



Faculty of Engineering & Technology
Electrical & Computer Engineering Department

SOFTWARE ENGINEERING – COMP433

Final Project Report



InkLink
NEWSAGENT SHOP

Prepared by Group 5:

Hatem Hussein 1200894

Mohammad Abu Shams 1200549

Joud Hijaz 1200342

Instructor: Dr. Samer Zein

Section: 1

Date: 20-06-2024



Phase 1 – Business Statement

Newsagent Shop offers daily newspaper subscriptions with home delivery. Our newspaper business aims to provide daily newspapers to customers while providing an efficient home delivery service to ensure customer convenience and save their time while staying informed of the latest news and developments.

Our store provides many services to customers, including newspaper selection. We offer a varied selection of newspapers, catering to various interests and preferences. Customers can select from a variety of titles, including local news and international headlines. Subscription Options: We provide a seamless and easy subscription service for customers based on their lifestyle. Home Delivery, We provide customers with the service of delivering selected newspapers to their homes every morning, all the while maintaining accuracy and timeliness without any problems. Application services allow customers to manage their subscriptions, favorites, payment method, and fill out personal information from the home's website, phone number, etc., as well as determine whether to renew or cancel the subscription, modify the types of newspapers they want at any time, update billing information, or easily explore new publications. And the last service is subscription management. Customers receive assistance and guidance from store employees or through our remote application.

Customers of the Newsagent shop range from 100 to 200 every month, depending on demand. The clientele is diverse and includes both individuals and businesses. The store serves a variety of interests and languages by providing a wide selection of newspapers, including both national (about seven categories) and international periodicals. The shop's monthly sales volume is usually rather high, with an average of 500 to 1000 newspapers sold, depending on many factors such as geographic location, customer traffic, and subscriber uptake.

The customer can choose from our daily newspapers that we offer; they can select either from the store or online. We have different subscription plans that customers can opt for, which typically offer discounts and home delivery. The customer can choose the duration of the subscription (one day, one month, six months, or one year). The longer the subscription period, the greater the discount percentage. Our shop can accept different payment methods (cash and card). For the online customer, they pay upfront for the entire subscription period. Our shop arranges for the daily newspapers to be delivered to the customer's specified address at a specific time. The shop generates revenue from the sales of newspapers, both in-store and through subscriptions. The subscription model offers a steady stream of income as customers pay for a series of deliveries in advance. We also generate revenue through advertising partnerships and supplementary product sales.

Phase 2 – Requirements Elicitation

2.1 User Requirements

Listing all user requirements in the system:

UR1. The system shall allow users to sign up (creating new accounts) and sign in into their accounts online.

UR2. The system shall enable users to search for newspapers and magazines and view their availability.

UR3. The system shall support flexible newspaper ordering options, including subscription-based orders (Weekly, Monthly, Yearly) and one-time orders (Home Delivery, In-Store Pickup).

UR4. The system shall enable workers to manage newspaper and magazine inventory.

UR5. The system shall generate comprehensive sales reports and analytics.

UR6. The system shall offer multiple payment methods with generating receipts.

UR7. The system shall handle refunds and confirm them to customers promptly.

UR8. The system shall support a customer loyalty program that tracks points and rewards customers based on their purchases.

2.2 System Requirements

Describing each user requirement with its corresponding system requirements:

UR1. The system shall allow users to sign up (creating new accounts) and sign in into their accounts online.

1.1 Users shall sign up by entering personal details (ID, name, phone number, and birthdate), and the system will send an email to verify their email address.

1.2 Users shall log in with their email and password, which the system will verify against existing database entries.

1.3 If users forget their password, the system shall send a secure link to their registered email to reset it within 5 minutes on average. This process is secure to protect user data. Password reset process shall be completed within 5 minutes on average.

1.4 After a successful login, the system shall create a secure session, which will automatically end if the user is inactive for 30 minutes to protect their account.

UR2. The system shall enable users to search for newspapers and magazines and view their availability.

2.1 Users shall search using keywords, with results based on title, author, or topic.

2.2 The system shall display the current stock status (in stock/out of stock) in search results.

2.3 Users shall filter and sort results by date, popularity, and availability.

2.4 Selecting an item shall display details which are a brief description, publication frequency, and pricing.

2.5 The system shall process and display search results within 1 second on average, which is the time taken by the system to process a user's search query and display the search results.

UR3. The system shall support flexible newspaper ordering options, including subscription-based orders (Weekly, Monthly, Yearly) and one-time orders (Home Delivery, In-Store Pickup).

3.1 Users shall subscribe to newspapers with options for delivery frequency (daily, weekly, monthly, yearly) and manage subscriptions through their account.

3.2 Users shall place one-time orders with options for home delivery or in-store pickup.

3.3 Users shall change delivery frequency for subscriptions through their profile.

3.4 The system shall send email confirmations for orders and reminders for upcoming deliveries or subscription renewals.

3.5 Subscriptions shall auto-renew unless canceled by the customer, with a reminder sent three days before renewal.

UR4. The system shall enable workers to manage newspaper and magazine inventory.

4.1 Workers shall have an interface to update stock amounts, add, or remove items.

4.2 Inventory data shall update in real-time with sales and new stock entries. This helps keep inventory accurate and avoid over selling.

4.3 The system shall generate detailed reports on inventory and sales trends for workers to use. Reports should be generated within 5 seconds on average to provide timely insights for workers.

4.4 The system shall provide alerts for low stock levels to help workers reorder in time. Alerts should be issued within 1 minute of reaching the low stock threshold to allow workers sufficient time to reorder.

UR5. The system shall generate comprehensive sales reports and analytics.

5.1 The system shall create sales reports at specified intervals (daily, weekly, monthly, yearly), detailing total sales, sales by category, and individual item performance.

5.2 The system shall provide an analytical dashboard showing real-time data and trends for authorized users.

5.3 The system shall export sales data and reports in formats like CSV and PDF.

5.4 The system shall track and report metrics which are customer retention, average transaction value, and sales growth.

UR6. The system shall offer multiple payment methods with generating receipts.

6.1 The system shall support payments via credit/debit cards, digital wallets (PayPal, Apple Pay), and cash for in-store purchases.

6.2 The system shall generate digital receipts for transactions, printable in-store or emailable, detailing items purchased, total cost, payment method, and transaction date.

6.3 The system shall provide secure and encrypted payment processing for all methods using SSL/TLS encryption, tokenization, and PCI DSS compliance. It will also implement multi-factor authentication and undergo regular security audits.

6.4 The system shall adhere to security standards including PCI DSS, ISO/IEC 27001, GDPR, SOC 2, OWASP guidelines, and the NIST Cybersecurity Framework. It shall also undergo regular security audits, including internal and external audits, penetration testing, and automated vulnerability assessments.

UR7. The system shall handle refunds and confirm them to customers promptly.

7.1 The system shall process refunds through any supported payment method, including initiation, eligibility check, processing, and updating inventory and financial records.

7.2 The system shall generate a receipt for each refund, detailing refunded items, original transaction details, refund amount, payment method, and refund date, printable in-store or emailable.

7.3 The application shall allow order cancellations within 30 minutes of order placement.

7.4 The application shall not accept product returns after delivery, as the cancellation period will have expired.

UR8. The system shall support a customer loyalty program that tracks points and rewards customers based on their purchases.

8.1 Customers shall join the loyalty program online or in-store with staff assistance.

8.2 The system shall automatically award points based on spending, tracked in real-time and added to customer accounts after purchases.

8.3 Customers shall view and redeem points for rewards or discounts online or in-store.

8.4 Administrators shall manage loyalty program rules (point rates, reward levels, point expiration) with real-time updates shared with customers.

8.5 The system shall generate reports on the loyalty program, including total points awarded, redeemed, and remaining, and participation rates

2.3 Cost and Effort Estimation The whole method has been changed → Function Points

2.3.1 Assumptions

- One developer working full-time (100%) has a monthly salary.
- The cost calculation includes the base salary and a percentage of profit.

2.3.2 Effort Estimation Schedule

UR	Function Points	Effort (*20)	# of Days	Notes
UR1	2	$2*20 = 40$	5	Account management system
UR2	3	$3*20 = 60$	8	Search functionality with availability status
UR3	4	$4*20 = 80$	10	Flexible subscription and ordering options
UR4	4	$4*20 = 80$	10	Inventory management for staff
UR5	3	$3*20 = 60$	8	Sales reporting and analytics system
UR6	4	$4*20 = 80$	10	Payment processing with multiple methods
UR7	4	$4*20 = 80$	10	Refund management and immediate confirmation
UR8	3	$3*20 = 60$	8	Customer loyalty program
Total	27	540	69	

2.3.3 Costs Calculation

- **Total Effort:**

→ Total Effort in Days = 69

- **Developer Effort:**

→ 1 Developer: 69 Days / 22 Days per Month = 3.14 months \approx 4 months

→ 2 Developers: 69 Days / (22 Days per Month \times 2) = 1.57 months \approx 2 months

- **Salaries:**

→ Monthly salary for Developer 1: \$1400

→ Monthly salary for Developer 2: \$1600

→ Total Monthly Salary for 2 Developers: \$1400 + \$1600 = \$3000

- **Total Salary Cost:**

→ For 2 Developers: 2 \times 3000 = \$6000 for two months

- **Cost with Profit Margin:**

- Minimum Offer (30% profit):

→ 2 Developers: 6000 \times 1.3 = **\$7800**

- Maximum Offer (50% profit):

→ 2 Developers: 6000 \times 1.5 = **\$9000**

Phase 3 - Scenario, Actors, Use Cases, and Activity Diagram Development

3.1 Scenario Analysis

Scenario 1: Subscription Management *Written By Joud Hijaz - 1200342*

Normal:

A customer enters their username and password to log into the newsagent app. After logging in successfully, they go to the "My Subscriptions" section from the dashboard. The customer chooses a newspaper from their subscription list and adjusts the delivery frequency (daily, weekly, or monthly) to their liking. Upon clicking "Save Changes," the system updates the subscription details in the database. A "Your subscription has been updated successfully" message appears on the screen, and the customer then gets an email notification with the updated subscription details.

