**Name Zain Arshad**

**Class Bse-4A**

**Reg-No Fa19-Bse-082**

# **Assigned Use Cases**

1. Create Account
2. Login Account
3. Deliver Order
4. Ship Order
5. Payment
6. Return Items

# **CHAPTER 1 Brief Use Case**

#### **Use Case: Create Account**

User opens the Construction and Materials Rent App and clicks on the register button. The system takes the user to the registration page. The user is required to enter information asked by the system. The user then enters his full name, email, address, phone number and password. The system then checks if the user is already registered or not. If he is not registered the system validates and registers him as a customer.

#### **Use Case: Login Account**

User opens the Construction and Materials Rent App and clicks on the login button. The system takes the user to the login page. The user is required to enter information asked by the system which is the username and the password. The system then checks if the user is already registered or not. If he is registered the system validates and logs him in.

#### **Use Case: Deliver order**

After the user has selected items to rent and then order the items.The user will provide payment and shipping information. The system will respond with confirmation of the order and a tracking number that the user can use to check on order status in the future. System verifies the availability of selected product, payment from customer and then delivers or track order to the customer address.

#### **Use Case: Ship Order**

A customer opens a checkout with items to purchase. The cashier uses the POS system to record each purchased item. The system presents a running total and line-item details. The customer enters payment information, which the system validates and records. The system updates inventory. The customer receives a receipt from the system and then leaves with the items.

#### **Use Case: Payment**

Customer opens the Construction and Materials Rent App to place order. After placing the order customer requests for the payment process. System shows the secure payment method to the customers. Payment method involves cash on delivery or online payment through Easypaisa or ATM.

#### **Use Case: Return item**

Customers return the item. System check that the return item is OK or not. System set the deadline for returning orders if the deadline pass send calculate penalty according to the rent item to customer. If the deadline is not pass system allows to the customer to return the item and confirm the items. System also checks return item ID and quantity.

# **CHAPTER 2 Fully Dressed Use Case**

| | **Use Case UC1: Create Account** | | --- | | **Scope**: Construction and Materials Rent App  **Level**: User goal  **Primary** **Actor**: Customer  **Stakeholders and Interests**:  - Customer: want to register an account in the Construction and Materials Rent App.  - Manager: want to register an account in the Construction and Materials Rent App.  **Preconditions**: User is identified and authenticated. |   **Success Guarantee** (or Postconditions): username is available, and account is registered. Account is not already registered.  **Main Success Scenario (or Basic Flow):**   1. User clicks on register an account button. 2. System redirects him to the registration page 3. System asks him to provide the details. 4. The user enters the details. 5. System then registers the account when user clicks on register 6. System then takes the user to his dashboard where he can access the store and his own profile.   **Extensions (or Alternative Flows):**  \*a. At any time when the user tries signing up:   1. user enters the username and password he used when registering his account. 2. The system then verifies his credentials when he clicks on the register button. 3. when the credentials matches then he is redirected to his profile.   \*b. At any time when the user tries signing up:   1. user enters the password and username he used when registering his account. 2. The system then verifies his credentials. 3. If the credentials don’t match, then the system displays an error either his password or username is incorrect.   **Special Requirements:**   * + 1. Text should be visible from 2 meters.     2. Color scheme should be used which is clearly visible.     3. Special characters can be used in username and password.     4. It should take less than 1 minutes to register an account.   **Technology and Data Variations List**:   1. Languages used is java. 2. Software used to design interface is NetBeans, Gui Swing.  **Use Case UC2: Login** |
| --- | --- | --- |
| **Scope**: Construction and Materials Rent App  **Level**: user goal  **Primary** **Actor**: customer, admin  **Stakeholders and Interests**:  - Customer: want to login in the Construction and Materials Rent App  - Manager: want to login in the Construction and Materials Rent App  **Preconditions**: User is identified and authenticated. |

**Success Guarantee** (or Postconditions): username and password are valid, and account is registered. User must be able to login.

**Main Success Scenario (or Basic Flow):**

1. User clicks on login button.
2. System redirects him to the log in page
3. System asks him to provide the details.
4. The user enters username, and password
5. System then takes the user to his dashboard where he can access the store and his own profile.

**Extensions (or Alternative Flows):**

\*a. At any time when the user tries to login:

1. user enters the username and password he used when registering his account.
2. The system then verifies his credentials when he clicks on the register button.
3. when the credentials matches then he is redirected to his profile.

