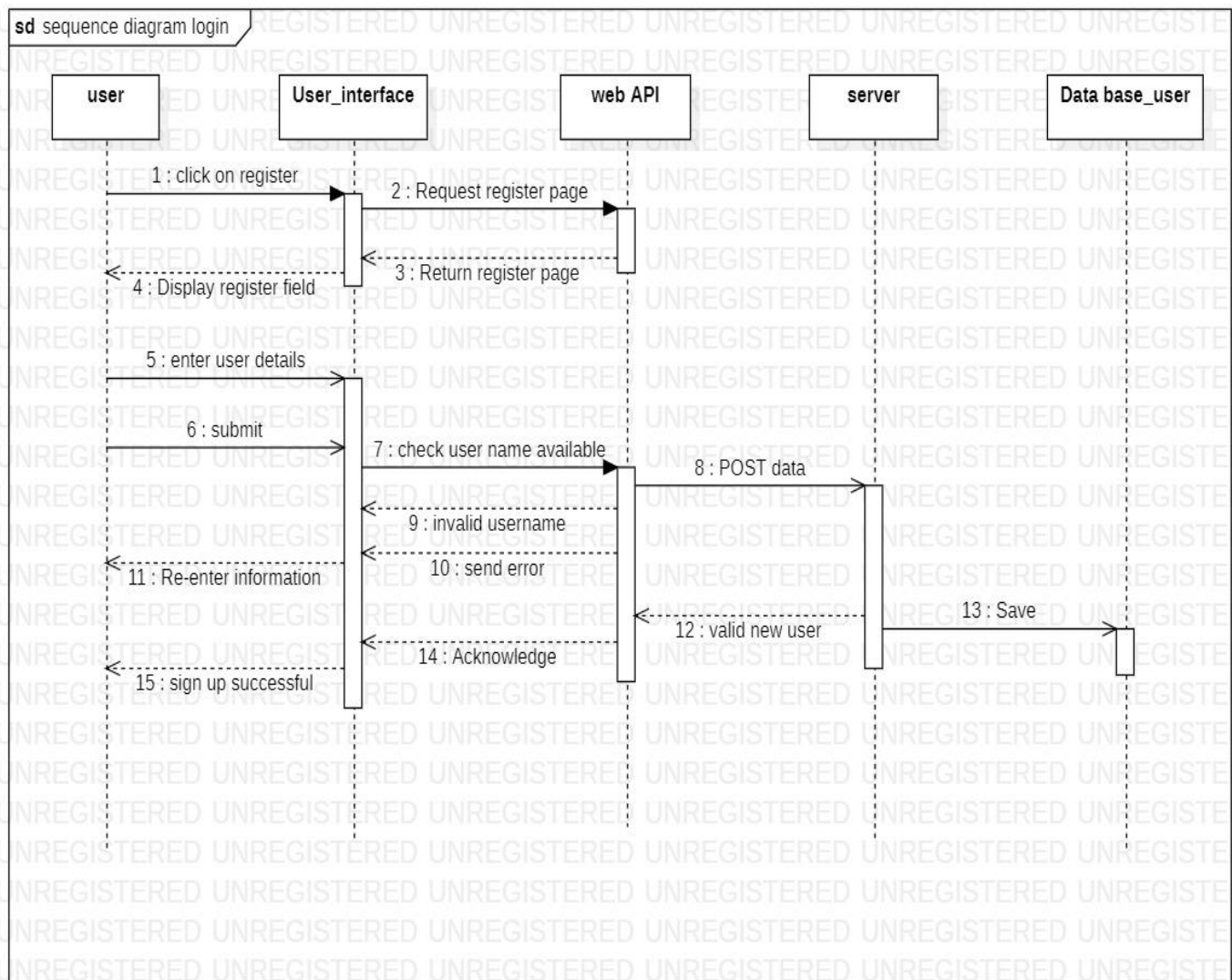


Figure 3 Sequence diagram user login

- User request for registration on the home page.
- The user enters their user details on the home page.
- Confirms their details by clicking the submit button.
- Username availability is checked in web API.
- If the user gets an available username, then the user posts that valid data in the service.
- Finally, the user saves their data by storing it in the database.
- If the user gets an invalid username, the web API sends an error message to the home page.
- Again, the user re-enters signup credentials.
- The end-user gets the signup successful message on their home page for their confirmation.

4.2 Sequence diagram admin:

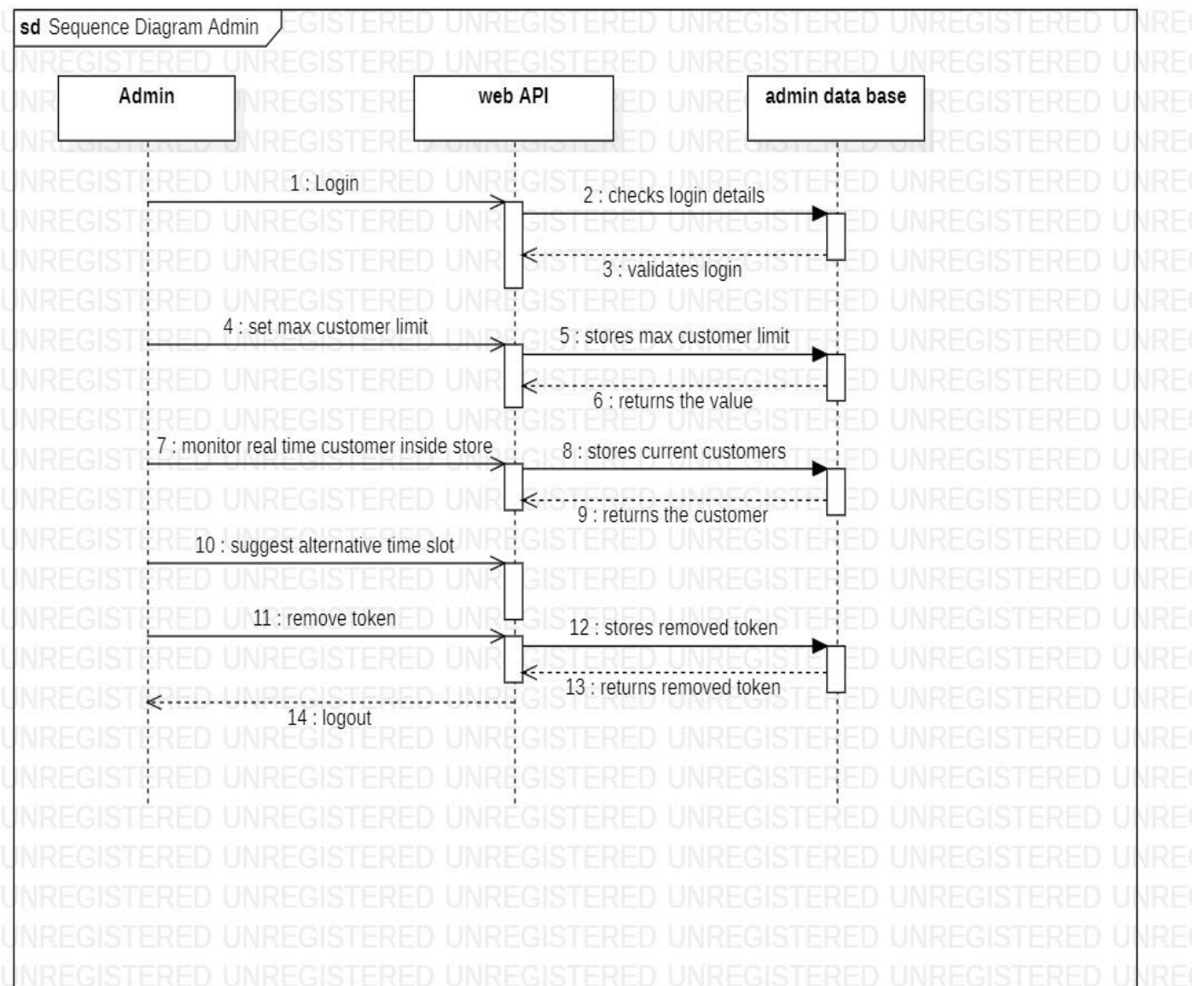


Figure 4 Sequence diagram Admin login

- The store manager (admin) inserts the login credentials into the booking system application.
- The admin login credentials are validated through system memory.
- The admin can now add alternative timeslots.
- The admin can set the maximum customer limit into the application and send the entire store info (store id, store location, timeslots).