Alternative:

The customer logs into the app, navigates to the "My Subscriptions" section, and opts to cancel a subscription by clicking the "Cancel Subscription" button for the chosen newspaper. After confirming in the dialog, the system cancels the subscription, updates the database, and removes it from their active list. A "Your subscription has been canceled" message displays, and the customer gets an email confirming the cancellation and any related adjustments, such as refunds.

Error:

The customer logs into the app and experiences a delay while accessing the "My Subscriptions" page. They try to modify the delivery frequency of their subscription and click "Save Changes," but a server timeout causes a failure in processing the request. An error message "Failed to update your subscription due to a system error. Please try again later" appears. The subscription remains unaltered at its previous settings, and the customer doesn't receive any email notification about the attempted change.

Scenario 2: Online Payment and Receipt Generation

Written By Mohammad Abu Shams - 1200549

Normal Flow:

A customer chooses a newspaper, places it in their shopping cart, and moves to the checkout page. They are given various payment options such as credit cards, PayPal, and Apple Pay. After choosing their desired payment method, they input the necessary payment information. The transaction is then securely processed using an encrypted system. Following a successful transaction, a digital receipt is created by the system, listing the items purchased, total cost, and payment method used. This receipt is emailed to the customer, giving them a secure record of their purchase.

Alternative Flow:

If the customer changes their payment method at checkout, such as switching from a credit card to PayPal, the system modifies the payment details accordingly. The transaction is securely processed, and a digital receipt showing the new payment method is emailed to the customer.

Error Flow:

If the payment process encounters invalid card details, the system promptly identifies the issue, alerts the customer via a secure interface, and stops the transaction. No receipt is issued, and the customer is given the option to either correct their payment details or choose an alternate payment method to finalize the purchase securely.

Scenario 3: Inventory Management

Written By Hatem Hussein-1200894

Normal:

An employee utilizes a specialized interface integrated into the newsagent shop's management system, tailored for streamlined inventory management. Upon logging in, the employee navigates to the inventory section to adjust stock levels following the arrival of new newspaper shipments. They input the updated quantities, and the system promptly synchronizes the inventory data in real-time. This functionality guarantees the accuracy of inventory levels, reflecting immediate adjustments post-sales or new stock inputs. Additionally, the interface enables the addition of new items and the removal of outdated ones, ensuring the inventory remains current and accurate.

Alternative:

Upon routine inspection, the worker identifies a low stock alert for a popular magazine. The system promptly notifies the worker when stock levels drop critically low. The worker swiftly reorders through the interface, updating the expected delivery date. Upon arrival and processing of the new stock, the system is updated, preventing overselling.

Error:

The worker attempts to update the stock level but encounters an issue when the system temporarily loses connectivity to the database server. Despite entering the updated figures, the system fails to save these changes due to the connectivity error. An error message, "Failed to update inventory. Please check your network connection and try again," is displayed on the screen. The worker may need to retry the update or contact IT support if the problem persists.

Scenario 4: Refund Processing

Written By Joud Hijaz - 1200342

Normal:

A customer opts to cancel their order shortly after placing it. Within the 30-minute window, they access their account on the newsagent app, locate their order history, and choose the order for cancellation. Clicking 'Request Refund,' the system verifies the request's timing and proceeds to refund using the original payment method. Concurrently, inventory is updated to re-list the item, and financial records are adjusted accordingly. An itemized receipt of the refund, including original transaction details, refund amount, and method used, is automatically generated and emailed to the customer, confirming reimbursement to their payment method.

Alternative:

A customer tries to request a refund after the 30-minute window has passed. The system checks the order timestamp against the current time, denying the refund and displaying a notification stating, "Refund request denied. Cancellation period expired." This informs the customer that the transaction cannot be reversed due to timing constraints.

Error:

During the refund request within the allowed timeframe, a system error occurs, possibly due to connectivity issues or service interruption. The system acknowledges the error with a message: "Refund processing failed due to a system error. Please retry or contact customer support." The customer may need to reach out to support for manual processing, ensuring resolution and adherence to system guidelines.

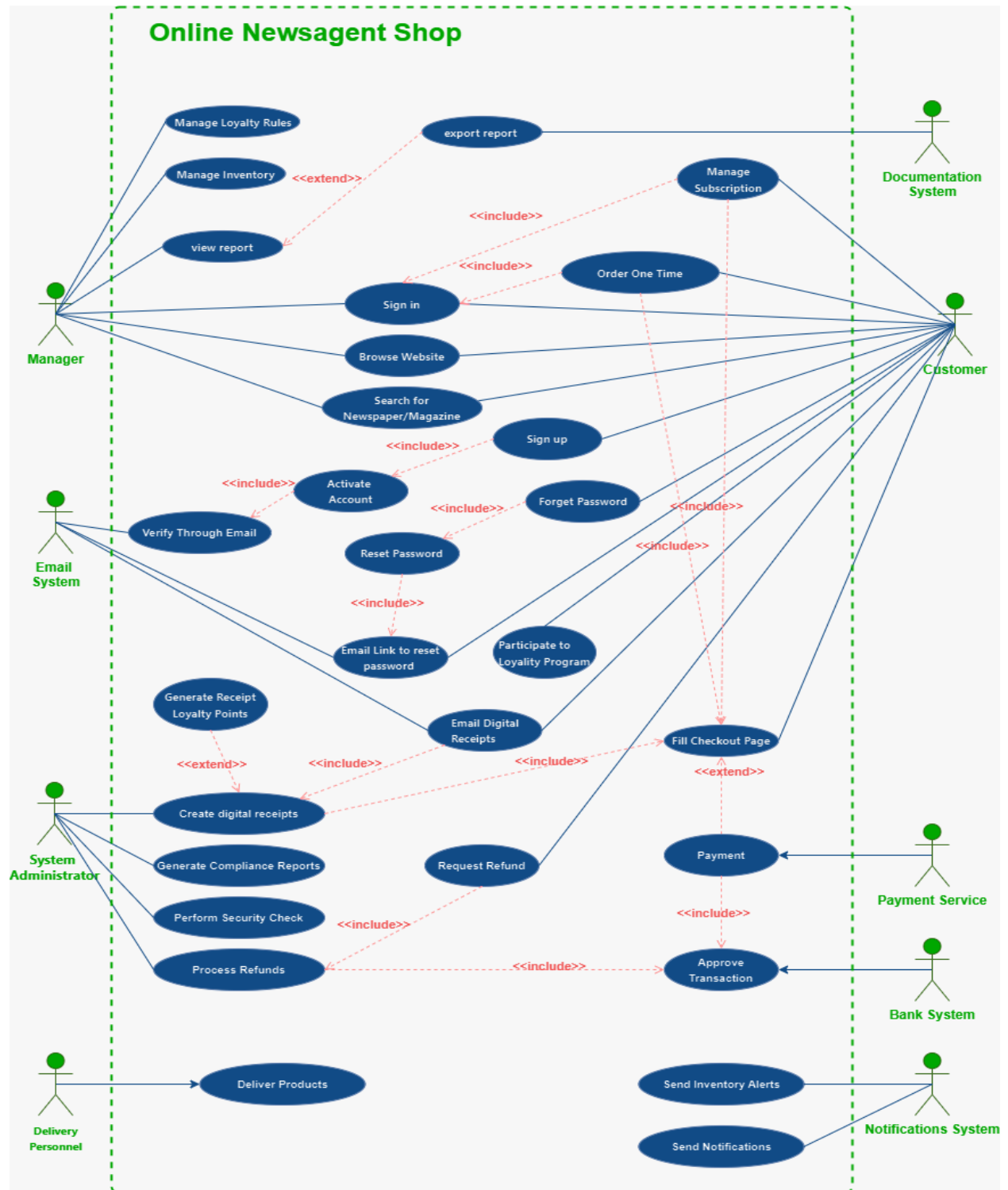
3.2 Actors Analysis

The below table describes all actors engaged with the system:

Actor	Description
Customer	Engages with the system to manage subscriptions, search for products, place orders, and process payments.
Manager	Oversees inventory and sales operations, ensuring the system runs smoothly and efficiently.
System Administrator	Maintains the system's functionality, performs security checks, and ensures compliance with data protection standards.
Email System	Automates sending of emails for verification, notifications, and receipts.
Payment Service	Processes payments and interacts with external banking systems to ensure secure transactions.
Bank System	Manages the financial transactions and security protocols related to payment processing.
Notifications System	Manages sending automated alerts and notifications to customers and staff.
Documentation System (Like Excel)	Utilized for generating downloadable reports and data analysis.
Delivery Personnel	Responsible for the physical delivery of newspapers and magazines to customers.

3.3 Use-Case Diagram

[Click here to see/download in better resolution](#)



3.4 Use-Case Specification

Use-Case Specification 1: User Sign Up *Written By Hatem Hussein-1200894*

1.1 Description

Allows new users to create an account by entering their personal details and verifying their email address.

The actor for this use-case is the user.

