Amr Khaled Gaber Abdelbaset (team leader)	**21100834**
Mohamed Talat Ahmed Elslamwy	**21100809**
Ahmed Mohamed Elmohamady	**21100821**
Ahmed Mohamed Dawood	**21100820**
Mohamed Ahmed Saad Ali	**21100871**

1-[20marks]List of user functional and nonfunctionalrequirements(make them in sperate lists). You must clearly separate between system services (i.e., functions and operations) and system rules (i.e., Must have rules and Shall not have rules). Make sure you do not write contradicting requirements, and you should use precise clear English for describing the requirements.

Services:

1. User Management Service
2. Photo Sharing Service
3. Activity Feed Service
4. Search Service
5. Reporting Service
6. Profile Service

Functional Requirements:

1.1. The User Management Service shall allow users to create a new account with a unique username and password.

1.2. The User Management Service shall allow users to log in to their account using their username and password.

1.3. The User Management Service shall allow users to edit their account information.

1.4. The User Management Service shall allow users to delete their account.

1.5. The User Management Service shall prevent multiple accounts with the same username.

1.6. The User Management Service shall store user passwords securely.

2.1. The Photo Sharing Service shall allow users to upload photos.

2.2. The Photo Sharing Service shall allow users to share photos with other users.

2.3. The Photo Sharing Service shall allow users to delete their own photos.

2.4. The Photo Sharing Service shall support different file types and sizes.

2.5. The Photo Sharing Service shall display the upload progress to the user.

2.6. The Photo Sharing Service shall allow users to comment on photos and posts made by other users.

3.1. The Activity Feed Service shall display a personalized feed of activity from the users the current user follows.

3.2. The Activity Feed Service shall display photos and comments made by the users the current user follows.

3.3. The Activity Feed Service shall display new follower notifications.

4.1. The Search Service shall allow users to search for other users by username.

4.2. The Search Service shall allow users to search for photos by keywords.

4.3. The Search Service shall allow users to search for comments by keywords.

5.1. The Reporting Service shall allow users to report inappropriate content or behavior.

5.2. The Reporting Service shall allow administrators to review reported content and act if necessary.

6.1. The Profile Service shall allow users to view their own profile, which displays their activity and personal information.

6.2.The Profile Service shall allow users to edit their own profile information.

6.3.The Profile Service shall display the number of followers and following for each user.

6.4.The Profile Service shall display the user's uploaded photos.

Non-functional Requirements:

7.1. The system must be available 24/7, with minimal downtime for maintenance.

7.2. The system must be able to handle a large number of users and user activity without slowing down.

7.3. The system must be secure and protect user data and personal information.

7.4. The system must be easy to use and navigate, with intuitive interfaces and clear instructions.

7.5. The system must be compatible with a variety of devices and platforms, including desktop and mobile.
7.6. The system must be accessible to users with disabilities, including visually impaired users.

7.7. The system must be scalable, allowing for future growth and expansion.

7.8. The system must be able to backup and restore data to prevent data loss in the event of a system failure.
7.9. The system must comply with relevant laws and regulations, including privacy laws and copyright laws.
7.10. The system must be customizable, allowing users to personalize their experience and settings.

- Non-functional requirement 7.2 (system must be able to handle many users and user activity without slowing down) relates to functional requirements 1.1, 1.2, 2.1, 2.6.

Rules:

1. Users must create a unique username and password when registering for an account (Functional requirement 1.1).
2. Users must log in using their registered username and password (Functional requirement 1.2).
3. Users must be able to edit their account information, such as their profile picture and bio (Functional requirement 1.3).
4. Users must be able to delete their own account if desired (Functional requirement 1.4).
5. Users cannot create multiple accounts with the same username (Functional requirement 1.5).
6. User passwords must be stored securely (Functional requirement 1.6).
7. Users can upload and share photos with other users (Functional requirement 2.1 and 2.2).