4.3 Sequence diagram for booking token:

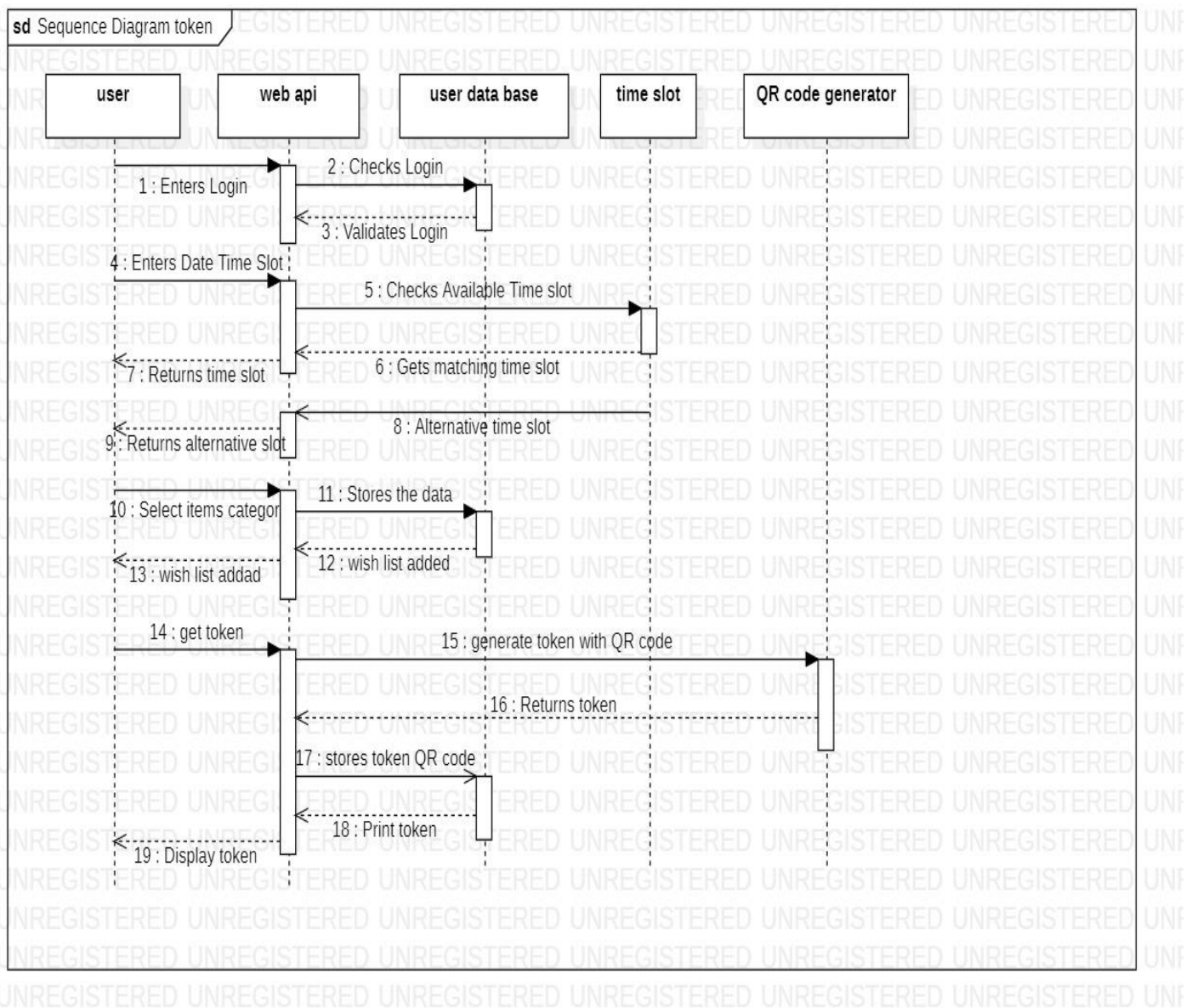


Figure 5 Sequence diagram for booking Token

- User(customer) enters their login credentials in the booking system.
- The booking system validates the login credentials by referring them to the customer database.
- The user enters the expected time and date in the booking system.
- Users will get a matching time slot when there is a free time slot available for that particular time.
- Otherwise, the booking system returns the alternative time slot.
- The user selects the category of the expected item in the booking system.
- Booking system stores that data in the customer database.
- Finally, the user clicks the get token to generate the token.
- The booking system generates the token with a QR code and stores it in the customer database.
- Booking system prints the token and displays it to the user.

5. Flow Chart Diagram

A flow chart is another important diagram to describe the dynamic aspects of the system. It is a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. There are two types of users in our application. One is a customer, and another is an admin.

5.1 Flow chart diagram admin:

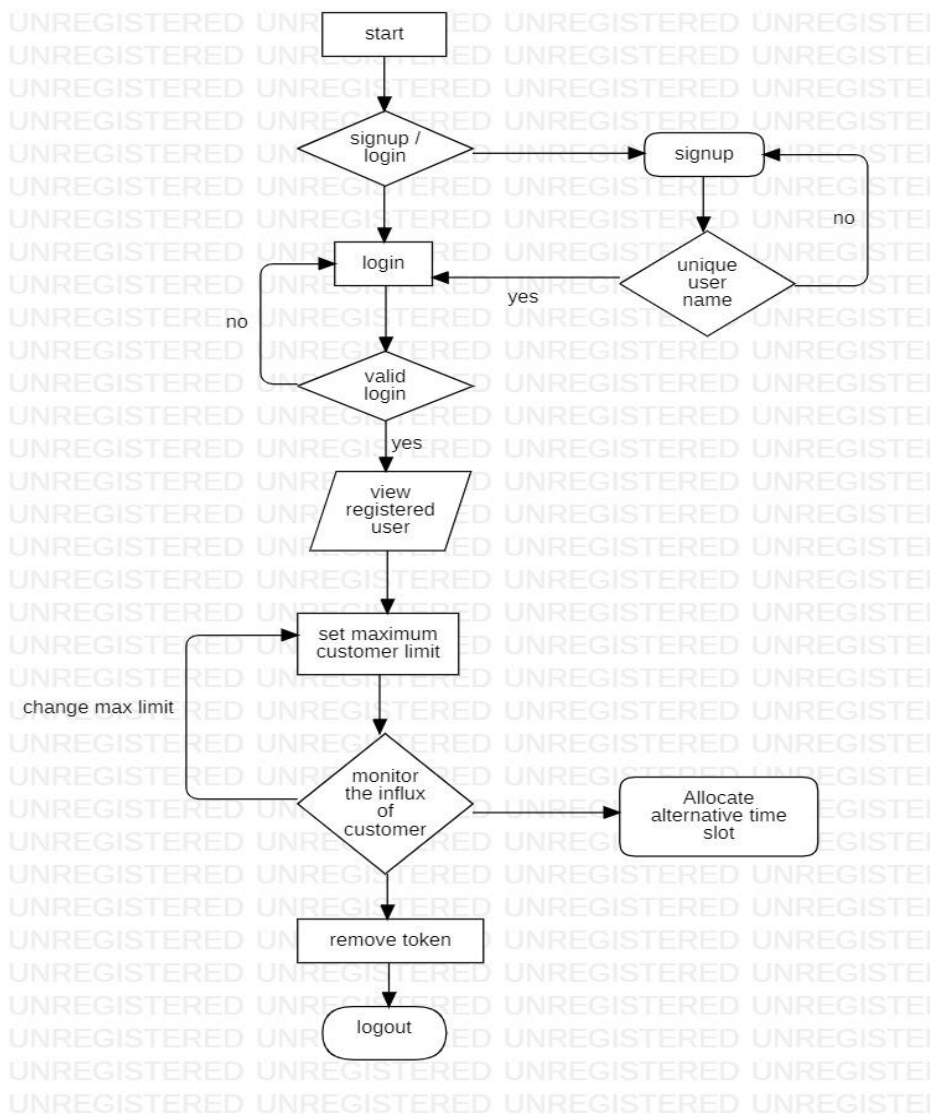


Figure 6 Flow chart diagram Admin

- Admin is the store manager of the supermarket; he is responsible for the control of influx of customers inside the supermarket.
- He / she will register in the web application in order to access the admin data base.
- After registering in the web application, he can log in to the web application.
- He will set the limit for maximum customers inside the store.
- He can change the set limit, if the situation demands.
- He can view the user registered in the data base.
- He will allocate alternative time slot for the customers based on the available time slot.
- He can remove the token in case of any malfunction or if the customer forgets to perform exit token action.
- To make the queue functioning without deadlock.
- Finally, the admin will logout of the web application.