1.2 Sequence/Flow of Events

1. User selects the "Sign Up" option.
2. The system presents a form requesting personal details such as ID, name, phone number, and birthdate.
3. User fills out the form and submits the information.
4. The system checks for any pre-existing account with similar details. If an account exists, the system displays a message: "An account with this email already exists."
5. The system verifies that the entered password meets security requirements (minimum length, complexity).
6. If the email is unique and the password is secure, the system creates the new account. The system sends a verification email to the user's provided email address.
7. The user must click on the verification link sent to their email to activate their account. Once clicked, the system verifies the email and fully activates the account.
8. After email verification, the user is directed to a confirmation page stating, "Your account has been successfully activated. You can now log in."
9. If there are errors in the form (e.g., missing mandatory fields, password does not meet criteria), the system displays relevant error messages prompting the user to correct them.

1.3 Data

User ID, name, phone number, birthdate, email address.

1.4 Stimulus/Trigger

User selects "Sign Up" and submits the registration form.

1.5 Special Requirements

The system shall ensure that the email verification process is completed within 2 minutes of account creation to enhance security and user engagement.

1.6 Pre-Conditions

User is not currently logged in and is on the registration page.

1.7 Post-conditions/Response

The system creates a new user account.

User receives an email to verify their email address.

User's account becomes active after email verification.

The customer has all user properties the system offers.

Use-Case Specification 2: User Sign In

Written By Mohammad Abu Shams - 1200549

2.1 Description

Allows existing users to access their accounts by entering their registered email and password.

The actor for this use-case is the user.

2.2 Sequence/Flow of Events

1. User navigates to the "Sign In" page.
2. User inputs their registered email address and password into the respective fields on the "Sign In" page.
3. The system checks the inputted credentials against the stored data in the database. If the credentials match, the system proceeds to the next step.
4. Upon successful validation, the system creates a secure session for the user. The user is redirected to their account dashboard or the homepage, now logged in.
5. If the credentials do not match any account in the database, the system displays an error message: "Invalid email or password." The user is given the option to try again or reset their password.
6. The system limits the number of failed login attempts to prevent brute force attacks. After several failed attempts, the system temporarily locks the account and suggests the user reset their password or contact customer support.
7. If the user clicks on "Forgot Password," they are prompted to enter their email address. The system sends a password reset link to the provided email if it's associated with an active account. The user must follow the link to create a new password and regain access to their account.

2.3 Data

User email, and password.

2.4 Stimulus/Trigger

User enters credentials and selects the "Sign In" option.

2.5 Special Requirements

The system shall ensure that the login process does not exceed 5 seconds, maintaining a smooth and efficient user experience.

The system shall enforce a security lockout after five consecutive failed login attempts to protect user accounts from unauthorized access.

2.6 Pre-Conditions

The user must have an active account created and verified.

2.7 Post-conditions/Response

If credentials are correct, the user gains access to their account and is redirected appropriately.

If incorrect, the system displays an error message and offer the user the chance to reset their password or try again.

Use-Case Specification 3: Newspaper Search *Written By Joud Hijaz - 1200342*

3.1 Description

Allows users to search for newspapers and magazines and check availability.
The actor for this use-case is the customer.

3.2 Sequence/Flow of Events

1. User enters keywords in the search bar on the newsagent shop's website.
2. The system processes the search query, matching keywords with titles, authors, or topics. System checks inventory in real-time to display current availability.
3. Search results are displayed with options to filter by date, popularity, or availability. Each item shows title, author, availability status, and a brief description.
4. If no matches are found, the system displays "No results found. Try different keywords."

3.3 Data

Keywords for search, titles, authors, topics, availability status.

3.4 Stimulus/Trigger

User types into the search bar and submits a search query.

3.5 Special Requirements

The search function shall retrieve results in no more than 2 seconds to ensure quick access to information.

3.6 Pre-conditions

User is logged into the system.

3.7 Post-conditions/Response

The system shows a list of newspapers and magazines that match the search criteria.

Use-Case Specification 4: Inventory Management

Written By Hatem Hussein - 1200894

4.1 Description

Allows workers to manage the inventory of newspapers and magazines.

The actor/s for this use-case is the workers (Inventory Managers) and the system itself.

4.2 Sequence/Flow of Events

1. Employee navigates to the "Inventory Management" section.
2. System displays current stock levels and management options.
3. Employee inputs new quantities for newspapers or magazines based on recent shipments or sales. System updates the inventory database in real-time.
4. Employee can add new items or remove outdated ones from the inventory. System updates accordingly and confirms changes.
5. If a connectivity issue occurs during updates, the system displays: "Failed to update inventory. Please check your network connection and try again."

4.3 Data

Inventory levels, new item details, old item details.

4.4 Stimulus/Trigger

Inventory levels change or updates are needed.

4.5 Special Requirements

The system shall alert workers of low inventory levels within 1 minute of reaching the predefined threshold.

4.6 Pre-conditions

Worker has logged into the inventory management system.

4.7 Post-conditions/Response

The system provides updated inventory data and alerts for low stock levels.

Use-Case Specification 5: Payment Processing

Written By Mohammad Abu Shams - 1200549

5.1 Description

Manages various payment methods and generates digital receipts for each transaction.
The actor/s for this use-case is the Customer, and the E-Payment system.

5.2 Sequence/Flow of Events

1. The customer selects their desired items (newspapers, magazines) and moves to the checkout page.
2. The customer is presented with various payment options including credit cards, PayPal, and Apple Pay.
3. Depending on the selected method, the customer inputs their payment details such as card number, expiration date, CVV for credit cards, or logs into their digital wallet accounts.
4. The payment details are validated by the system to check for any input errors or potential fraud. Once validated, the information is sent securely to the payment gateway.
5. The payment gateway forwards the payment request to the respective bank or financial service. The bank processes the payment and sends back the transaction status (approved or declined).
6. If the transaction is approved, the system generates a digital receipt detailing the purchase.
7. The receipt is then emailed to the customer, confirming the transaction.
8. If the transaction is declined or an error occurs, the system alerts the customer with a detailed message about the failure.
9. The customer is prompted to try another payment method or correct the existing details.

5.3 Data

Payment method details, transaction details.

5.4 Stimulus/Trigger

User chooses to pay for their order.

5.5 Special Requirements

Payment processing shall comply with the latest PCI DSS standards to ensure secure transactions.

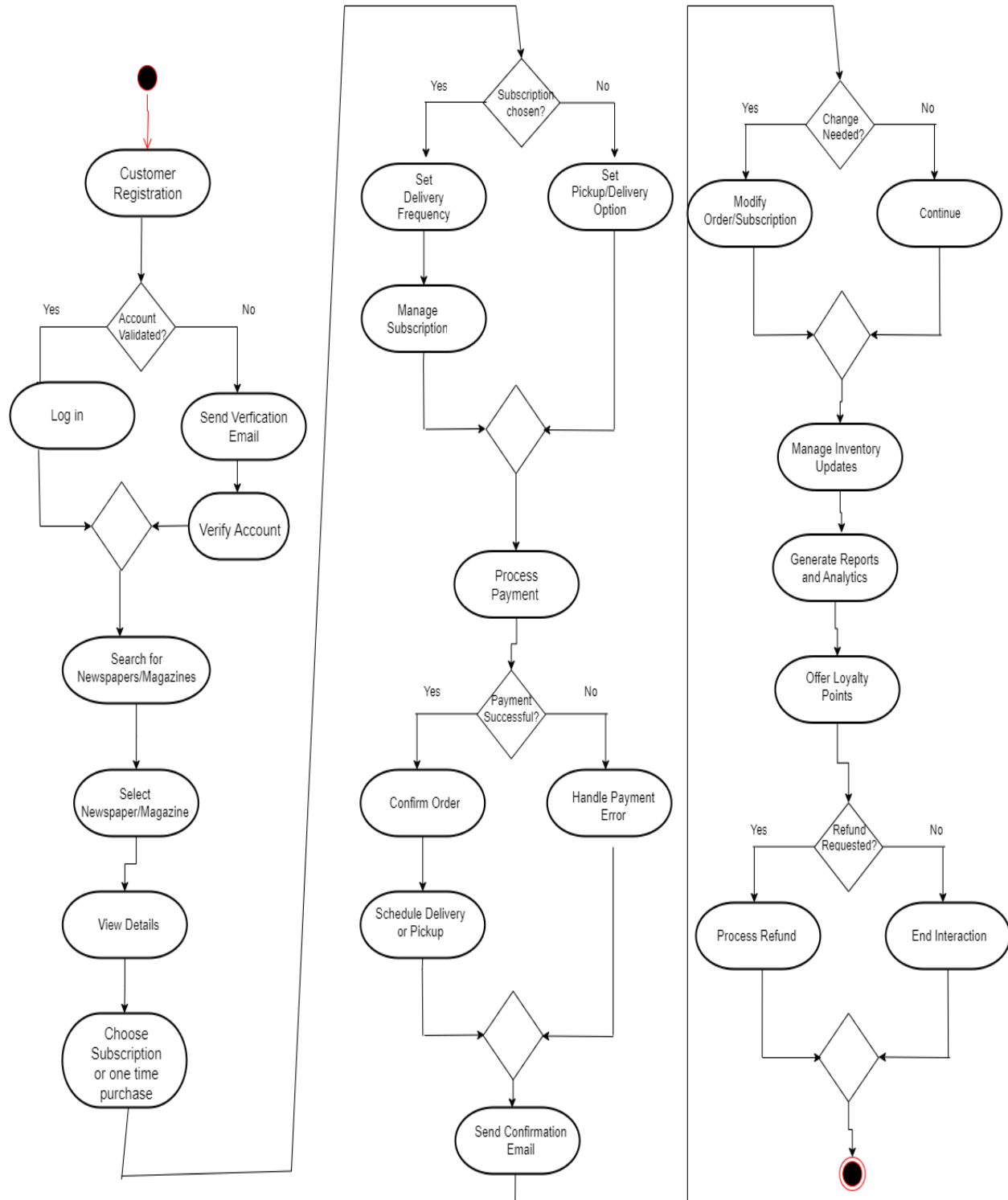
5.6 Pre-conditions

User has selected items to purchase and proceeded to checkout.