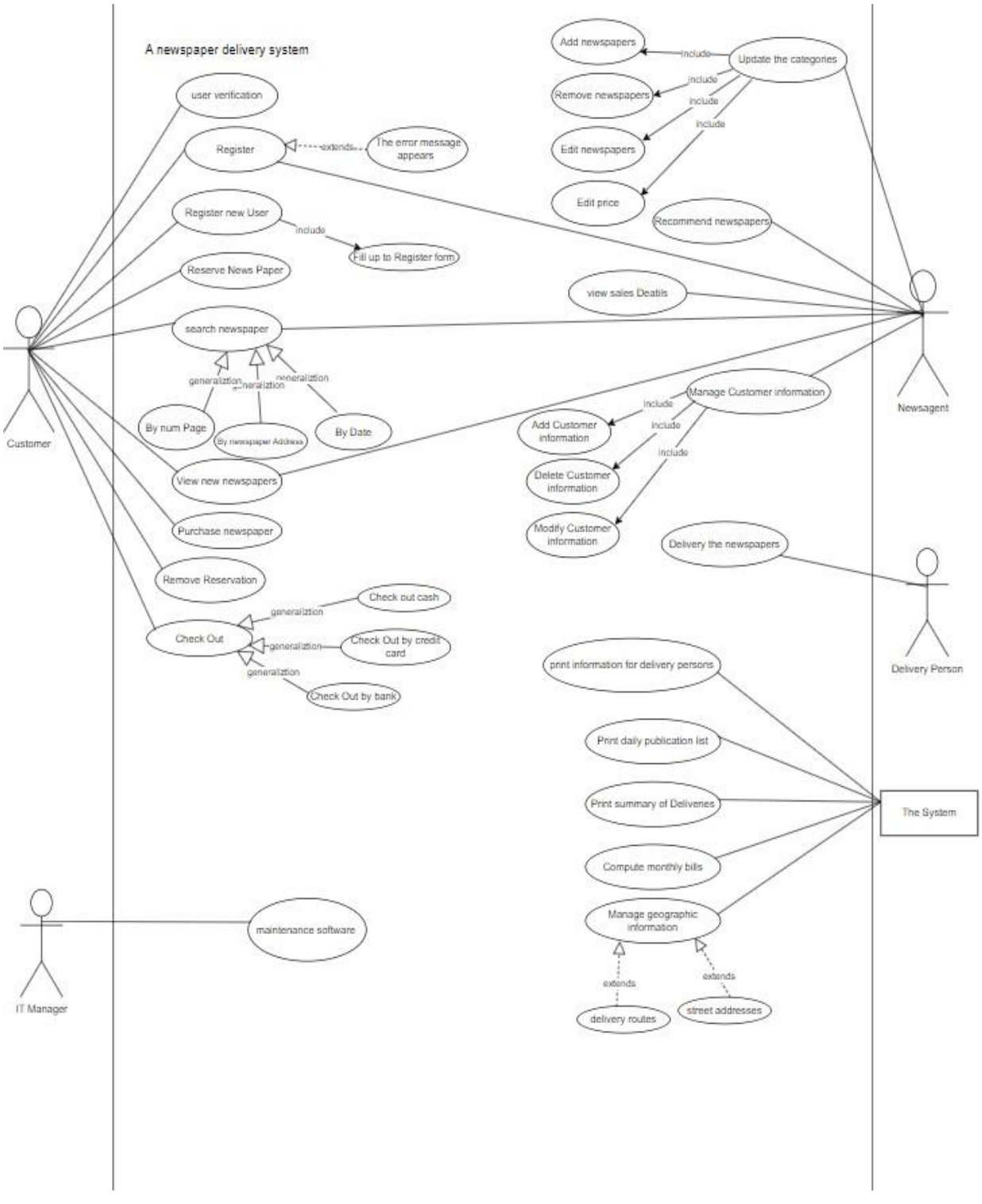
8. Users can delete their own photos (Functional requirement 2.3).
9. The system should support various file types and sizes for photo uploads (Functional requirement 2.4).

10. The system should display the upload progress to the user (Functional requirement 2.5).
11. Users can comment on photos and posts made by other users (Functional requirement 2.6).
12. The system should display a personalized feed of activity from the users the current user follows (Functional requirement 3.1).
13. The system should display photos and comments made by the users the current user follows (Functional requirement 3.2).
14. The system should notify users when they gain new followers (Functional requirement 3.3).
15. Users can search for other users by username (Functional requirement 4.1).
16. Users can search for photos and comments by keywords (Functional requirement 4.2 and 4.3).
17. Users can report inappropriate content or behavior (Functional requirement 5.1).
18. Administrators can review reported content and take action if necessary (Functional requirement 5.2).
19. Users can view their own profile, which displays their activity and personal information (Functional requirement 6.1).
20. Users can edit their own profile information (Functional requirement 6.2).
21. The system should display the number of followers and following for each user (Functional requirement 6.3).
22. The system should display the user's uploaded photos on their profile (Functional requirement 6.4).
23. The system should be available 24/7, with minimal downtime for maintenance (Non-functional requirement 7.1).
24. The system must be able to handle a large number of users and user activity without slowing down (Non-functional requirement 7.2).
25. User data and personal information must be stored securely (Non-functional requirement 7.3).
26. The system should be easy to use and navigate, with intuitive interfaces and clear instructions (Non-functional requirement 7.4).
27. The system should be compatible with a variety of devices and platforms, including desktop and mobile (Non-functional requirement 7.5).
28. The system should be accessible to users with disabilities, including visually impaired users (Non-functional requirement 7.6).
29. The system should be scalable, allowing for future growth and expansion (Non-functional requirement 7.7).
30. The system should be able to backup and restore data to prevent data loss in the event of a system failure (Non-functional requirement 7.8).
31. The system must comply with relevant laws and regulations, including privacy laws and copyright laws (Non-functional requirement 7.9).

32. Users should be able to personalize their experience and settings (Non-functionalrequirement 7.10).

2-[30marks]Structured system use cases. Make sure you start with the core use cases of the system. Clearly, indicate your assumption when you use extension, inclusion, and generalization, and then for each use case, specify the corresponding scenario.For use case descriptions, every team member must provide one complete use case description. No need to describe all use cases, just focus on the main ones.