5.2 Flow chart diagram customer:

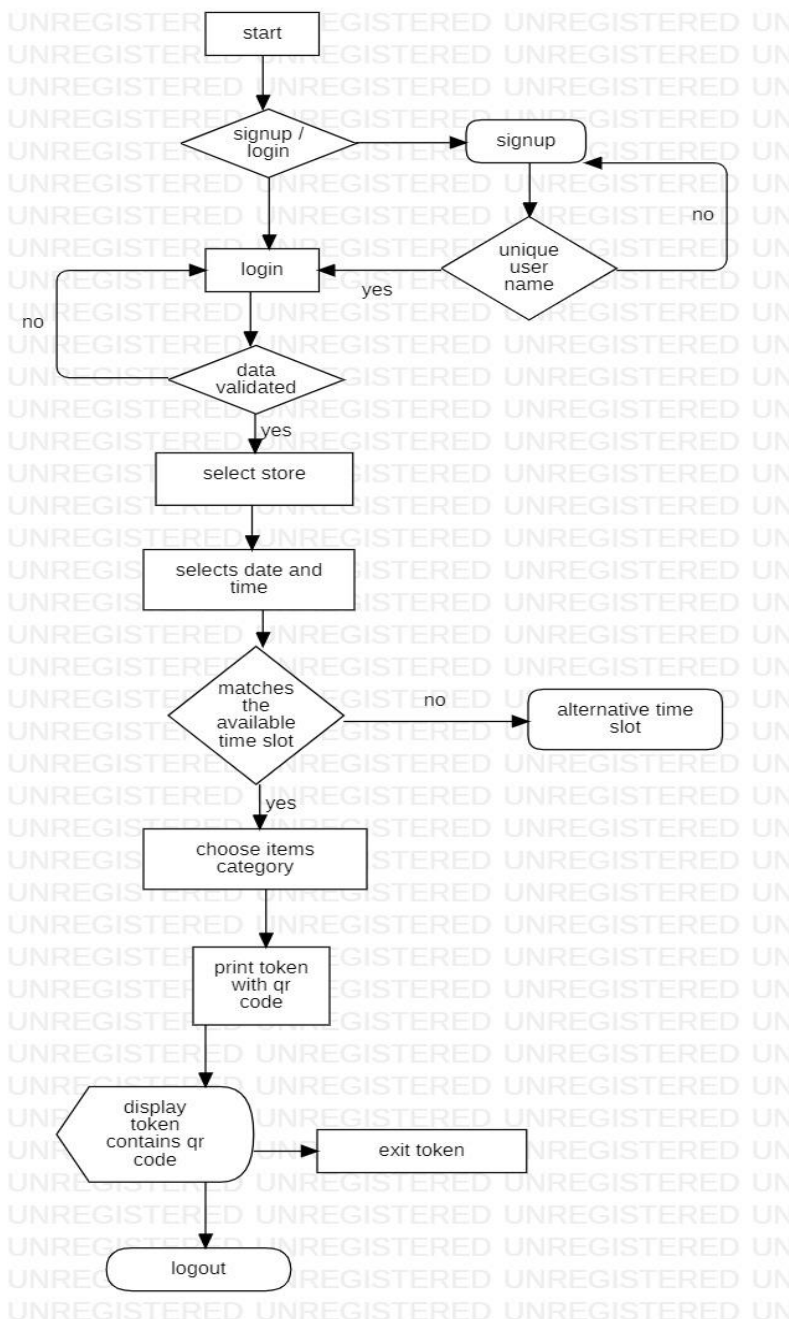


Figure 7 Flow chart diagram Customer

- Customer will register in the web application; customers data is stored in the user data base.
- After registering he will log in to the web application.
- In this step customers data is validated by verifying in the user data base.
- He will choose the store.
- He will select the date and time slot.
- If the search query does not match the available time slot, he will be redirected to the alternative time slot.
- If it matches with available time slot perfectly, then the user can choose the items category which they wish to buy.
- He will perform the print qr code action in order to print the token.
- Token will have token number and qr code along with it.
- He will perform exit token action to delete the token.
- Finally, he can log out of the web application.

6. Design and Implementation

6.1. User data base:

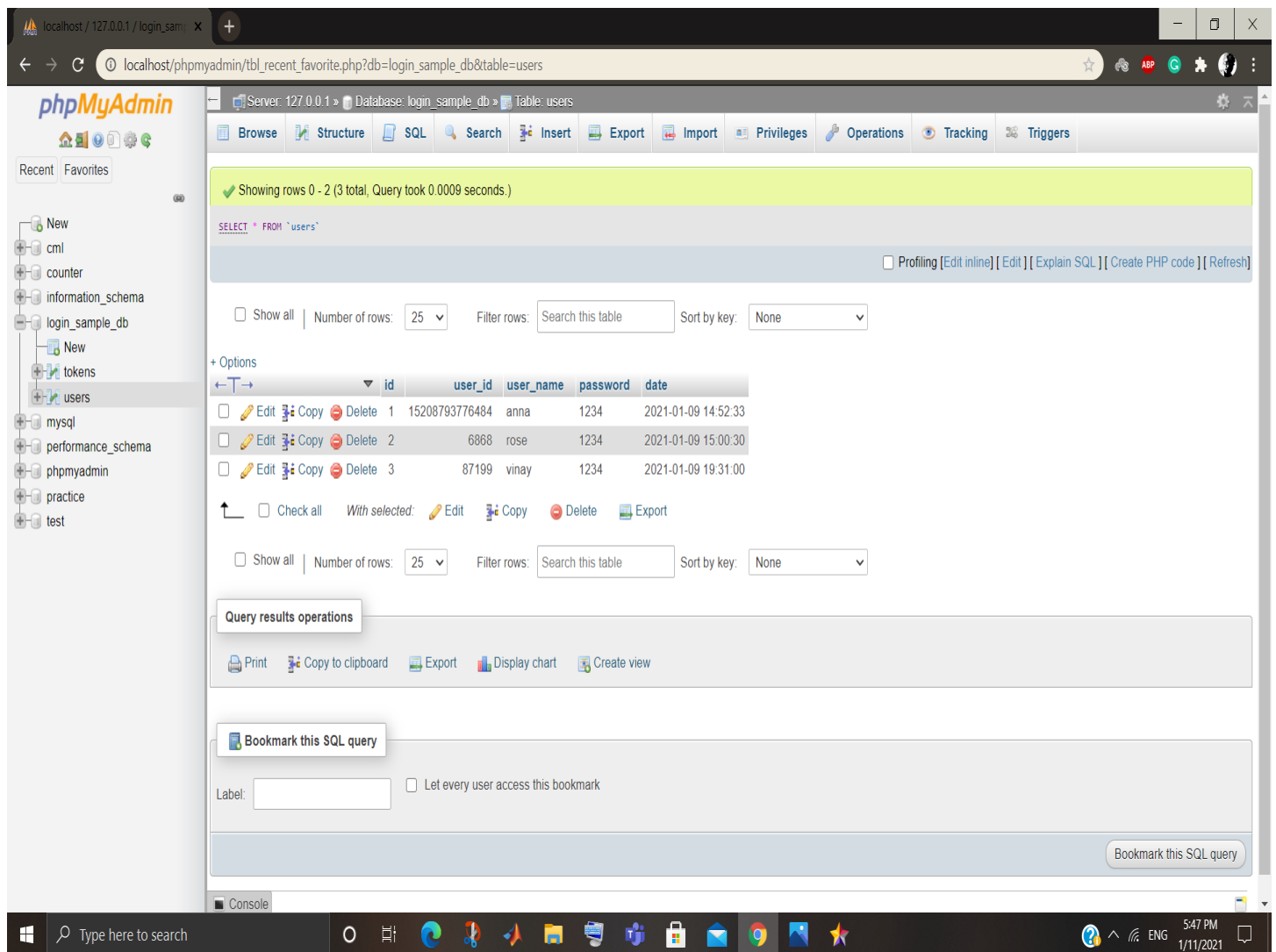


Figure 8 User Database

- We used xampp control panel software for the server.
- All the files are created with the help of sublime text editor in php format.
- Google chrome was used as the browser for the implementation and testing purpose.
- Mysql was used for storing our admin and customers data in the data base.
- Database name: login_sample_db
- Table: users
- Create: id, username, password, date.
- Id: primary key, auto increment , data type: int
- Username and password: datatype: varchar.