5.7 Post-conditions/Response

The system generates and sends a digital receipt to the user.

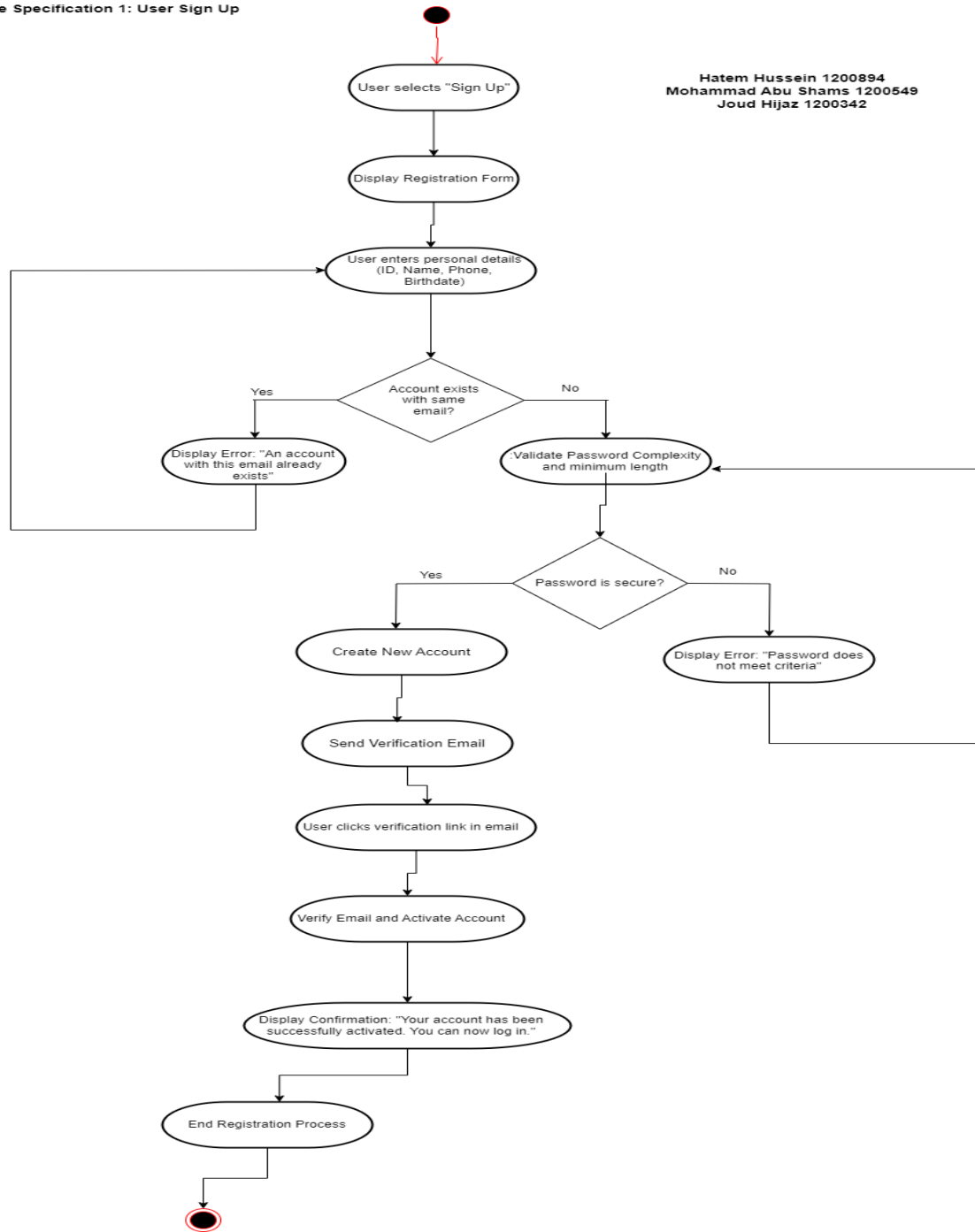
3.5 Activity Model/Diagram



3.6 Instance Activity Diagrams

Use-Case Specification 1: User Sign Up *Designed By Hatem Hussein-1200894*

Use-Case Specification 1: User Sign Up

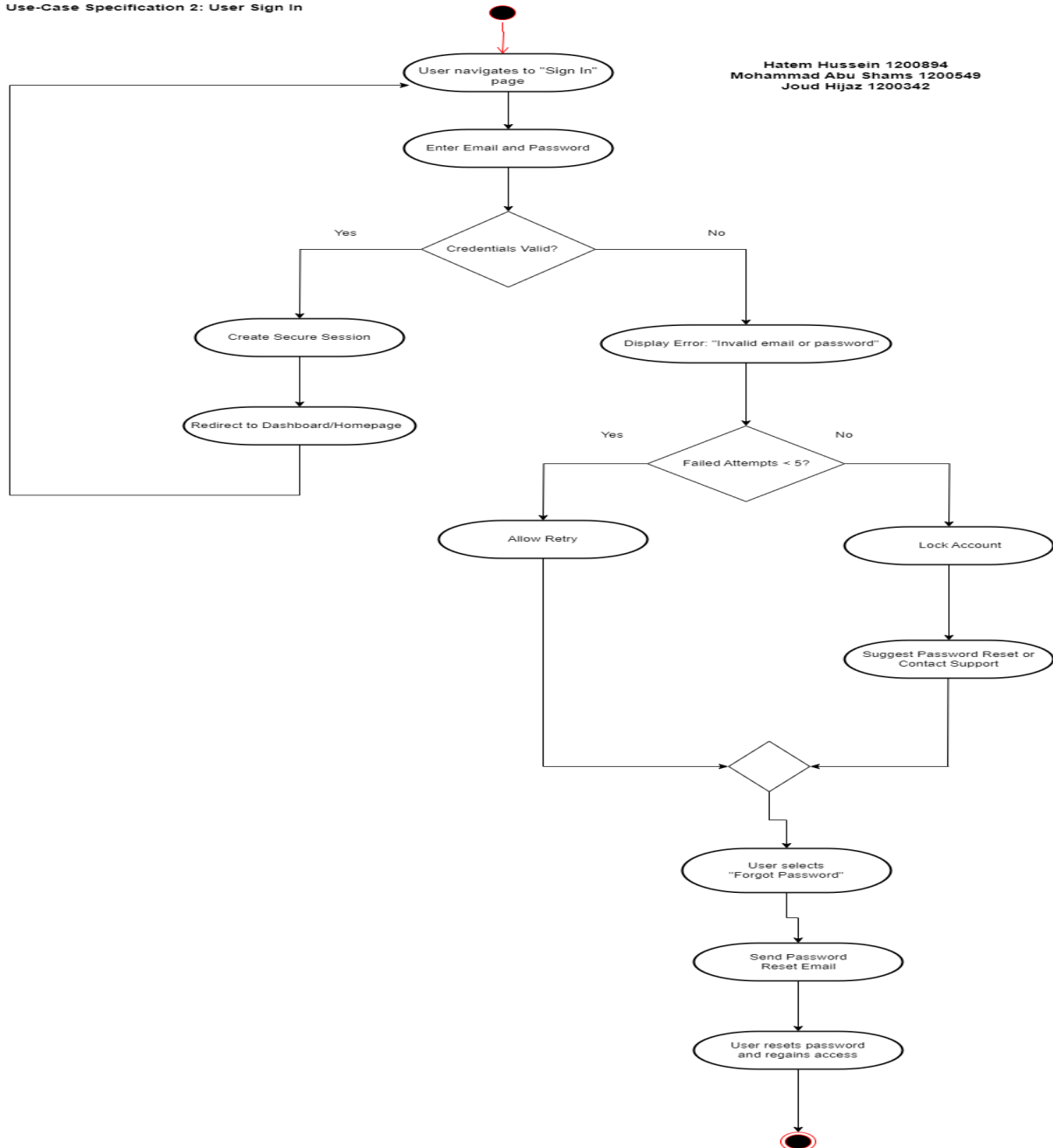


Hatem Hussein 1200894
Mohammad Abu Shams 1200549
Joud Hijaz 1200342

Use-Case Specification 2: User Sign In

Designed By Mohammad Abu Shams - 1200549

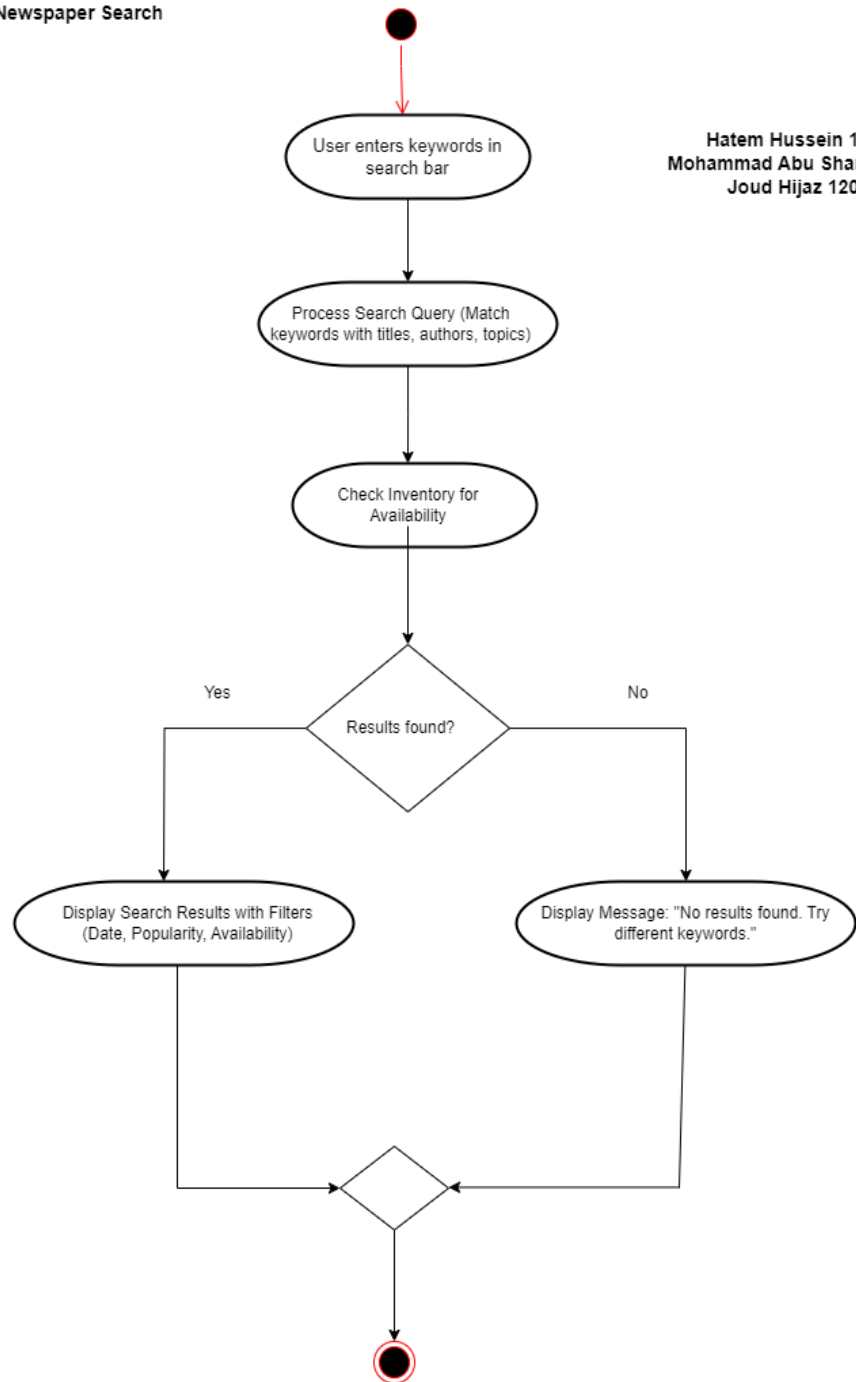