Use Case Digram



use case descriptions

Use Case name	News Paper Purchase
Trigger	Members send an order form to newsagent
Actors	Customers,Newsagent
Description	Allows the Customer to purchase News Paper
Preconditions	1- Customers applies for a purchase a book 2- Newsagent is accepting orders

Use Case name	Compute customer bill
Trigger	The system can Compute customer bill
Actors	Customers,System
Description	The system should automatically compute monthly bills for each customer
Preconditions	The system has customer informationand publication delivery data for the month

Use Case name	Manage Customer Information
Trigger	The Newsagent can Manage Customer Information
Actors	Customers ,Newsagent
Description	The system should allow the newsagent to add, modify, or delete customer information, including delivery preferences and billing information.
Preconditions	Newsagent is logged into the system

Use Case name	Print Summary Report
Trigger	The System can Print Summary Report
Actors	Newsagent, System
Description	The system should generate a summary of who received what publications each day for the newsagent.
Preconditions	Newsagent is logged into the system

Use Case name	Print Delivery List
Trigger	The System can Print Delivery List and the delivery person can view this
Actors	Delivery Person, System
Description	The system prints a list of publications tobe delivered to each address for each delivery person every day.
Preconditions	Delivery person is logged into the system

Scenario: Print Delivery List

Actors: Delivery Person, System

Preconditions: Delivery person is logged into the system

Actor Intentions/ Main scenario:

.The delivery person selects the "Print Delivery List" use case

.The system retrieves the delivery schedule for the current day

.The system generates a list of publications to be delivered to each address

.The system prints the list for the delivery person

Alternate Flow: If a delivery person goes to the Customer and he is not available in his address or the address is wrong , the newsagent or delivery person can mark the customer as "on hold" in the system, and the system will not print the publications for
.that customer during that time

Scenario: Print Summary Report

Actors: Newsagent, System

Preconditions: Newsagent is logged into the system

Actor Intentions/ Main scenario:

.The newsagent selects the "Print Summary Report" use case

.The system retrieves the delivery history for the selected date range

The system generates a summary report showing who received what publications each
.day

.The system prints the report for the newsagent

Alternate Flow: None

Scenario: Compute Customer Bill

Actors: Newsagent, System

Preconditions: Newsagent is logged into the system

Actor Intentions/ Main scenario:

.The newsagent selects the "Compute Customer Bill" use case

.The system retrieves the delivery history for the selected month

The system computes the bill for each customer based on their subscription and delivery .frequency

.The system generates a bill for each customer

The system prints the bills for the newsagent to deliver to customers along with their .newspapers

Alternate Flow: If a customer disputes their bill or has a billing question, the newsagentcan access the customer's billing information in the system and address the issue.

Scenario: Manage Customer Information

Actors: Newsagent, System

Preconditions: Newsagent is logged into the system

Actor Intentions/ Main scenario:

.The newsagent selects the "Manage Customer Information" use case

.The system displays a list of current customers

The newsagent can add, remove, or update customer information such as delivery .address or delivery frequency

.The system saves the changes

Alternate Flow: If a customer adds a new publication to their regular delivery, the newsagent or delivery person updates the customer's information in the system to .include the new publication

Scenario: Manage Geographic Information

Actors: Newsagent, System

Preconditions: Newsagent is logged into the system

Actor Intentions/ Main scenario:

.The newsagent selects the "Manage Geographic Information" use case

The system allows the newsagent to input geographic information such as street names .and house numbers

The system uses the information to print the delivery list for each delivery person in the .order in which publications are

deliveredAlternate Flow: None

Scenario: News Paper Purchase

Actors: Newsagent, Customer

Preconditions: Newsagent and Customer is logged into the

systemCustomer view the News Papers

Actor Intentions/ Main scenario:

The Customer View all News Paper and make a

purchaseThe newsagent View purchase process and

confirms it The system displays a information of a

purchase process.

Alternate Flow: If a publication is out of stock or unavailable, the newsagent or delivery person can mark it as such in the system and make a note to follow up with the customerlater.

If a customer requests a specific delivery time, the newsagent or delivery person can notethe requested time in the system and adjust the delivery route accordingly.

3-[20marks]Identified subsystems, their functionalities and required interfaces.Also show the traceability matrix between the requirements and the subsystems.

Interface definition:

Interface 1: registration management system

- Public void create_acc(String name,string pass);
- Public data[] login(String name,string pass);
- Public void Edit_info(String name,string pass);
- public Boolean deleteInformation(String name,string pass);
- public Boolean check_redundancy(String name,string pass);

Interface 2: newspaper management service

- public void reserve_newsp(newspaper spec);
- public data[] search();
- public void view_products(newspaper spec);
- public void purchase_products(newspaper newpaper,payment payment);
- public data[] delete_reserve(newspaper newspaper);

Interface 3: payment interface

- public void payment_With_CreditCard ()
- public void validatePINNumber()
- public void paymentCash ()

Interface 4: newsagent management system

- public data[] add _newspaper(newspaper newspaper);
- public void remove_newspaper(newspaper newspaper);
- public void edit_newspaper(newspaper newspaper);

- public void edit_price();

Interface 5: manage customer service.

- public data[] add _customer_info();
- public void delete _customer_info();
- public data[] modfiy _customer_info();

Interface 6: delivery management service.