\*b. At any time when the user tries to login:

1. user enters the password and username he used when registering his account.
2. The system then verifies his credentials.
3. If the credentials don’t match, then the system displays an error either his password or username is incorrect.

**Special Requirements:**

* + 1. Text should be visible from 2 meters.
    2. Color scheme should be used which is clearly visible.
    3. Special characters can be used in username and password.
    4. It should take less than 1 minutes to register an account.

**Technology and Data Variations List**:

* 1. Languages used is java.
  2. Software used to design interface is NetBeans, Gui Swing.

**Open Issues:**

- What are the system requirements?

- What customization is needed for different businesses?

- Must the user log out before exiting the software?

- Can the customer directly buy without logging in, or does the customer have to register an account?

| **Use Case UC3: Return Items** |
| --- |
| **Scope**: Construction and Materials Rent App  **Level**: user level  **Primary** **Actor**: Customer  **Stakeholders and Interests**:  - Customer: Wants to return item and fast service with minimal effort. Wants proof of purchase to support returns.  - Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded. Wants some fault tolerance to allow sales capture even if server components (e.g., remote credit validation) are unavailable. Wants automatic and fast update of accounting and inventory.  - Manager: Wants to be able to quickly perform override operations, and easily debug Cashier problems.  **Preconditions**:   * Customer is identified and authenticated. * Customer return item if deadline of returning item is not pass |

**Success Guarantee** (or Postconditions): customer successfully return the items. receipt generate. Customers give full payment.

**Main Success Scenario (or Basic Flow):**

1. Customer goes to the Construction and Materials Rent App to return item
2. Customer wants to return the item
3. Customer asks the system to return item
4. System check that the return item is OK or not
5. System set the deadline for returning orders if the deadline has passed the system generate extra payment according to the days used over the deadline or if the item is not as its original condition then the customer will to pay extra according to penalty of that item.
6. If the deadline is not pass system allows to the customer to return the order and pay the full payment.
7. System request to customer to pay payment of returned item.
8. Customer paid the payment of returned product.

**Extensions (or Alternative Flows):**

\*a. At any time, Manager will manage all the details of system

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. manager restarts System, logs in, and requests recovery of all data
2. System recovers all the information

2a. System detects anomalies preventing recovery:

* 1. System signals error to the manager, records the error, and enters a clean state
  2. Manager starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation and enters the ID to retrieve the sale.
2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

* 1. System signals error to the manager
  2. manager probably starts new sale and re-enters all items.
  3. Manager continues with sale (probably entering more items or handling payment).

2a. Customer fails to return item

* System send message to customer that order will not return
* System send payment back to the customer

2b. Customer successfully returns the item

* System allows the customer to return item
* System asks for the payment process

**Special Requirements:**

. Text must be visible from 1 meter.

- user authorization response within 30 seconds 90% of the time.

- system fails then recover immediately

- Language internationalization on the text displayed.

-use color theme that is easily visible

- id and password must require registering the account

-Use at least one capital letter in login

-Password shows as a hidden digit

-If wrong entry at id or password uses red bar below the text

**Technology and Data Variations List**:

1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment signature captured on paper receipt
6. Use java language
7. Design use cases in NetBeans software using GUI swing

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

* Unprofessional design of software
* Login failures
* No category suggestions, photos or prices on sales events
* Bad user experience
* Lack of personalization
* Missing or fake products reviews
* Missing products information
* Too complex check-out process
* Payment failures
* Lack of security and privacy leak
* Not having flexible return policy

| **Use Case UC4: Track Order** |
| --- |
| **Scope**: Construction and Materials Rent App  **Level**: User goal  **Primary** **Actor**: Customer  **Stakeholders and Interests**:  - Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of ordered items and prices.  *- Company: wants to ensure that order is successfully placed by the customer.*  *- Manager: Wants to be able to quickly perform override operations, and easily debug customer problems.*  **Preconditions**  User is identified and authenticated. |

**Success Guarantee** (or Postconditions):

The user will have a tracking ID for the order.

**Main Success Scenario (or Basic Flow):**

1.Customer selects the item from the item list to place the order.

2. System will request for payment.

3.The user will provide payment and shipping information.

4.The system will respond with confirmation of the order and a tracking number that the user can use to check on order status in the future. After verification order will be delivered to the customer’s address.