Use-Case Specification 2: User Sign In



Use-Case Specification 3: Newspaper Search

Designed By Joud Hijaz – 1200342

Use-Case Specification 3: Newspaper Search

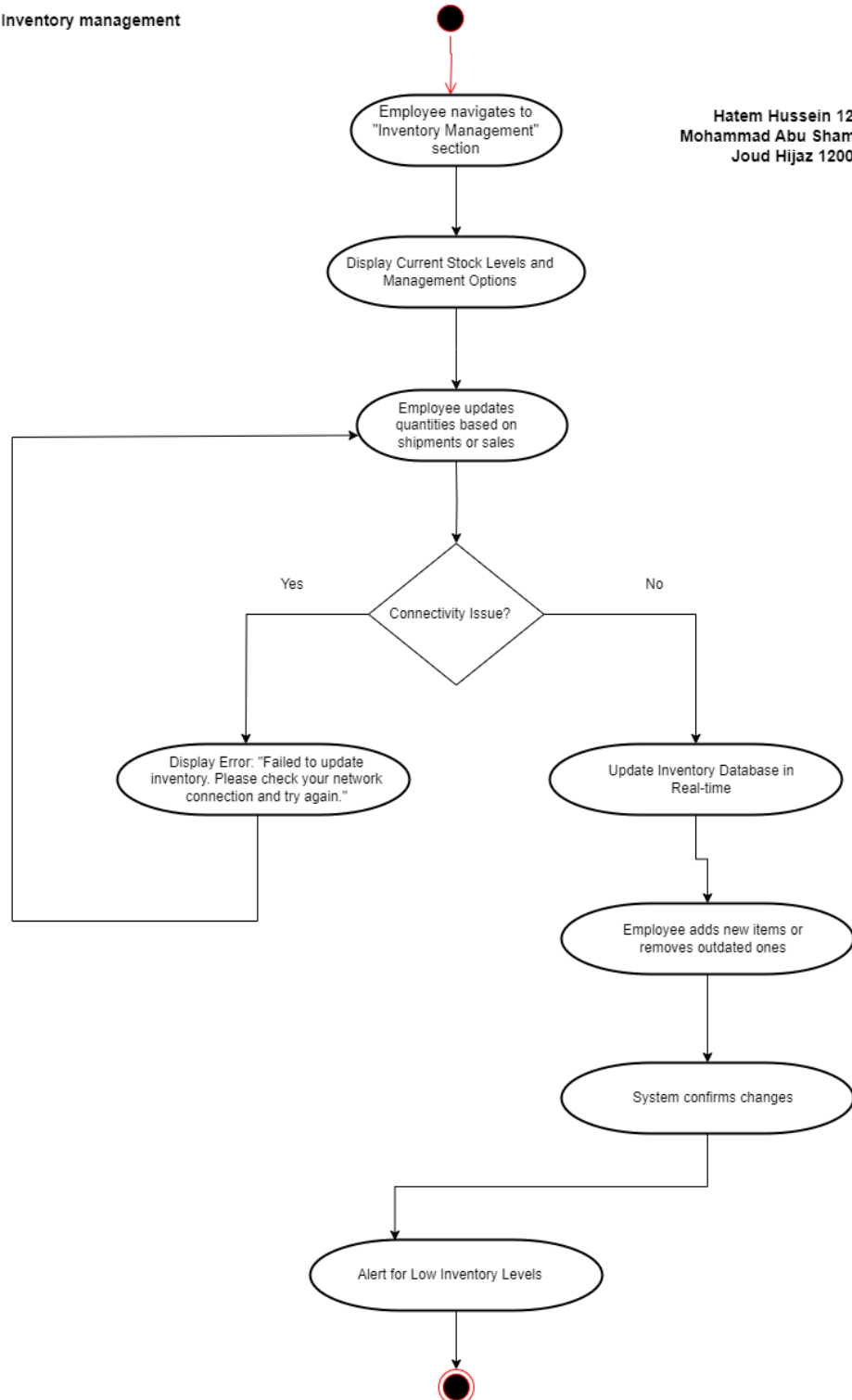


Hatem Hussein 1200894
Mohammad Abu Shams 1200549
Joud Hijaz 1200342

Use-Case Specification 4: Inventory Management

Designed By Hatem Hussein-1200894

Use-Case Specification 4: Inventory management



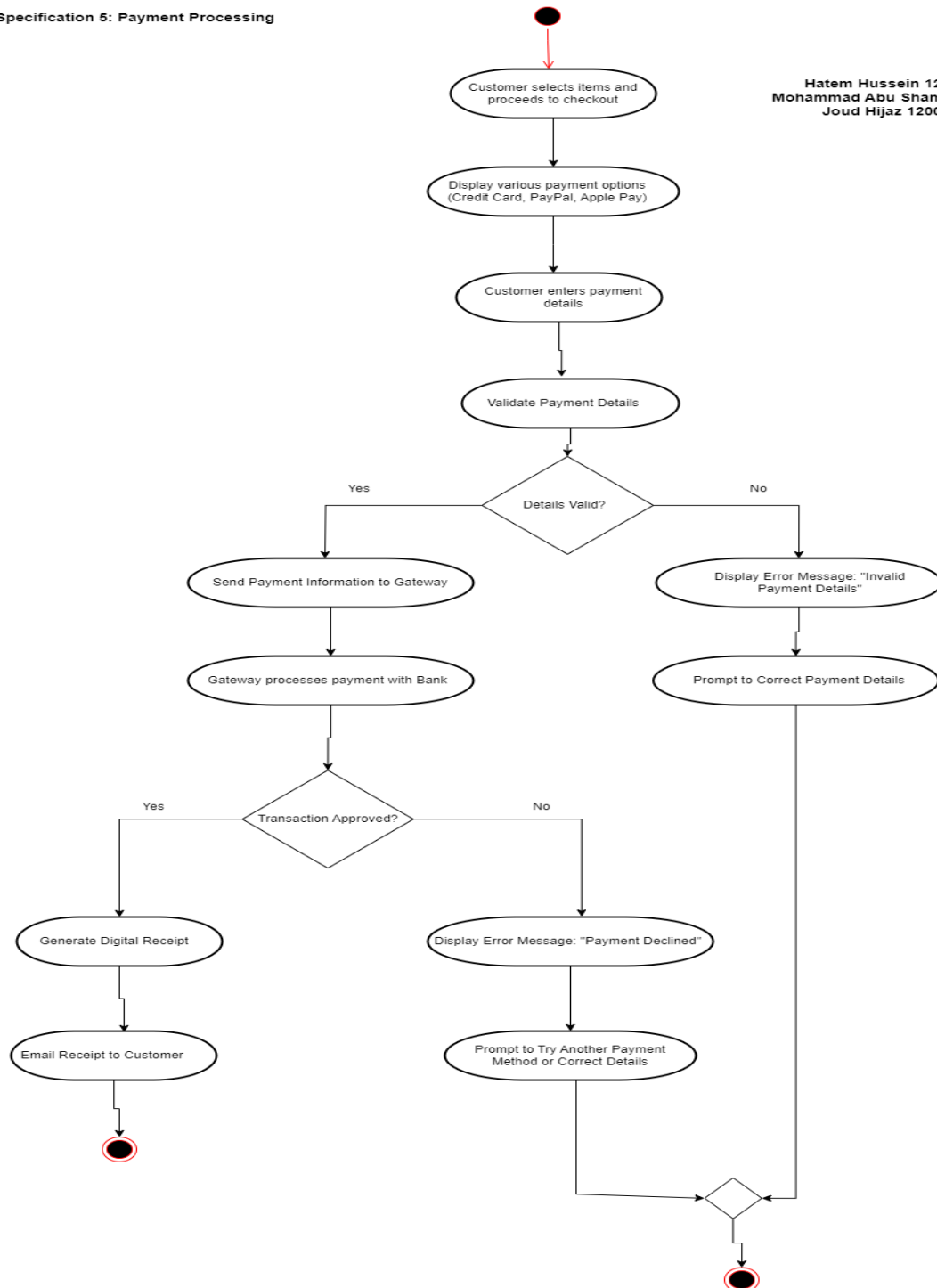
Hatem Hussein 1200894
Mohammad Abu Shams 1200549
Joud Hijaz 1200342