- public void print _info();
- public void compute_bills();

Interface 7: geographic management service

- public string address_info()

number	**Use case/system**	**geographic management service**	**delivery management service.**	**newsagent management system**	**manage customer service.**	**registration management system**
1	Getting living data information	X	X			X
2	Helping maintenance			X		X
3	Search for newspaper			X	X	

4	checkout		X	X	X	
5	Create roles				X	
6	Add newspaper.				X	
7	reservation			X	X	X
8	Print information	X	X	X		
9	registration				X	X
10	Manage geographic info	X	X			

number	Use case/system	Costumer communication subsystem	Tracking of costumer subsystem	Database subsystem	Generate Statistics subsystem	
1	Getting living data information	X	X	X		
2	Helping maintenance			X	X	
3	Search for newspaper	X		X		
4	checkout	X				
5	Create roles	X				
6	Add newspaper.			X		
7	reservation	X		X	X	
8	Print information					
9	registration	X				
10	Manage geographic info	X		X	X	

4-[20marks]Test requirements for each use case. Each team member provide the test requirements for one core use case.

Test Requirements for the “Register Customer” use case:

1. Registration Form Validation: Verify that all required fields (such as name, email, phone number, address, etc.) are validated properly before submitting the registration form. Test that the form should not allow the submission of incomplete or incorrect data.
2. Unique Email Address: Verify that each customer registering with the system must have a unique email address, i.e., no two customers can have the same email address.
3. Password Strength: Verify that the password entered during registration should be at least 8 characters long and include a combination of uppercase, lowercase, numbers and special characters.
4. Email Notification: Verify that the customer receives a confirmation email after registration with the necessary details such as name, email, phone number, address, and other relevant information.
5. Login Authentication: Verify that a customer can only log in with the correct email and password they used during registration.
6. User Profile Information: Verify that the customer profile information is saved and displayed correctly after registration, including name, email, phone number, and address.
7. Database Integration: Verify that the customer’s information is saved in the system’s database correctly and can be retrieved or updated as needed.
8. Security and Privacy: Verify that the registration process is secure and that customer data is protected, such as using encryption and following industry-standard security protocols.
9. Scalability: Verify that the registration process can handle a large number of users registering at the same time and still perform without any lag or downtime.
10. Usability: Verify that the registration process is intuitive and easy to use for all types of users, with clear instructions and error messages provided where necessary.

test requirements for a “search” feature in a newspaper app:

1- The search feature should allow users to search for keywords and phrases within the app’s database of news articles.
2- The search results should be presented in a clear and organized manner, with relevant articles displayed at the top of the list.
3- The search feature should be able to handle a large volume of requests without slowing down or crashing the app.
4- Users should be able to refine their search results using filters such as date, category, and author.
5- The search feature should support fuzzy matching, so that users can still find relevant articles even if they misspell a keyword or phrase.
6- The search feature should be easily accessible from any screen within the app, with a prominent search bar visible at all times.
7- The search feature should be secure, with user data protected against unauthorized access or hacking attempts.
8- The search feature should be intuitive and easy to use, with a minimal learning curve for new users.
9- The search feature should be optimized for speed and responsiveness, with quick loading times and minimal lag between searches.
10- The search feature should be reliable, with minimal downtime or errors even during periods of heavy usage.

11- The search feature should be scalable, able to handle an increasing number of users and search requests as the app's popularity grows.
12- The search feature should be compatible with a wide range of devices and operating systems, including both iOS and Android.

test requirements for "customer purchase newspaper" in a newspaper app:

1- Payment Integration Test: Verify that the payment gateway is working fine and users are able to make successful payments for the newspaper.
2- Subscription Test: Ensure that users are able to subscribe to a newspaper and receive daily/weekly/monthly editions as per their subscription.
3- Notification Test: Verify that users receive notifications about new editions, subscription expiry, and other important events.
4- Account Management Test: Ensure that users are able to manage their account details like name, email, phone number, and subscription details.
5- Content Delivery Test: Verify that users are able to view and access the newspaper content without any issues or delays.
6- Offline Access Test: Test that users are able to download and access the newspaper even when they are offline.
7- Compatibility Test: Verify that the newspaper app is compatible with different devices, operating systems, and browsers.
8- Performance Test: Test the app's performance under different network conditions and ensure that it loads quickly and smoothly.
9- Security Test: Verify that the app has proper security measures in place to protect users' personal and payment information.
10- Usability Test: Evaluate the app's user interface and ensure that it is user-friendly and easy to navigate.

test requirements for "newsagent add newspaper" in a newspaper app:

1- Verification of Newspaper Information: he system should allow the newsagent to enter the newspaper's name, publisher, price, publication date, and any other relevant information. Verify that all the entered information is correctly saved and displayed in the app.

2- Validation of Unique Newspaper Name: The system should validate that the newspaper name entered by the newsagent is unique and not already present in the system. Verify that the app returns an error message if the name is already in use.

3- Verification of Newspaper Availability: The system should allow the newsagent to mark the newspaper as available or unavailable for sale . Verify that the newspaper's availability status is correctly updated and displayed in the app.

4- Verification of Adding Newspaper Image: The system should allow the newsagent to add an image of the newspaper cover to make it more attractive to customers. Verify that the image is correctly uploaded and displayed in the app.

5- Verification of Newspaper Price Calculation: The system should calculate the newspaper price based on the publisher's price and any additional markup set by the newsagent. Verify that the price calculation is accurate and displayed correctly in the app.

6- Verification of Newspaper Publication Date: The system should verify that the publication date entered by the newsagent is valid and not in the future. Verify that the app returns an error message if an invalid date is entered.