**Extensions (or Alternative Flows):**

\*a. At any time, Manager will manage all the details of system

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. manager restarts System, logs in, and requests recovery of all data
2. System recovers all the information

2a. System detects anomalies preventing recovery:

1. System signals error to the manager, records the error, and enters a clean state
2. Manager starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation and enters the ID to retrieve the sale.
2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

1. System signals error to the manager
2. manager probably starts new sale and re-enters all items.
3. Manager continues with sale (probably entering more items or handling payment).

2a. customer enters the address, order-id and order will be tracked.

2b. if order-id is invalid then user will not be able to track the order.

**Special Requirements:**

. Text must be visible from 1 meter.

- user authorization response within 30 seconds 90% of the time.

- system fails then recover immediately

- Language internationalization on the text displayed.

-use color theme that is easily visible

- Id and password must require registering the account

-Use at least one capital letter in login

-Password shows as a hidden digit.

-If wrong entry at id or password uses red bar below the text

**Technology and Data Variations List**:

1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment signature captured on paper receipt
6. Use java language
7. Design use cases in NetBeans software using GUI swing.

**Open Issues:**

* Login failures
* Bad user experience
* Missing products information
* Complex check-out process
* Payment failures
* Not having flexible return policy

| **Use Case UC5: Process sale** |
| --- |
| **Scope**: Construction and Materials Rent App  **Level**: User goal  **Primary** **Actor**: Cashier  **Stakeholders and Interests**:  Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her  salary.  - Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her  salary.  - Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her  salary.  - Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her  salary.  - Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of ordered items and prices.  - Company: wants to ensure that order is successfully placed by the customer.  - Manager: Wants to be able to quickly perform override operations, and easily debug customer problems.  **Preconditions**  Cashier is identified and authenticated |

**Success Guarantee** (or Postconditions):

Receipt is generated. Payment authorization approvals are recorded.

Sale is saved. Tax is correctly calculated. Accounting and Inventory are

updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded

Sale is saved. Tax is correctly calculated. Accounting and Inventory are

updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded

Sale is saved. Tax is correctly calculated. Accounting and Inventory are

updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded

Sale is saved. Tax is correctly calculated. Accounting and Inventory are

updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recordedSale is saved. Receipt is generated. Payment authorization approvals are recorded.

**Main Success Scenario (or Basic Flow):**

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules.
5. System presents total with taxes calculated.
6. Cashier tells Customer the total and asks for payment.
7. Customer pays and System handles payment.
8. System logs completed sale and sends sale and payment information to the external accounting system (for accounting and commissions) and Inventory system (to update inventory).
9. System presents receipt.
10. Customer leaves with receipt and goods.

**Extensions (or Alternative Flows):**

\*a. At any time, Manager requests an override operation:

1. System enters Manager-authorized mode.

2. Manager or Cashier performs one Manager-mode operation. e.g., cash balance change, resume a suspended sale on another register, void a sale, etc.

3. System reverts to Cashier-authorized mode.

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Cashier restarts System, logs in, and requests recovery of prior state.

2. System reconstructs prior state.

2a. System detects anomalies preventing recovery:

1. System signals error to the Cashier, records the error, and enters a clean state.

2. Cashier starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation and enters the ID to retrieve the sale.

2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

1. System signals error to the Cashier.

2. Cashier probably starts new sale and re-enters all items.

3. Cashier continues with sale (probably entering more items or handling payment).

**Special Requirements:**

. Text must be visible from 1 meter.

- user authorization response within 30 seconds 90% of the time.

- system fails then recover immediately

- Language internationalization on the text displayed.

-use color theme that is easily visible

- Id and password must require registering the account

-Use at least one capital letter in login

-Password shows as a hidden digit.

-If wrong entry at id or password uses red bar below the text

**Technology and Data Variations List**:

1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment signature captured on paper receipt
6. Use java language
7. Design use cases in NetBeans software using GUI swing.

**Open Issues:**

* Login failures
* Bad user experience
* Missing products information
* Complex check-out process
* Payment failures
* Not having flexible return policy

| **Use Case UC6: Payment** |
| --- |
| **Scope**: Construction and Materials Rent App  **Level**: User goal  **Primary** **Actor**: Customer, Manager  **Stakeholders and Interests**:  - Customer: will pay and rent order  - Manager: will receive payment  **Preconditions**  User is identified and authenticated. |

**Success Guarantee** (or Postconditions):

Payment is done by customer successfully and order is purchased.