Use-Case Specification 5: Payment Processing

Designed By Mohammad Abu Shams - 1200549

Use-Case Specification 5: Payment Processing

Hatem Hussein 1200894
Mohammad Abu Shams 1200549
Joud Hijaz 1200342



Phase 4 - System Modeling and Design

4.1 System Class modelling and Analysis

Identifying the system analysis classes:

1. User

- **Attributes:** UserID, Name, Email, Password, PhoneNumber, BirthDate, Address
- **Methods:** SignUp(), SignIn(), ResetPassword(), UpdateProfile()

2. Subscription

- **Attributes:** SubscriptionID, UserID, NewspaperID, DeliveryFrequency, StartDate, EndDate, Status
- **Methods:** CreateSubscription(), UpdateSubscription(), CancelSubscription(), RenewSubscription()

3. Newspaper

- **Attributes:** NewspaperID, Title, Author, Category, Availability, Price, Description
- **Methods:** AddNewspaper(), UpdateNewspaper(), RemoveNewspaper(), SearchNewspaper()

4. Order

- **Attributes:** OrderID, UserID, NewspaperID, Quantity, TotalPrice, OrderDate, DeliveryMethod, Status
- **Methods:** CreateOrder(), UpdateOrder(), CancelOrder(), ProcessRefund()

5. Inventory

- **Attributes:** NewspaperID, StockLevel, ReorderLevel
- **Methods:** UpdateStockLevel(), CheckStockLevel(), GenerateInventoryReport()

6. Payment

- **Attributes:** PaymentID, OrderID, UserID, Amount, PaymentMethod, PaymentDate, Status
- **Methods:** ProcessPayment(), ValidatePayment(), GenerateReceipt()

7. Report

- **Attributes:** ReportID, ReportType, Data, GeneratedDate
- **Methods:** GenerateSalesReport(), GenerateInventoryReport(), GenerateLoyaltyReport()

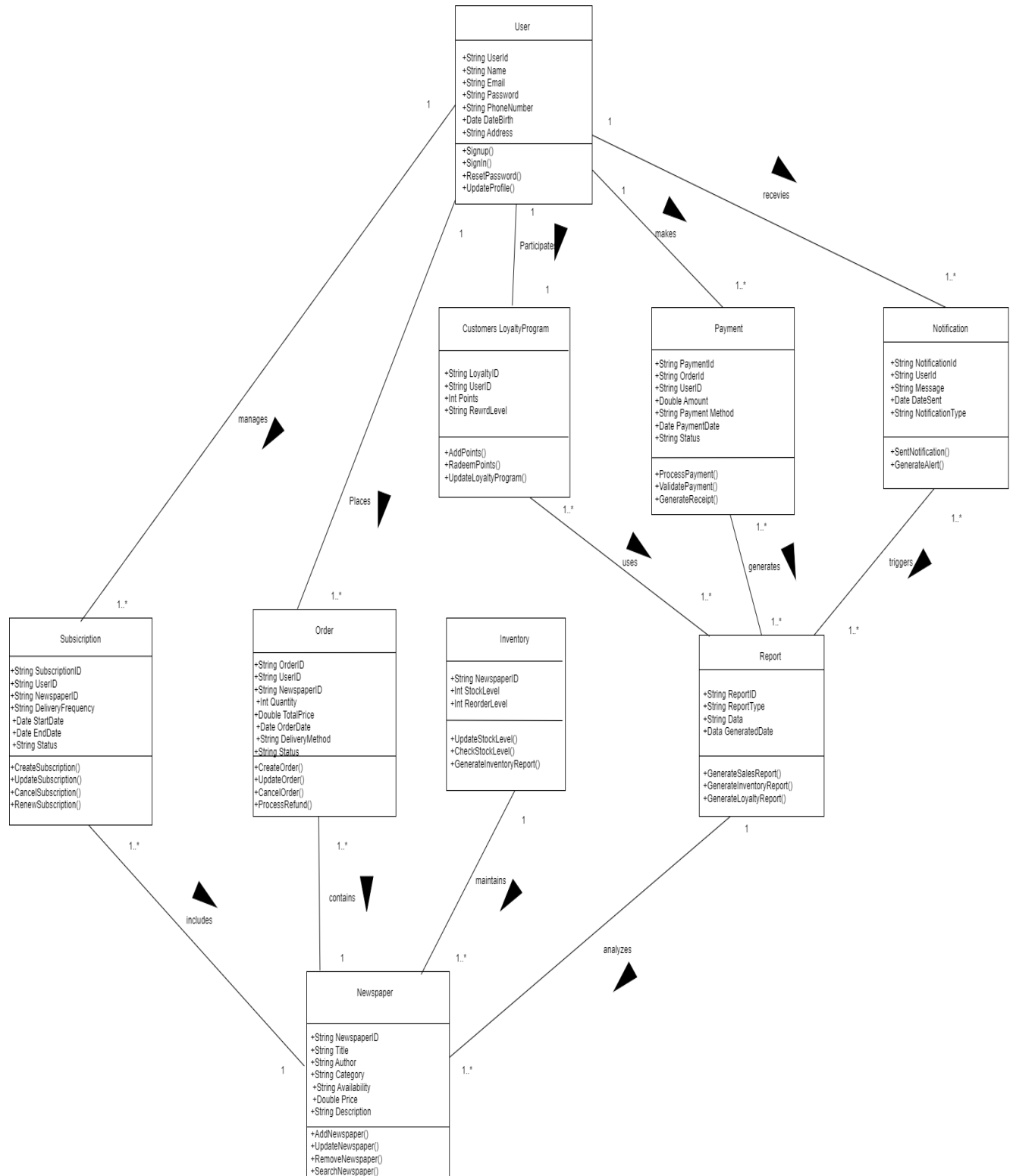
8. Customer LoyaltyProgram

- **Attributes:** LoyaltyID, UserID, Points, RewardLevel
- **Methods:** AddPoints(), RedeemPoints(), UpdateLoyaltyProgram()

9. Notification

- **Attributes:** NotificationID, UserID, Message, DateSent, NotificationType
- **Methods:** SendNotification(), GenerateAlert()