6.2. Data base for storing token:

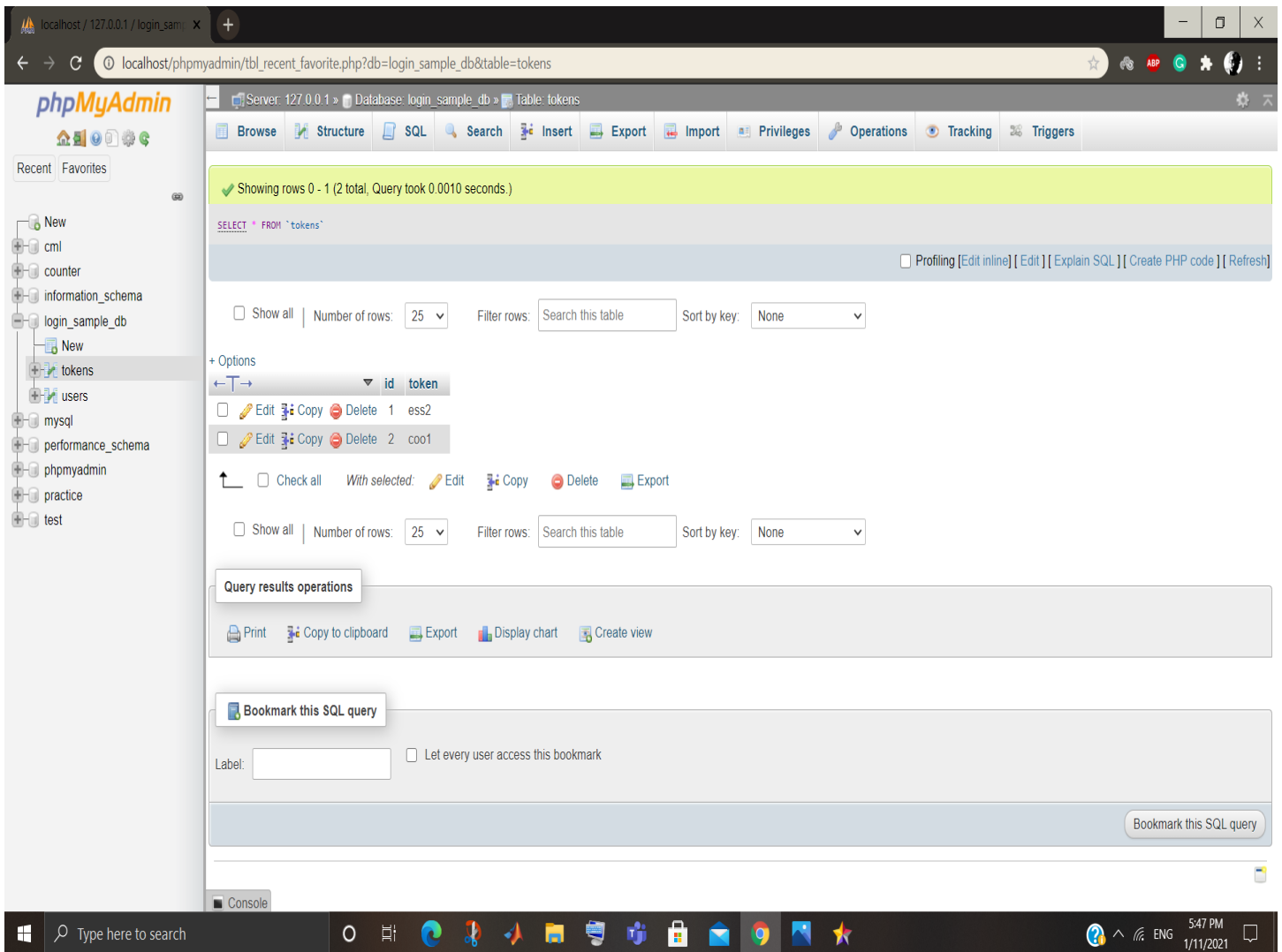


Figure 9 Database for storing Token

- **Data base name:** `login_sample_db`
- **Table:** `tokens`
- **Create:** `id, token`
- **Id:** primary key, data type: `int`
- **Token:** data type: `varchar`

7. Testing

7.1. Customer sign up page:

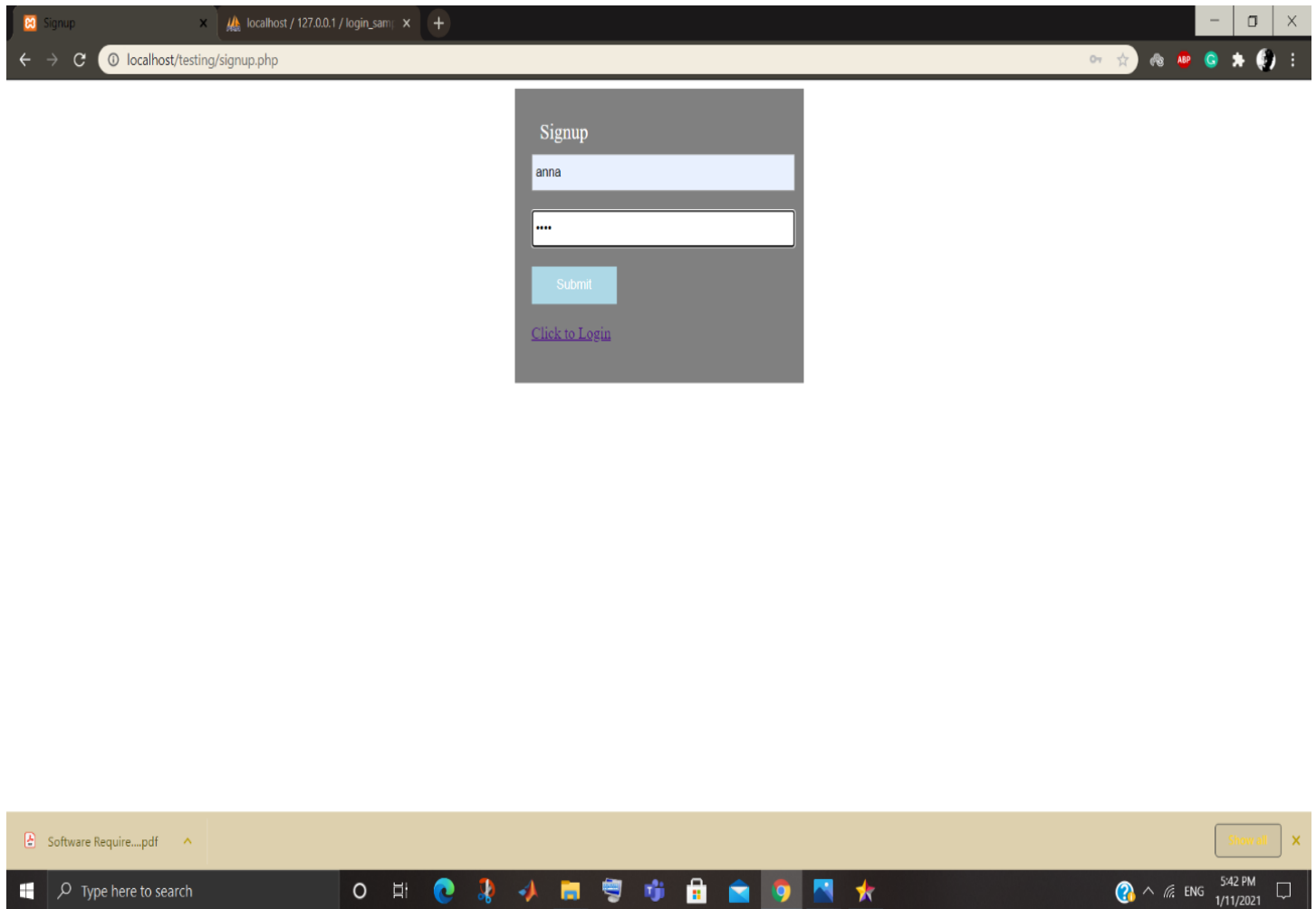


Figure 10 Testing customer sign-up

- Username: anna
- Password: 1234
- Submit
- After submitting it redirects to log in page.

7.2. Customer log in page:

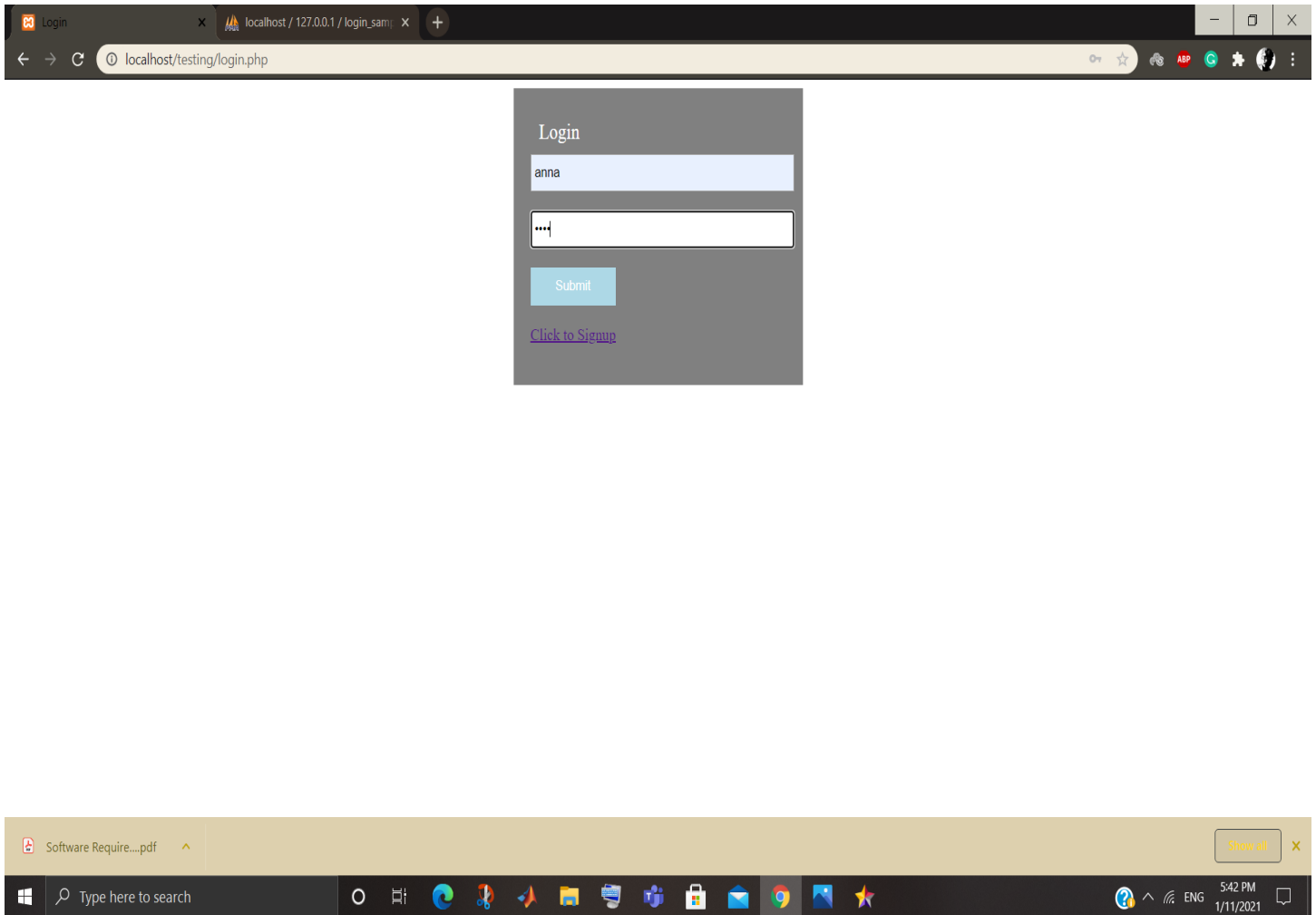


Figure 11 Testing customer login

- Username: anna
- Password: 1234
- Submit
- After submitting, it goes to index page.