7- Verification of Newspaper Category: The system should allow the newsagent to select the newspaper category (e.g., Sports, Business, Entertainment) for the newspaper. Verify that the selected category is correctly displayed in the app and can be used for filtering/searching.

8- Verification of Adding Newspaper Quantity: The system should allow the newsagent to add the quantity of newspapers available for sale. Verify that the quantity is correctly displayed in the app and updated as the newspapers are sold.

9- Verification of Adding Newspaper to Inventory: The system should add the newspaper to the inventory once it is added by the newsagent. Verify that the newspaper is correctly added to the inventory and available for sale in the app.

10- Verification of Access Control: The system should restrict access to the "newsagent add newspaper" use case to authorized personnel only.Verify that unauthorized users cannot access the use case or perform any actions related to adding newspapers.

test requirements for the "newsagent edit price" in a newspaper app:

1- User interface: The user interface for the "newsagent edit price" feature should be user-friendly and easy to navigate. The interface should allow users to select the newspaper they want to edit the price for, and display the current price of the newspaper.

2- Editing functionality: The app should allow the user to edit the price of the newspaper. The user should be able to enter a new price in a designated field or use a slider to adjust the price.

3- Validations: The app should validate the price entered by the user. The price should be within a specified range and should be in the correct format.

4- Saving functionality: After the user has entered a new price, the app should save the new price and display it to the user. The app should also save the new price to the database and ensure that it is correctly associated with the newspaper.

5- Integration testing: The app should be tested to ensure that it integrates with other parts of the system correctly. For example, the app should ensure that the new price is correctly displayed in the newspaper's details page.

6- Security: The app should be tested to ensure that only authorized users can access the "newsagent edit price" feature. The app should also be tested to ensure that user data is protected and that there are no vulnerabilities in the system.

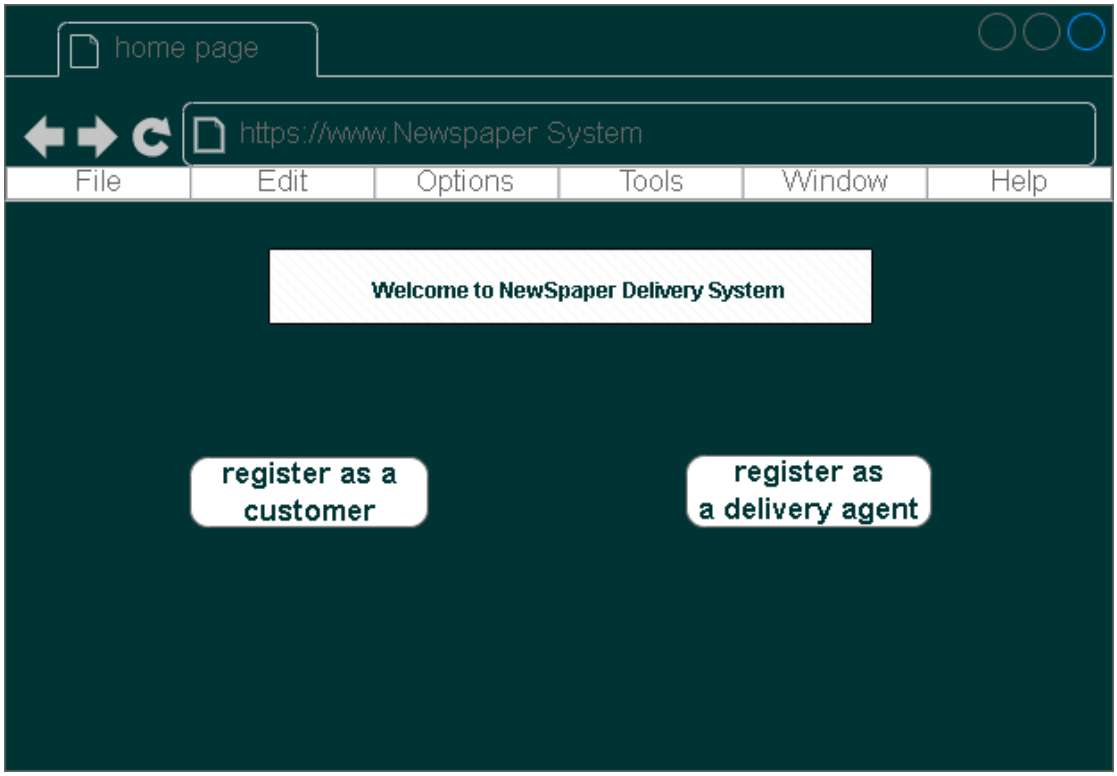
7- Performance testing: The app should be tested to ensure that it can handle a large number of simultaneous users editing prices without slowing down or crashing.

8- Usability testing: The app should be tested to ensure that it is easy to use and that users can complete the task of editing the price of a newspaper quickly and efficiently.

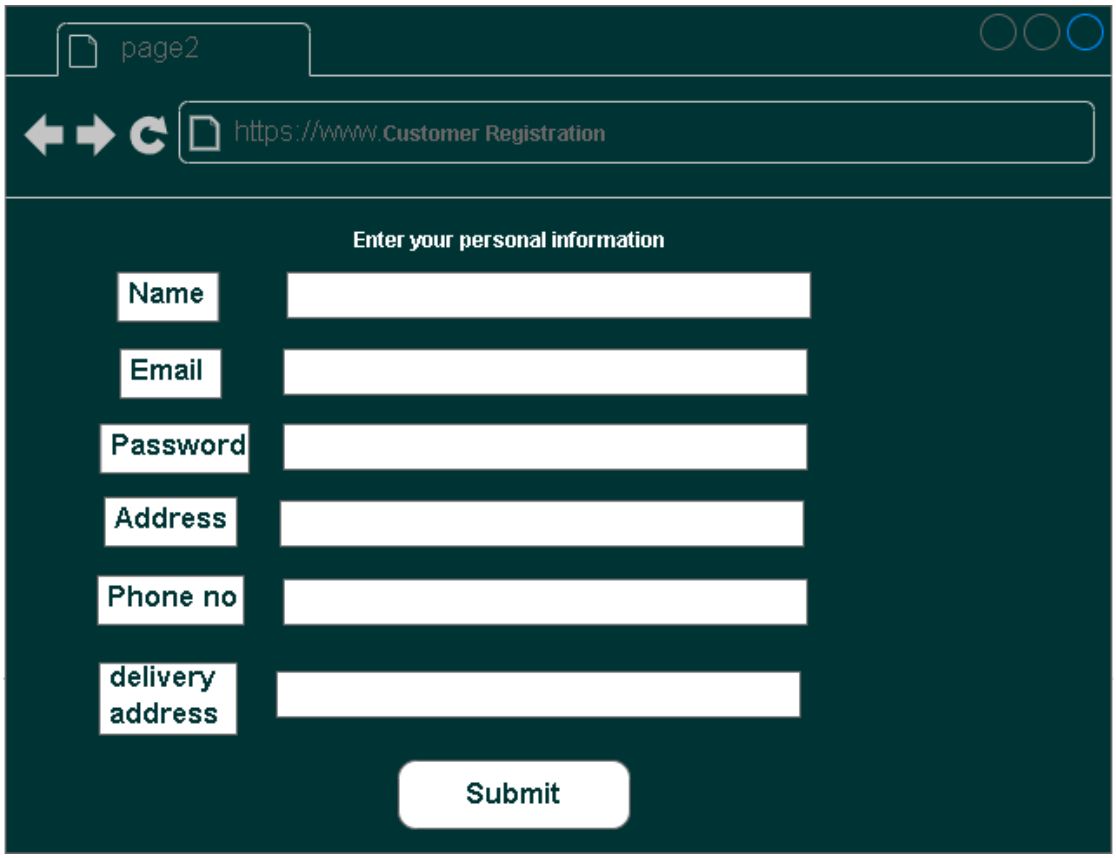
9- Accessibility testing: The app should be tested to ensure that it is accessible to users with disabilities. The app should comply with accessibility standards and guidelines, and be compatible with assistive technologies such as screen readers and voice commands.

10- Compatibility testing: The app should be tested to ensure that it is compatible with different devices, operating systems, and web browsers. The app should also be tested to ensure that it works correctly with different screen sizes and resolutions.

5-[10 marks] A draft for the user manual including the expected GUI.

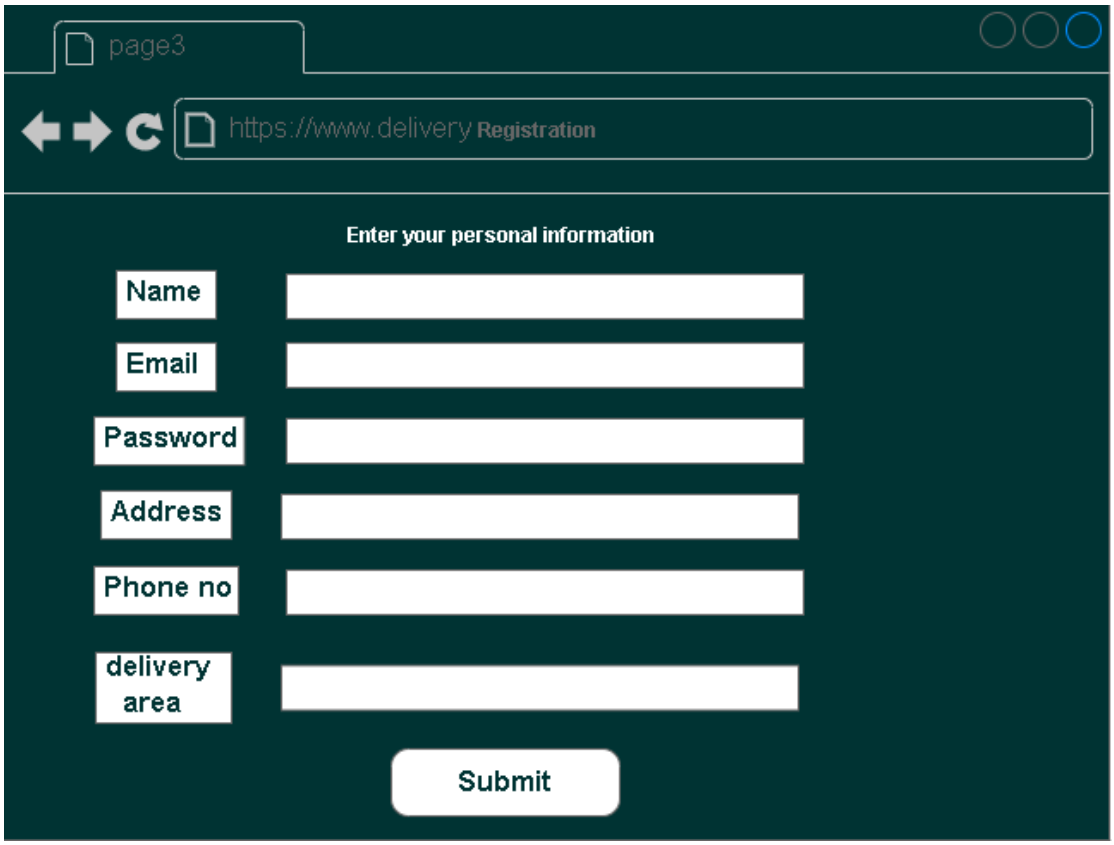


To use the Newspaper Delivery System, you must first register as a customer or delivery agent. Registration can be done via the system's user interface



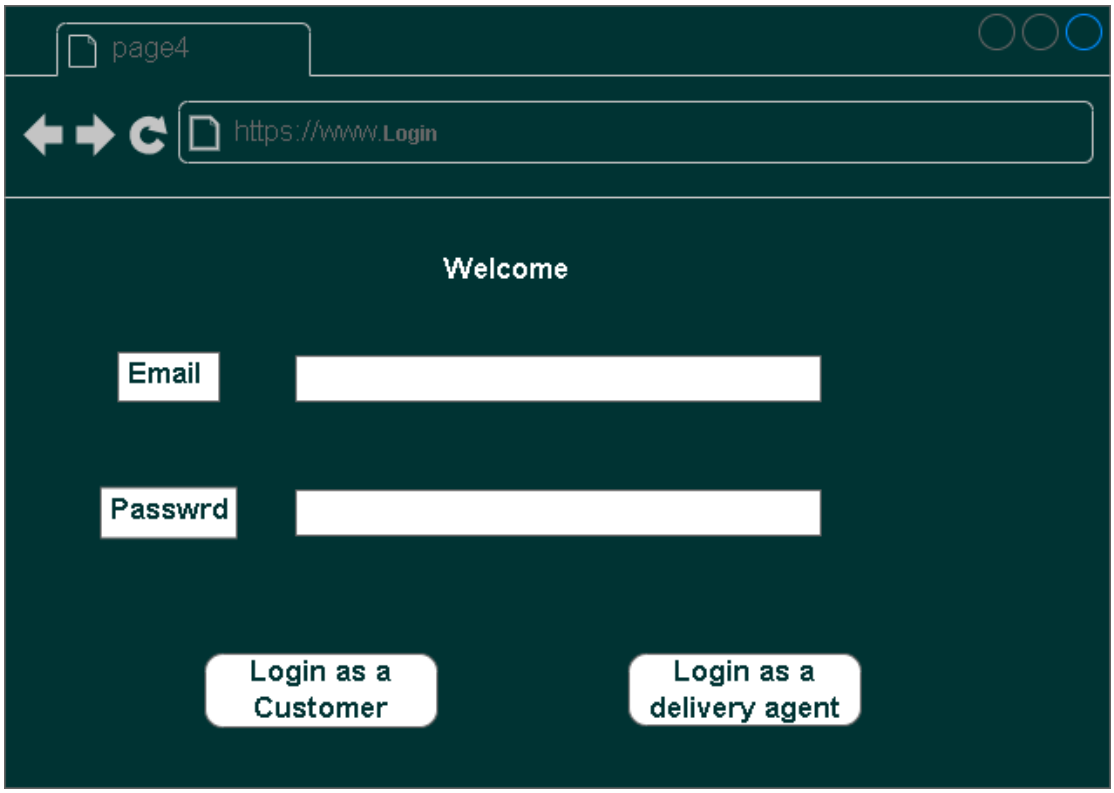
Customer Registration

To register as a customer, click on the "Register" button on the homepage. You will be asked to provide your personal information, including your name, email address, phone number, and delivery address. Once you have provided all the necessary information, click on the "Submit" button to complete the registration process



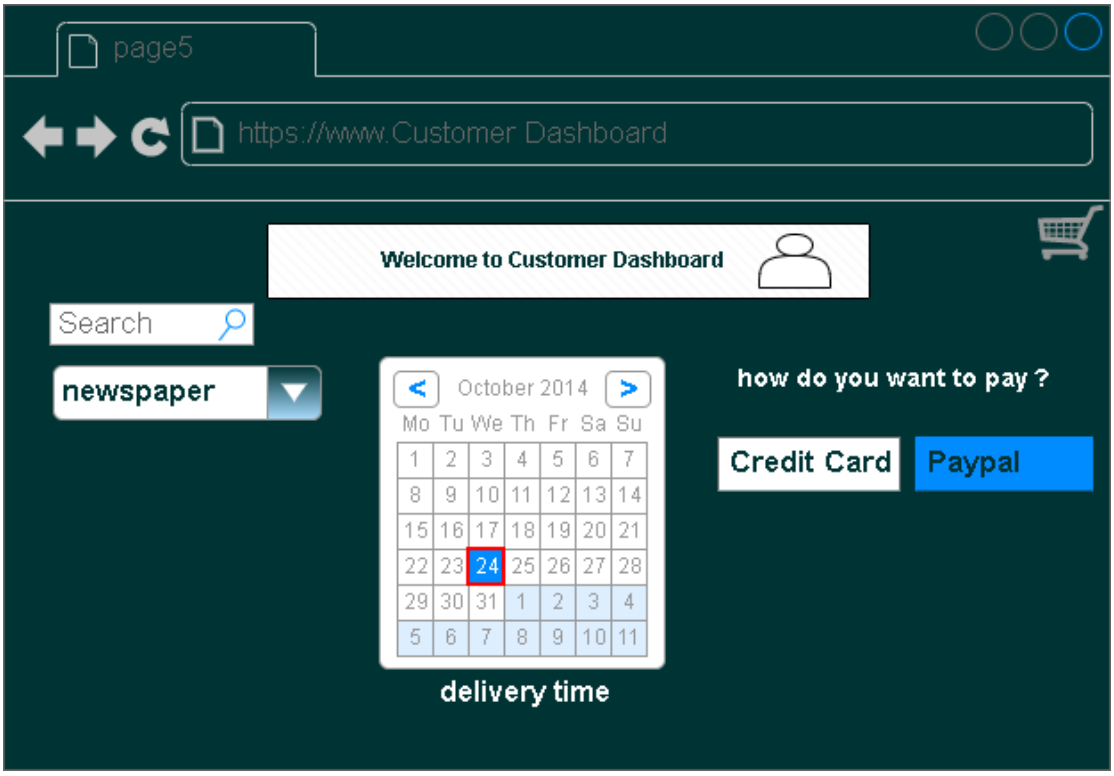
Delivery Agent Registration

To register as a delivery agent, click on the "Register" button on the homepage. You will be asked to provide your personal information, including your name, email address, phone number, and delivery area. Once you have provided all the necessary information, click on the "Submit" button to complete the registration process. You will be notified via email once your registration is approved by the system administrator.



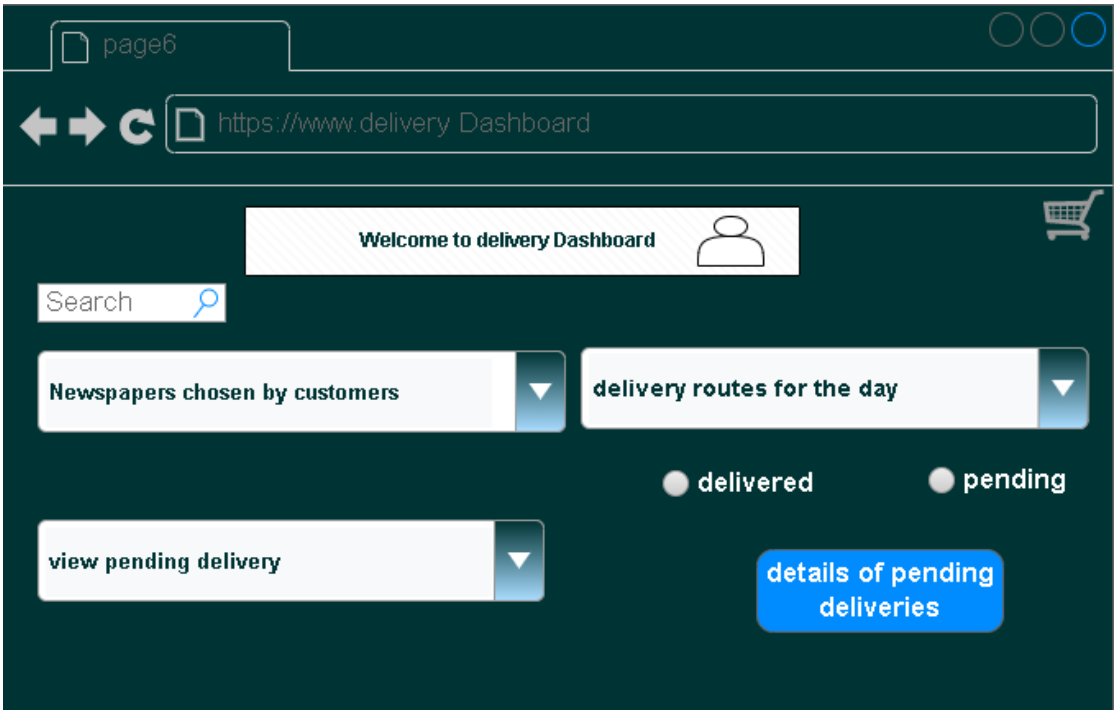
. You can now log in to the system and view delivery routes and pending deliveries.

. You can now log in to the system and place orders for newspapers.



Placing an Order To place an order

log in to the system using your registered email and password. Once you are logged in, you will be directed to the customer dashboard. From here, you can select the newspaper you want to order and choose your delivery preferences, such as the delivery time. You will also be asked to provide your payment information to complete the transaction. Once the order is placed, you will receive a confirmation email with the details of your order.



Viewing Delivery Routes and Pending Deliveries

To view delivery routes and pending deliveries, log in to the system using your registered email and password. Once you are logged in, you will be directed to the delivery agent dashboard. From here, you can view the delivery routes for the day, as well as the details of pending deliveries. You can also update the status of each delivery, such as marking it as "delivered" or "pending."