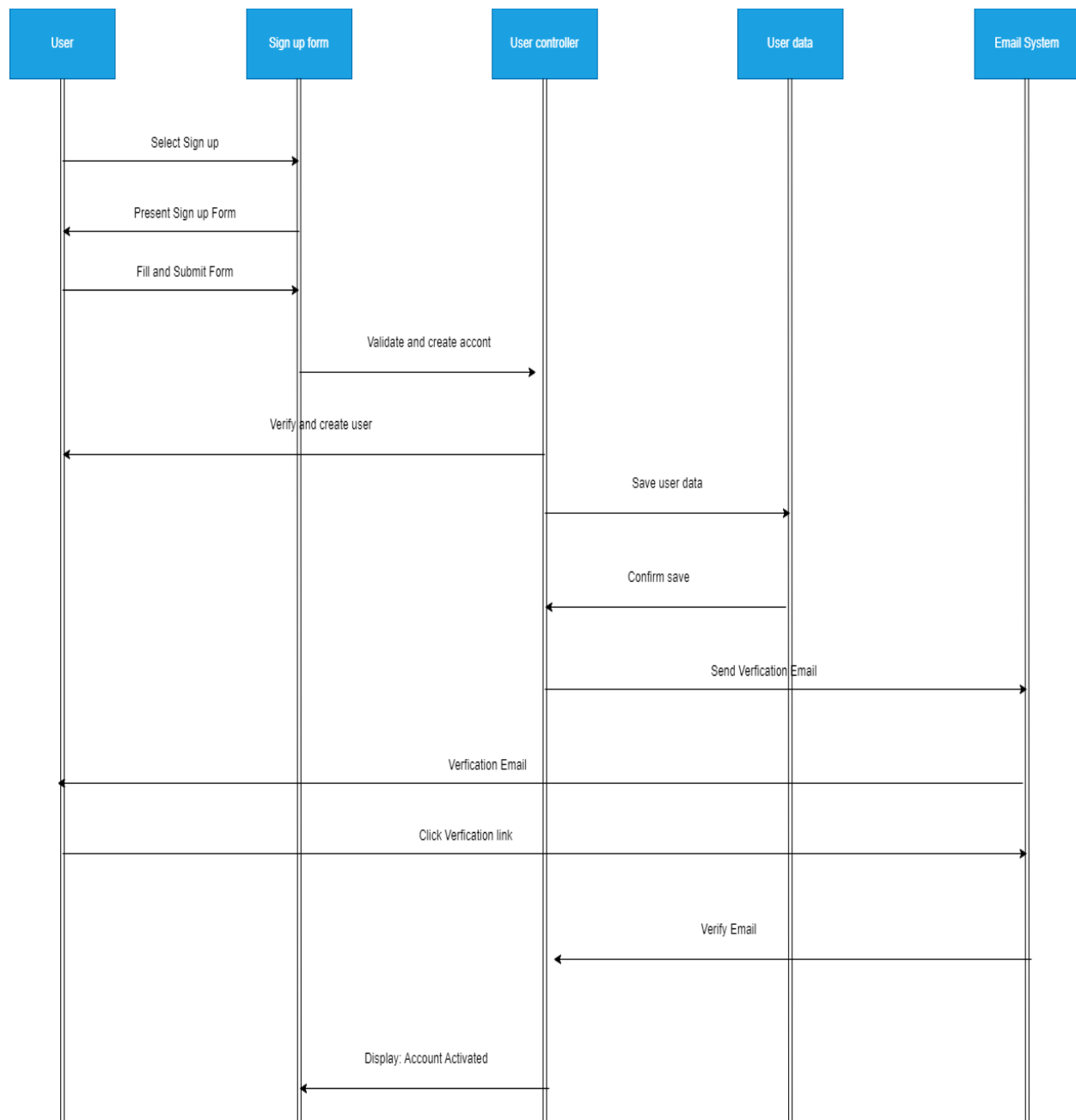
Below is the class diagram (see next page):



4.2 System Sequence modelling

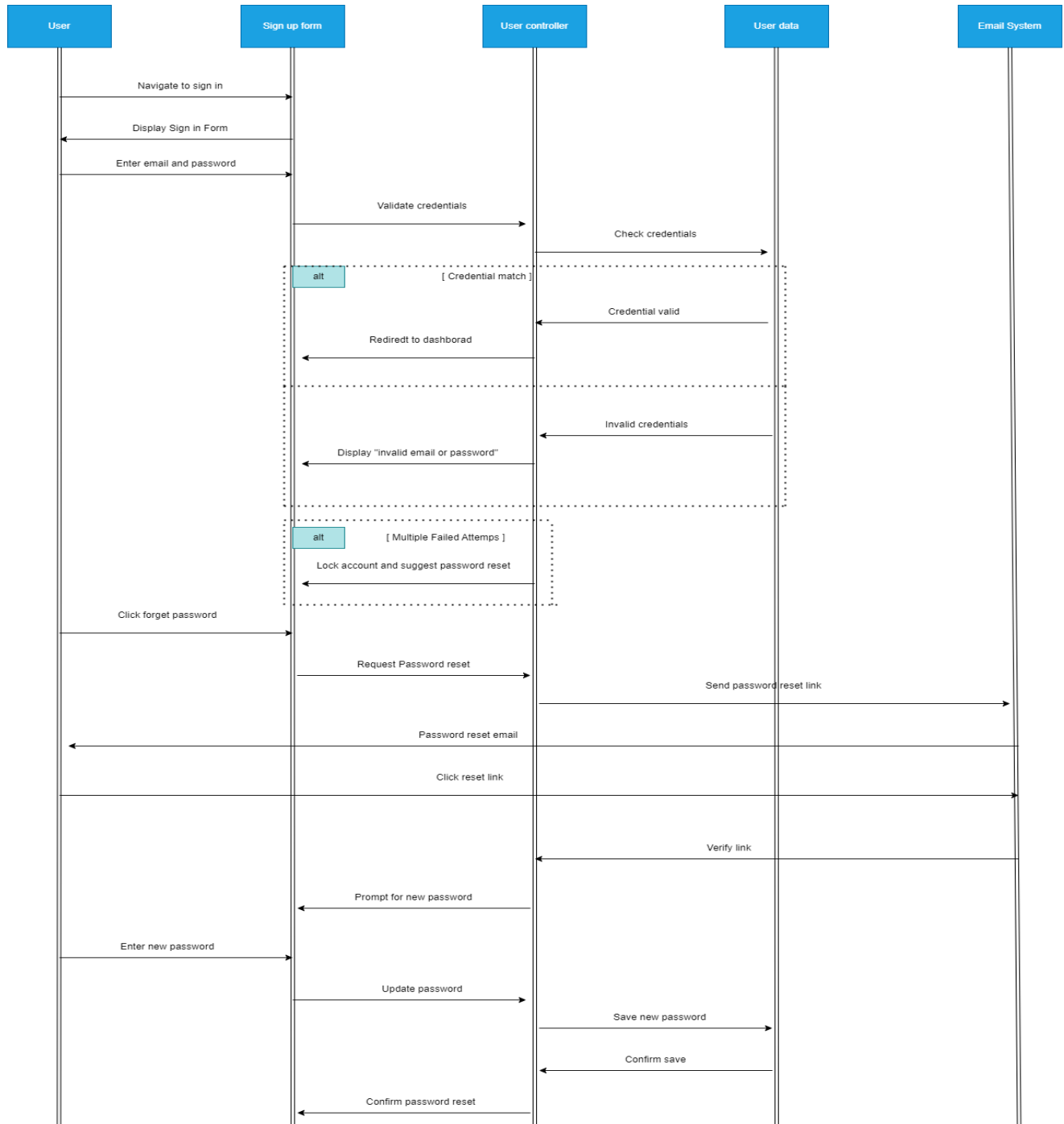
• Use-Case Specification 1: User Sign Up (Designed By Mohammad Abu Shams)

Let new users sign up by filling in their personal information and confirming their email address.



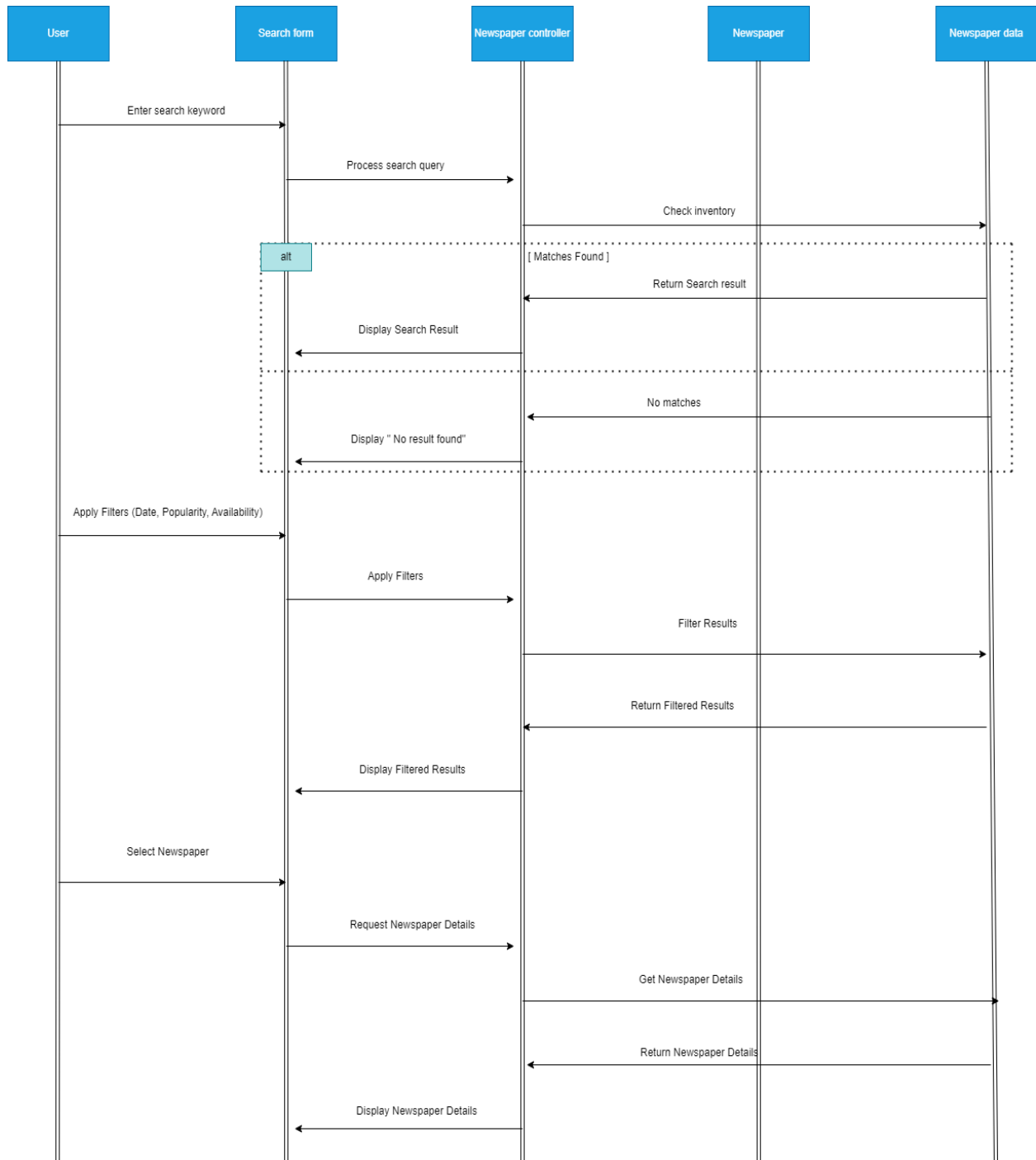
• Use-Case Specification 2: User Sign In (Designed By Joud Hijaz)

Let existing users log in by entering their registered email and password.