7.3. Customer home page:

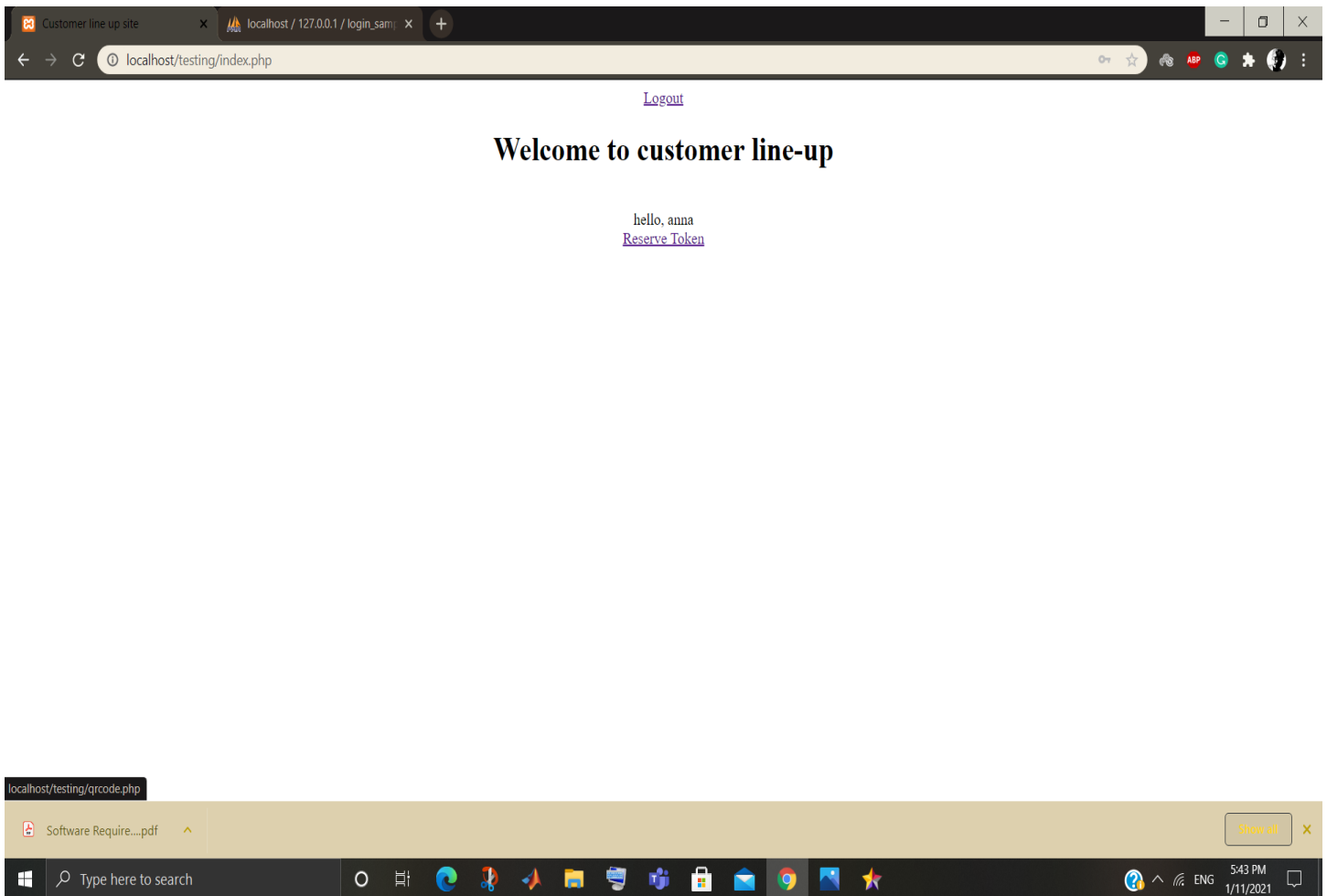


Figure 12 Testing customer homepage

- *Home page will show the username.*
- *If he clicks, reserve token.*
- *It goes to next page.*

7.4. QR code generation page:

generate QR (quick response) code

Supermarket store id for Esselunga: 1, for Coop: 2.

Enter your preferred date below: for example 01-01-2021. day-month-year format.

07-01-2021

Enter your expected time slot below: for example 11:00am-11:15am

10:00am-10:15am

Choose your preferred supermarket

☒ Esselunga
☐ Coop

Choose atleast one the category sections!

fruits vegetat
dairy products
stationery prod
cleaning prod

Get Token

You are required to fill all the fields.
[Go to home page](#)

Figure 13 Testing QR code generation

- User must give the date.
- User must mention time in the requested format.
- User must choose the preferred supermarket.
- User must choose category section.
- After completing the all field. He can submit the form.
- This will generate the qr code below.

7.5. Token with QR code:

The screenshot shows a web browser window with two tabs. The active tab is 'localhost/testing/qrcode.php'. The page content includes a form for generating a token. The form has the following fields and options:

- Input field: 'enter date here' (placeholder text: 'Enter your preferred date below. for example 01-01-2021. day-month-year format.')
- Input field: 'enter time here' (placeholder text: 'Enter your expected time slot below. for example 11:00am-11:15am')
- Radio buttons for 'Choose your preferred supermarket':
 - ☐ Esselunga
 - ☐ Coop
- Dropdown menu for 'Choose atleast one the category sections!':
 - fruits vegetab
 - dairy product
 - stationery pro
 - cleaning prod
- Button: 'Get Token'

Below the form, the generated token is displayed: 'Esselunga Token: ess1'. Below the token, the date and time are shown: 'Monday, 11th January 17:43'. A QR code is generated based on the token and date/time. At the bottom of the page, there is a 'Go to home page' link. The browser's address bar shows 'localhost/testing/qrcode.php'. The Windows taskbar is visible at the bottom of the screen.

Figure 14 Testing token generation with QR code

- *Qr code consists of all the data which user submitted.*
- *Her token number is ess1.*
- *It represents her token number in the queue.*
- *For esselunga it is like ess as prefix.*
- *It also shows date and time while generating the qr code.*
- *If user click the go to home page link.*
- *It goes to the home page.*

7.6. Logout:

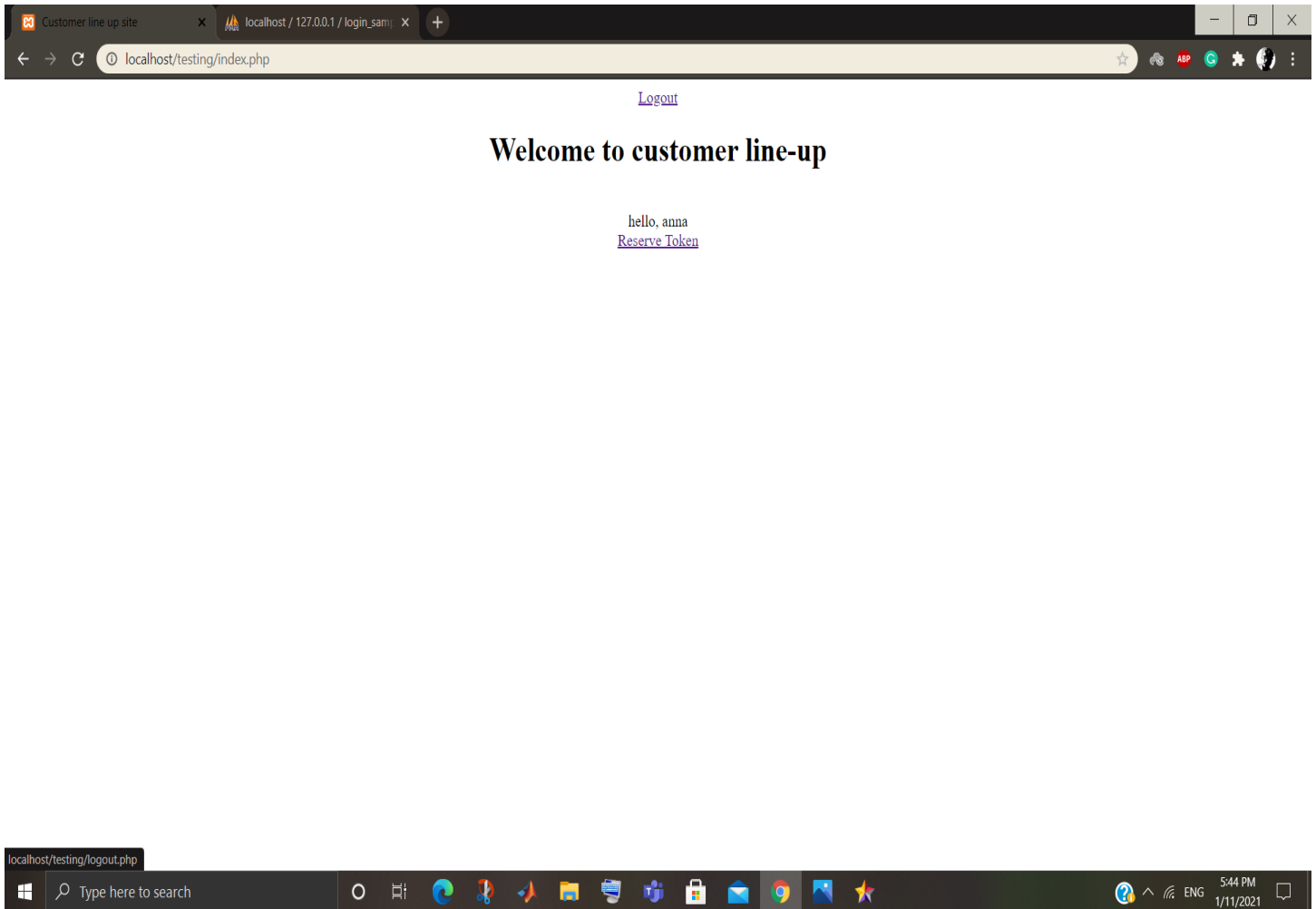


Figure 15 Testing customer logout

- Finally, the user anna can logout.

7.7. Second user:

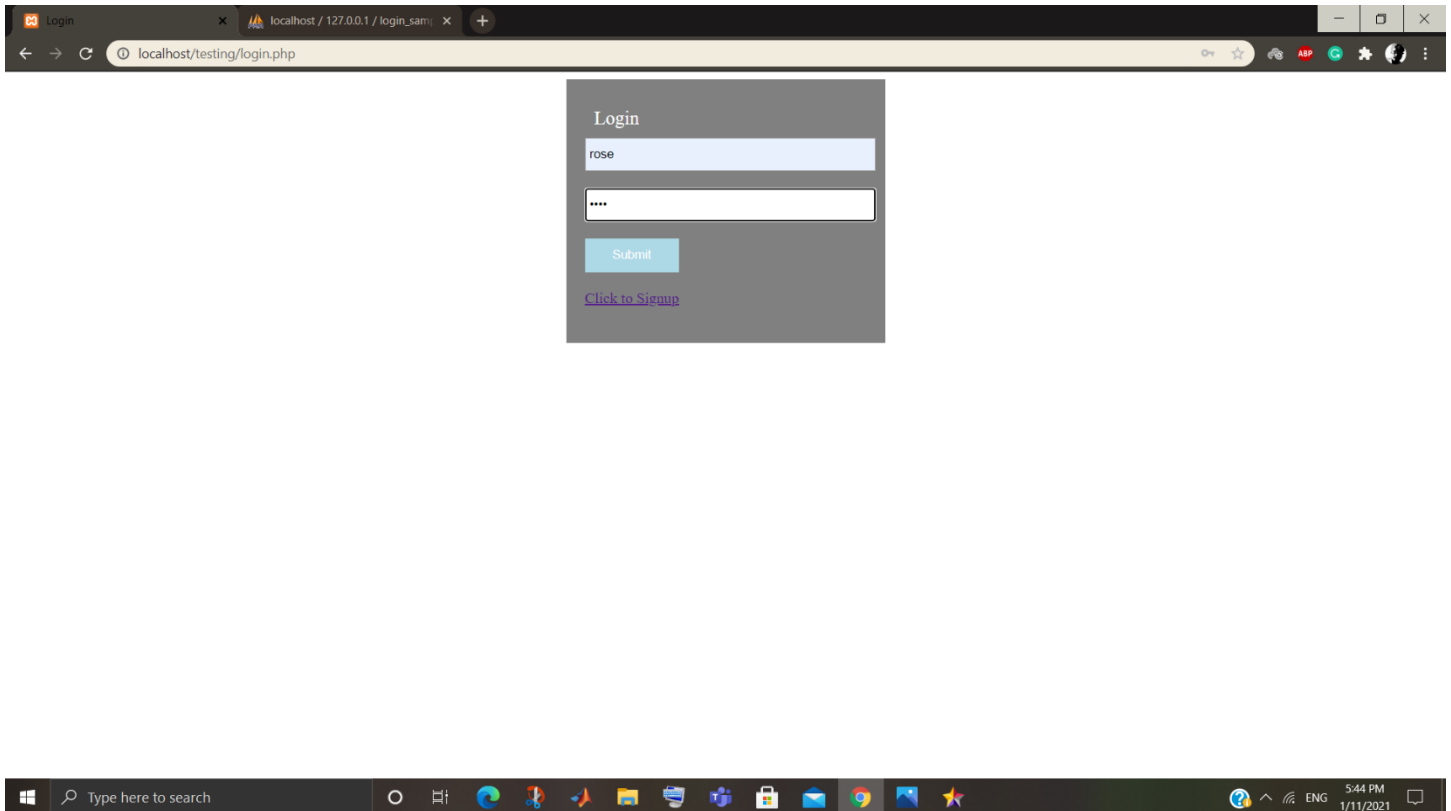


Figure 16 Testing second user login

- *Username: rose*
- *Password: 1234*
- *Submit.*

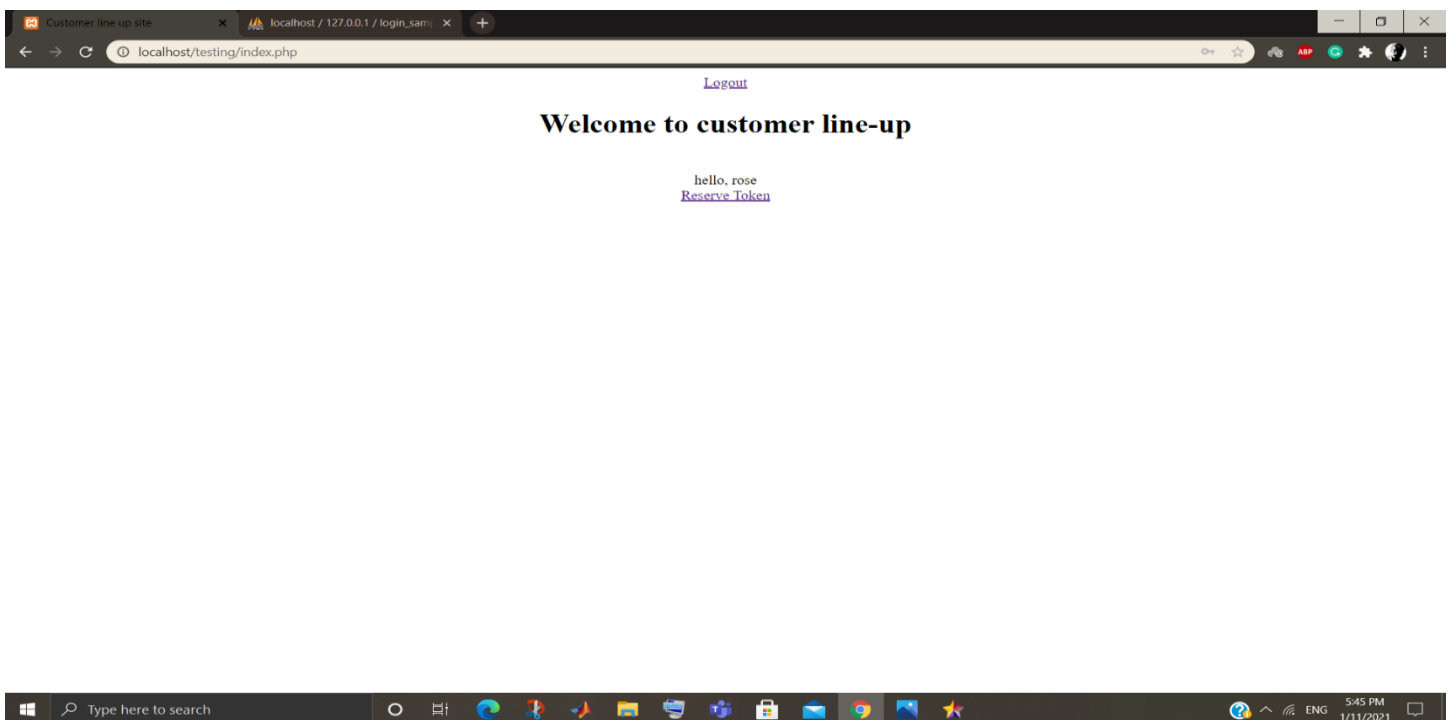


Figure 17 Testing second user homepage

- *It goes to home page of rose.*

7.8. QR code page for second user rose:

generate QR (quick response) code

Supermarket store id for Esselunga: 1, for Coop: 2.

Enter your preferred date below: for example 01-01-2021. day-month-year format.

07-01-2021

Enter your expected time slot below: for example 11:00am-11:15am

10:00am-10:15am

Choose your preferred supermarket

☒ Esselunga
☐ Coop

Choose atleast one the category sections!

fruits vegetab
dairy product
stationary pro
cleaning pro

Get Token

You are required to fill all the fields.
[Go to home page](#)

Figure 18 Testing second user QR code generation

- *Here, again she is choosing esselunga.*
- *So, the token will be generated according to the previous user in the queue.*

7.9. Token for second user rose:

The screenshot shows a web browser window with two tabs. The active tab is 'localhost/testing/qrcode.php'. The page content includes the following elements:

- A text prompt: "Enter your preferred date below. for example 01-01-2021. day-month-year format." followed by an input field labeled "enter date here".
- A text prompt: "Enter your expected time slot below. for example 11:00am-11:15am" followed by an input field labeled "enter time here".
- A text prompt: "Choose your preferred supermarket" followed by two radio buttons: "Esselunga" (selected) and "Coop".
- A text prompt: "Choose atleast one the category sections!" followed by a dropdown menu with options: "fruits vegetab", "dairy product", "stationery pro", and "cleaning prod".
- A "Get Token" button.
- A large heading: "Esselunga Token: ess2".
- A timestamp: "Monday, 11th January 17:45".
- A QR code.
- Footer text: "Good day!" and "Take screenshot before leaving!" with a link "Go to home page".

The browser's address bar shows 'localhost/testing/qrcode.php'. The Windows taskbar at the bottom shows the search bar, task view, and several application icons. The system clock indicates 5:46 PM on 1/11/2021.

Figure 19 Testing second user Token generation and QR code

- Her token number is *ess2*.
- It shows her number in the queue.

7.10 Third user:

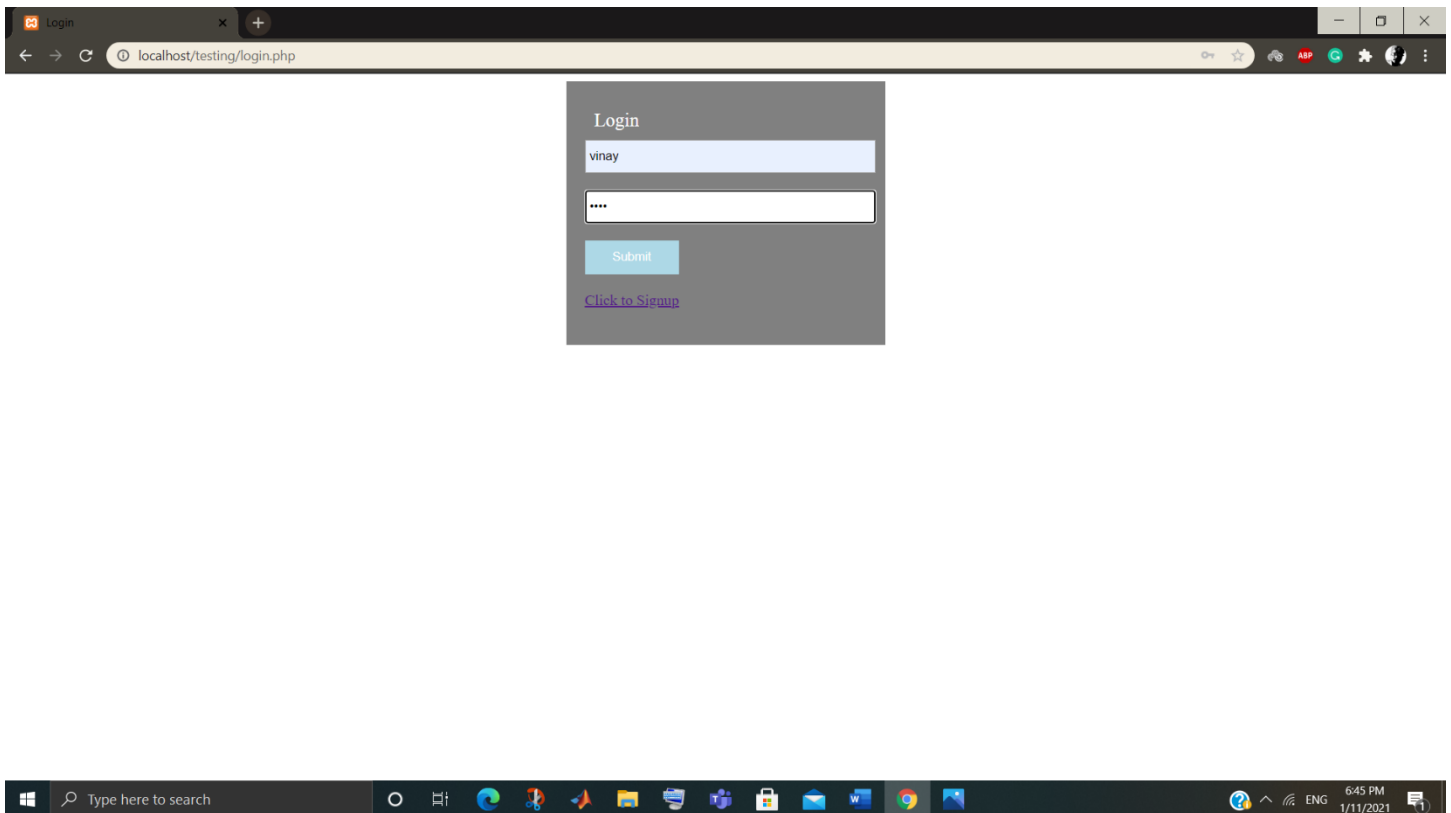


Figure 20 Testing third user login

- Username: vinay
- Password: 1234
- Submit.

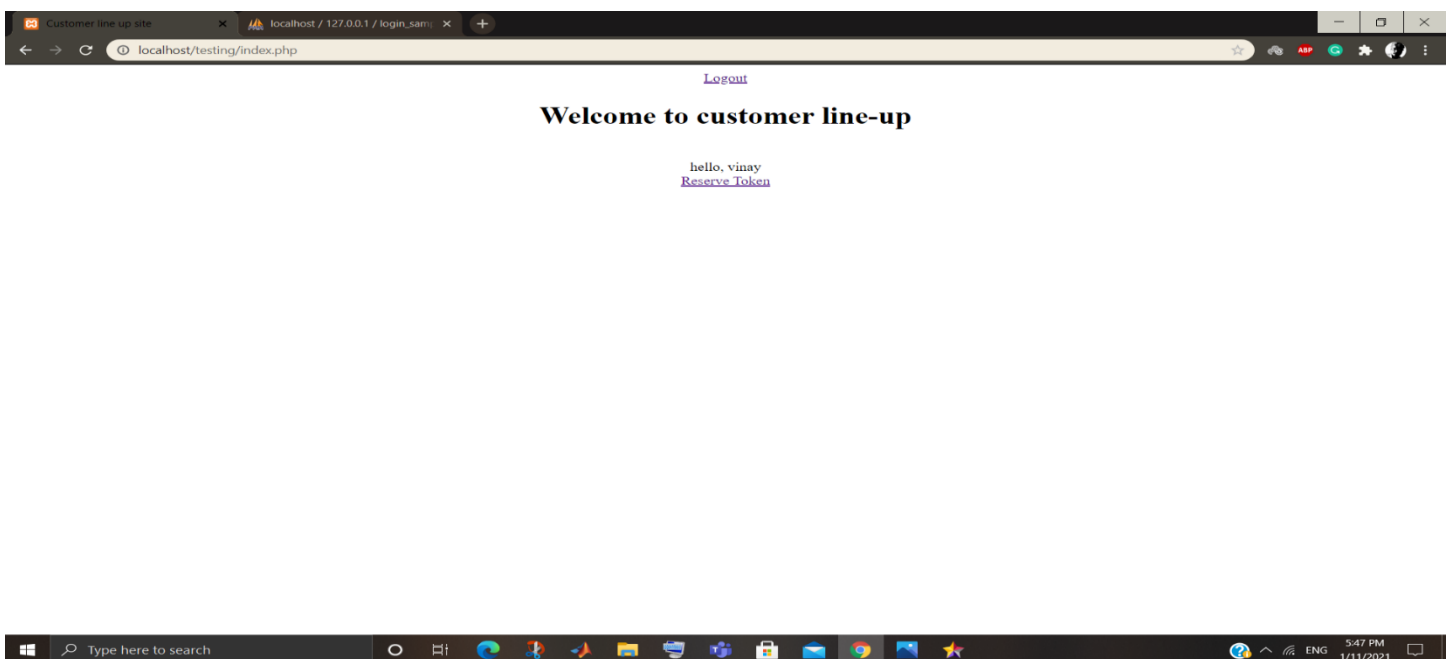


Figure 21 Testing third user homepage

- it shows the home page of vinay.

7.11. QR code page for third user:

generate QR (quick response) code

Supermarket store id for Esselunga: 1, for Coop: 2.

Enter your preferred date below, for example 01-01-2021, day-month-year format.

07-01-2021

Enter your expected time slot below, for example 11:00am-11:15am

10:00am-10:15am

Choose your preferred supermarket

☐ Esselunga
☒ Coop

Choose atleast one the category sections!

fruits vegetables
dairy products
stationery products
cleaning products

Get Token

You are required to fill all the fields.
[Go to home page](#)

Figure 22 Testing third user QR code generation

- *now the user chooses the coop as the supermarket.*
- *Generates the token.*

7.12.Token for third user:

localhost/testing/qrcode.php x localhost / 127.0.0.1 / login_sam x +

localhost/testing/qrcode.php

Enter your preferred date below. for example 01-01-2021. day-month-year format.

enter date here

Enter your expected time slot below. for example 11:00am-11:15am

enter time here

Choose your preferred supermarket

☐ Esselunga
☐ Coop

Choose atleast one the category sections!

fruits vegetab
dairy product:
stationery pro
cleaning prod

Get Token

Coop Token: cool

Monday,11th January 17:46

Good day! Take screenshot before leaving!

[Go to home page](#)

Type here to search

ENG 5:46 PM 1/11/2021

Figure 23 Testing third user Token generation and QR code

- *Now the token is like cool.*
- *He is the first user for the coop supermarket.*

8. Future Scope

- *The project is developed using an application that is capable to book a visit to available supermarket.*
- *Customer can also generate a QR code with token number on it.*
- *QR contains the data of wish list of sections which user would like to buy.*
- *The application should be improvised by adding a separate data base for admin to manage the traffic in real time in the future version release.*

9. Appendix

9.1. Code - Customer Lineup Homepage

```
<?php
session_start();
include("connection.php");
include("functions.php");
$user_data = check_login($con);
?>
<div align="center">
<!DOCTYPE html>
<html>
<head>
    <title>Customer line up site</title>
</head>
<body>

    <a href="logout.php">Logout</a>
    <h1>Welcome to customer line-up</h1>

    <br>
    hello, <?php echo $user_data['user_name']; ?>
    <br>
    <br>
    <a href="qrcode.php">Reserve Token</a>

</body>
</html>
</div>
```

9.2. Code - Connection

```
<?php

$dbhost = "localhost";
$dbuser = "root";
$dbpass = "";
$dbname = "login_sample_db";

if(!$con = mysqli_connect($dbhost,$dbuser,$dbpass,$dbname))
{
    die("failed to connect!");
}
```

9.3. Code - Functions

```
<?php

function check_login($con)
{
```

```

if(isset($_SESSION['user_id']))
{
    $id = $_SESSION['user_id'];
    $query = "select * from users where user_id = '$id' limit 1";

    $result = mysqli_query($con,$query);
    if($result && mysqli_num_rows($result) > 0)
    {
        $user_data = mysqli_fetch_assoc($result);
        return $user_data;
    }

    }
    // redirect to login
    header("Location login.php");
    die;
}

```

```

function random_num($length)
{
    $text = "";
    if($length < 5)
    {
        $length = 5;
    }

    $len = rand(4,$length);

    for ($i=0; $i < $len; $i++) {
        # code...
        $text .= rand(0,9);
    }

    return $text;
}

```

9.4. Code – Signup

```

<?php
session_start();
include("connection.php");
include("functions.php");

if($_SERVER['REQUEST_METHOD'] == "POST")
{
    // something was posted
    $user_name = $_POST['user_name'];
    $password = ($_POST['password']);

    if(!empty($user_name) && !empty($password) && !is_numeric($user_name))
    {
        //save to database
        $user_id = random_num(20);
        $query = "insert into users (user_id,user_name,password) values
        ('$user_id','$user_name','$password')";
    }
}

```

```

mysqli_query($con, $query);
header("Location: login.php");
die;
}else
{
echo "Please enter some valid information!";
}
}
?>
<!DOCTYPE html>
<html>
<head>
<title>Signup</title>
</head>
<body>
<style type="text/css">
#text{
height: 25px;
border-radius: 5px;
padding: 4px;
border: solid thin #aaa;
width: 100%;
}

#button{
padding: 10px;
width: 100px;
color: white;
background-color: lightblue;
border: none;
}

#box{
background-color: grey;
margin: auto;
width: 300px;
padding: 20px;
}

</style>
<div id = "box">
<form method = "post">
<div style="font-size: 20px;margin: 10px;color: white;">Signup</div>
<input id="text" type="text" name="user_name"><br><br>
<input id="text" type="password" name="password"><br><br>
<input id="button" type="submit" name="Signup"><br><br>
<a href="login.php">Click to Login</a><br><br>
</form>

</div>

</body>
</html>

```

9.5. Code - Login

```
<?php
session_start();
include("connection.php");
include("functions.php");

if($_SERVER['REQUEST_METHOD'] == "POST")
{
    // something was posted
    $user_name = $_POST['user_name'];
    $password = ($_POST['password']);

    if(!empty($user_name) && !empty($password) && !is_numeric($user_name))
    {
        //read from the database

        $query = "select *from users where user_name = '$user_name' limit 1";
        $result = mysqli_query($con, $query);

        if($result)
        {
            if($result && mysqli_num_rows($result) > 0)
            {
                $user_data = mysqli_fetch_assoc($result);

                if($user_data['password'] === $password)

                {
                    $_SESSION['user_id'] = $user_data['user_id'];
                    header("Location: index.php");
                    die;

                }
            }

            echo "wrong username or password!";
        }else
        {
            echo "wrong username or password!";
        }
    }
}

?>
<!/DOCTYPE html>
<html>
<head>
    <title>Login</title>
</head>
<body>
    <style type="text/css">
        #text{
            height: 25px;
```

```

border:radius: 5px;
padding: 4px;
border: solid thin #aaa;
width: 100%;
}

#button{
padding: 10px;
width: 100px;
color: white;
background-color: lightblue;
border: none;
}

#box{
background-color: grey;
margin: auto;
width: 300px;
padding: 20px;
}

```

```

</style>
<div id = "box">
<form method = "post">
<div style="font-size: 20px;margin: 10px;color: white;">Login</div>
<input id="text" type="text" name="user_name"><br><br>
<input id="text" type="password" name="password"><br><br>
<input id="button" type="submit" name="Login"><br><br>
<a href="signup.php">Click to Signup</a><br><br>
</form>

</div>

```

```

</body>
</html>

```

9.6. Code – QR Code

```

<div align="center">
<h1>generate QR (quick response) code</h1>
<!-- lets create qr in dynamic way -->
<form action="qrcode.php" method="post">
<p>Enter your preffered date below.
for example 01-01-2021. day-month-year format.</p>
<input type="text" name="date" placeholder="enter date here"/><br>
<p>Enter your expected time slot below.
for example 11:00am-11:15am</p>
<input type="text" name="time" placeholder="enter time here"/><br>
<p>Choose your preffered supermarket </p>
<input type="radio" name="supermarket" value="Esselunga"> Esselunga<br>
<input type="radio" name="supermarket" value="Coop"> Coop<br>
<p>Choose atleast one the category sections!</p>
<select multiple name="wishList[]" style="width:100px ;">
<option value = "Fruits_vegetables">fruits vegetables </option>
<option value = "Dairy_products">dairy products </option>

```

```

<option value = "Stationery_products">stationery products </option>
<option value = "Cleaning_products">cleaning products </option>
<option value = "Snacks_drinks">snacks drinks </option>
</select><br>
<input type="submit" name="btnsubmit" value="GENERATE"/><br>
</form>
<?php
    if(isset($_POST['btnsubmit'])&&(isset($_POST['wishList']))&&
(isset($_POST['supermarket'])) && (isset($_POST['date'])) && (isset($_POST['time'])) )
    {
        $date = trim($_POST['date']);
        $time = trim($_POST['time']);
        $supermarket = $_POST['supermarket'];
        $data = implode(", ", $_POST['wishList']);

        // here we have to display token number.
        echo "<h2> token number </h2>";

        // set timezone
date_default_timezone_set('Europe/Rome');
        // display the date and time with greetings
        function show_date(){
            return date('l, jS F H:i');
        }
        function greeting(){
            $hour = date('H');
            if($hour < 12){
                $greeting = "Good morning!";
            }
            else{
                $greeting = "Good day!";
            }
            return $greeting;
        }
        echo show_date();
        echo "<br/>". greeting();
        echo " <img src='https://chart.googleapis.com/chart?cht=qr&chs=150x150&chl= $date,
$time, $supermarket, $data' height=250 width=250/>";
    }else{
        echo "You are required to fill all the fields.";
    }
?>
<div align="center"></div>
<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<body>
<a href="index.php">Go to home page</a>

</body>
</html>
</div>

```

</div>

9.7. Code – Logout

<?php

```
session_start();
```

```
if(isset($_SESSION['user_id']))  
{  
    unset($_SESSION['user_id']);  
}
```

```
header("Location: login.php");  
die;
```

10. Project Reference

Score 2021

Project Proposal - CLup(Customers Lineup)

Project Sponsor - Matteo Rossi, Politecnico di Milano