**Main Success Scenario (or Basic Flow):**

1.Customer place an order and requests for the payment.

2.System displays the payment method to the customers.

3.System allows the customer to pay via cards or cash on delivery.

4.Customer pays for the ordered items and Manager will receive the payment.

**Extensions (or Alternative Flows):**

\*a. At any time, Manager will manage all the details of system

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. manager restarts System, logs in, and requests recovery of all data
2. System recovers all the information

2a. System detects anomalies preventing recovery:

1. System signals error to the manager, records the error, and enters a clean state
2. Manager starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation and enters the ID to retrieve the sale.
2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

1. System signals error to the manager
2. manager probably starts new sale and re-enters all items.
3. Manager continues with sale (probably entering more items or handling payment).

2a. Customer gives payment through the card and card is valid then order will be placed successfully.

2b. if card is invalid or there is not enough amount in the card then customer will try another method.

**Special Requirements:**

. Text must be visible from 1 meter.

- user authorization response within 30 seconds 90% of the time.

- system fails then recover immediately

- Language internationalization on the text displayed.

-use color theme that is easily visible

- Id and password must require registering the account

-Use at least one capital letter in login

-Password shows as a hidden digit.

-If wrong entry at id or password uses red bar below the text

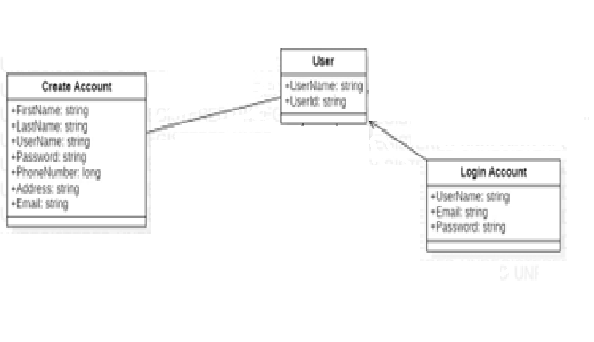
**Technology and Data Variations List**:

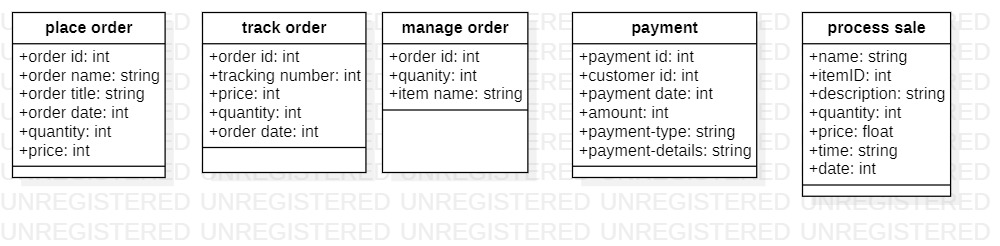
1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment signature captured on paper receipt
6. Use java language
7. Design use cases in NetBeans software using GUI swing.

**Open Issues:**

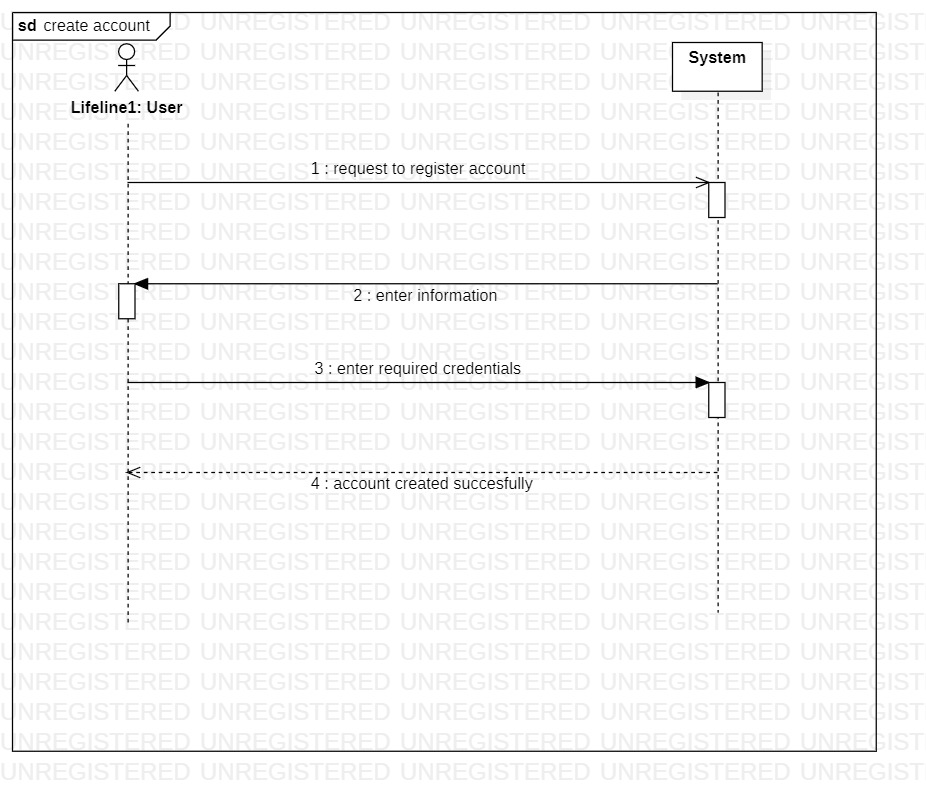
* Login failures
* Bad user experience
* Missing products information
* Complex check-out process
* Payment failures
* Not having flexible return policy

# **CHAPTER 3 Domain Model**





# **CHAPTER 4 System Sequence Diagram**



Table

Description automatically generated with low confidenceDiagram, table

Description automatically generated with medium confidenceTable

Description automatically generatedTable

Description automatically generated

Table

Description automatically generated

# **CHAPTER 5 Operation Contracts**

|  |  |
| --- | --- |
| Contract CO 1: Enter information | |
| Operation | Enter information required |
| Cross References: | Use Cases: Register Account |
| Preconditions: | User is not registered |
| Postconditions: | Account is successfully registered |

|  |  |
| --- | --- |
| Contract CO 2: Enter required credentials | |
| Operation | Enter required credentials |
| Cross References: | Use Cases: register account |
| Preconditions: | User is not registered |
| Postconditions: | User account is successfully created |

|  |  |
| --- | --- |
| Contract CO 3: Request login | |
| Operation | Request login |
| Cross References: | Use Cases: login |
| Preconditions: | Account is created |
| Postconditions: | User is logged in successfully |

|  |  |
| --- | --- |
| Contract CO 4: Enter valid credentials | |
| Operation | Enter valid credentials |
| Cross References: | Use Cases: login |
| Preconditions: | User requests to login |
| Postconditions: | User is successfully logged in |

|  |  |
| --- | --- |
| Contract CO 5: User request to place order | |
| Operation | user request to place order |
| Cross reference | Use Case: Place order |
| Preconditions | System checks that Stock is available to place order |
| Postconditions | User view and select items |

|  |  |
| --- | --- |
| Contract CO 6: User view and select items | |
| Operation | user view and select items |
| Cross reference | Use Case: Place order |
| Preconditions | System check that the user is verify and stock is available to select items |
| Postconditions | Ordered place successfully |

|  |  |
| --- | --- |
| Contract CO 7: Enter valid credentials | |
| Operation | user request to manage order |
| Cross reference | Use Cases: Manage order |
| Preconditions | System checks the availability of order to be managed |
| Postconditions | users select items to be managed |

|  |  |
| --- | --- |
| Contract CO 8: User request to track order | |
| Operation | user request to track order |
| Cross reference | Use Cases: Track order |
| Preconditions | User must have a tracking id |
| Postconditions | Provide tracking details |

|  |  |
| --- | --- |
| Contract CO 9: Provide tracking details | |
| Operation | Provide tracking details |
| Cross reference | Use Cases: Track order |
| Preconditions | User must have a tracking id |
| Postconditions | Order track successfully |

|  |  |
| --- | --- |
| Contract CO 10: User request for payment | |
| Operation | user request for payment |
| Cross reference | Use Cases: Payment |
| Preconditions | User must have online account for payment |
| Postconditions | users select payment method |

|  |  |
| --- | --- |
| Contract CO 11: User select payment method | |
| Operation | users select payment method |
| Cross reference | Use Cases: Payment |
| Preconditions | User must have online account for payment |
| Postconditions | Payment is done |

|  |  |
| --- | --- |
| Contract CO 12: Make Payment | |
| Operation | make Payment (amount: Money) |
| Cross reference | Use Cases: Process Sale |
| Preconditions | There is a sale underway |
| Postconditions | * A payment instance p was created * p was associated with current sale * The current sale was associated with the   store |