• Use-Case Specification 3: Newspaper Search (Designed By Hatem Hussein)

Allows users to search for newspapers and magazines and check availability.



4.3 System Design Goals

General Design Goals

1. Low Coupling

Description: Low coupling means keeping different parts of a system separate from each other. This way, each part can work on its own, and changes in one part don't heavily impact others.

Importance: Low coupling makes the system easier to maintain and grow. It allows for easier testing, updates, and changes to individual parts without disrupting the whole system.

How to Achieve Low Coupling:

- **Use Interfaces and Abstract Classes:** Create clear templates for parts of the system like `UserService`, `SubscriptionService`, and `PaymentService`. This lets you change how they work inside without affecting other parts.
- **Dependency Injection:** Manage how parts of the system depend on each other by providing their needs from outside, instead of having them create what they need on their own.
- **Modular Design:** Divide the system into clearly defined parts with specific roles and boundaries.

2. High Cohesion

Description: High cohesion means that everything in a part of the system is closely related and works together. Each part should focus on doing one task or a set of related tasks.

Importance: High cohesion makes the code easier to read, maintain, and reuse. It helps people understand the system better and reduces mistakes.

How to Achieve High Cohesion:

- **Single Responsibility Principle:** Make sure each class or module has only one job. For example, a `UserController` should only handle user-related requests, and a `UserService` should handle the business logic for user management.
- **Group Related Functions:** Organize your code by what it does, not by its type. For example, put all the code related to user management in one place.
- **Regular Refactoring:** Regularly review and update your code to keep it well-organized. Break down large or complex parts into smaller, easier-to-manage pieces.

Specific System Design Goal

3. Performance in Terms of Processing Speed

Description: The system should handle user requests and process data quickly to ensure fast response times. This includes making the search function efficient, providing quick access to inventory data, and promptly processing user requests.

Importance: High performance is crucial for keeping users happy and making the system easy to use. Slow processing can lead to a bad user experience, lower employee productivity, and losing customers.

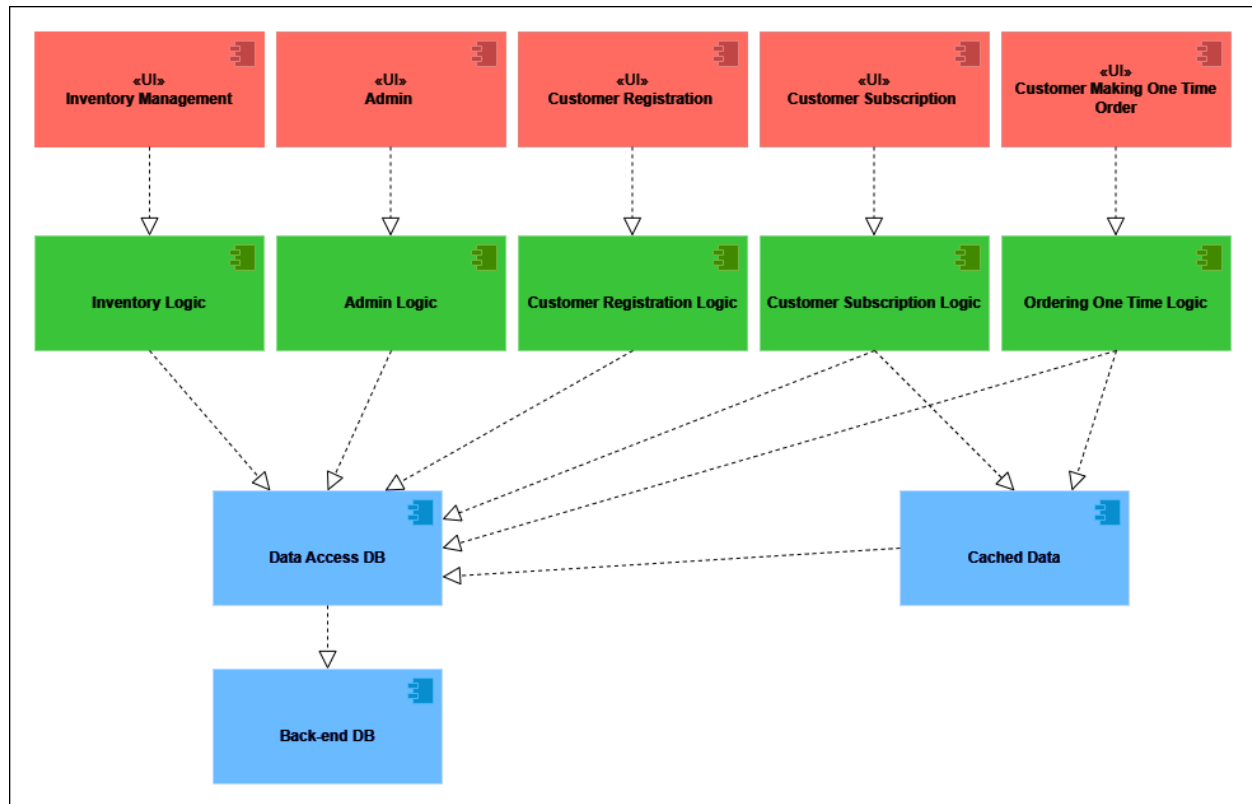
How to Achieve High Performance:

- **Caching Strategies:** Use in-memory caching like *Redis* or *Memcached* for data that is accessed often, and cache the results of complex operations or queries. Caching strategy is represented in the design by the cached data block in the Component Diagram.
- **Optimize Database Queries:** Make searches faster by using indexing, avoid slow queries by using joins, and manage database connections efficiently with connection pooling.

4.4 System Component Design

4.4.1 The Component Diagram

The below diagram represents the Component Diagram of our system:



NOTICE To see/download the diagram in better resolution click [here](#).

4.4.2 Components and Their Functionalities

Presentation Layer

1. Inventory Management (UI)

- **Functionality:** Provides the user interface for managing inventory. Users can add, update, and delete inventory items.
- **Interaction:** Interacts with the Inventory Logic to perform operations related to inventory management.

2. Admin (UI)

- **Functionality:** Offers the interface for administrative tasks. Admin users can manage various aspects of the system, including user accounts and system settings.
- **Interaction:** Connects with Admin Logic to perform administrative functions.

3. Customer Registration (UI)

- **Functionality:** Allows customers to register for an account. This interface captures customer details and manages account creation.
- **Interaction:** Utilizes Customer Registration Logic to handle the registration process.

4. Customer Subscription (UI)

- **Functionality:** Provides an interface for customers to subscribe to services. Customers can choose subscription plans and manage their subscriptions.
- **Interaction:** Communicates with Customer Subscription Logic to manage subscription services.

5. Customer Making One Time Order (UI)

- **Functionality:** Enables customers to place one-time orders. This interface facilitates the order process, allowing customers to select products and complete purchases.
- **Interaction:** Interfaces with Ordering One Time Logic to handle one-time order transactions.

Domain Layer

1. Inventory Logic

- **Functionality:** Contains the business rules and logic related to inventory management. It handles operations such as adding, updating, and retrieving inventory items.
- **Interaction:** Interacts with the Data Access DB to perform inventory operations and update the database accordingly.

2. Admin Logic

- **Functionality:** Manages the business rules and logic for administrative tasks. It includes functionalities like user management, system configuration, and monitoring.
- **Interaction:** Connects with the Data Access DB to perform administrative operations and manage system data.

3. Customer Registration Logic

- **Functionality:** Handles the logic for customer registration. It validates customer data, creates new accounts, and manages registration workflows.
- **Interaction:** Utilizes the Data Access DB to store and retrieve customer information during the registration process.

4. Customer Subscription Logic

- **Functionality:** Manages the subscription services. It handles the logic for subscribing to services, managing subscription plans, and billing.
- **Interaction:** Communicates with the Data Access DB and Cached Data for subscription management, searching operations, and billing operations.

5. One Time Ordering Logic

- **Functionality:** Contains the business rules for processing one-time orders. It manages order validation, payment processing, and order fulfillment.
- **Interaction:** Interfaces with the Data Access DB and Cached Data to process orders and update inventory.

Infrastructure Layer

1. Data Access DB

- **Functionality:** Acts as the primary data repository for the system. It manages data storage, retrieval, and update operations for various components.
- **Interaction:** Interacts with all logic components (Inventory Logic, Admin Logic, Customer Registration Logic, Customer Subscription Logic, Ordering One Time Logic) to perform database operations. It also communicates with the Back-end DB for persistent storage.

1. Cashed Data

- **Functionality:** Provides in-memory caching for frequently accessed data. It improves system performance by reducing the need to query the database for every request.
- **Interaction:** Works with various logic components (Customer Subscription Logic, Ordering One Time Logic) to cache data that is frequently accessed or computationally expensive to retrieve.

2. Back-end DB

- **Functionality:** Serves as the underlying database system for long-term data storage. It ensures data persistence and integrity.
- **Interaction:** Communicates with the Data Access DB to store and retrieve data as needed.

4.5 System Deployment Design

The below figure represents the Deployment Diagram of our system with each component having hardware and software details:

