**COMSATS University Islamabad, Abbottabad Campus**

**Department of Computer Science**

**Project Proposal**

**Construction Equipment Renting Store**

**CSC392 Object Oriented Software Engineering**

Group Members:

Mubeen Khalid (SP21-BSE-015)

Mugheer Islam Khattak (SP21-BSE-017)

Zain Arshad (FA19-BSE-082)

Table of Contents

[CHAPTER 1 PROJECT PROPOSAL 5](#_Toc123230498)

[Introduction: 6](#_Toc123230499)

[Construct ease is the e-commerce store where people can buy or rent any construction related equipment or material with ease. The “Constructease” is derived from Construction and easy. 6](#_Toc123230500)

[Vision and Business Case 6](#_Toc123230501)

[Use-Case Model 6](#_Toc123230502)

[CHAPTER 2 USE CASES 5](#_Toc123230503)

[Use Case Diagram 6](#_Toc123230504)

[Brief Case 6](#_Toc123230505)

[Mubeen Khalid (SP21-BSE-015) 6](#_Toc123230506)

[CHAPTER 2 Fully Dressed Use Case 5](#_Toc123230507)

[Use Case UC1: Add Items 6](#_Toc123230508)

[Use Case UC1: Search Items 6](#_Toc123230509)

[Use Case UC1: Update items 6](#_Toc123230510)

[Use Case UC1: Delete Items 6](#_Toc123230511)

[Chapter 3 System Sequence Diagram: 5](#_Toc123230512)

[Add items: 6](#_Toc123230513)

[Search Items: 6](#_Toc123230514)

[Update Items: 6](#_Toc123230515)

[Delete items: 6](#_Toc123230516)

[Mugheer Islam Khattak (SP21-BSE-017) 5](#_Toc123230517)

[Use Case: Rent Equipment 6](#_Toc123230518)

[Use Case: Rent Agreement 6](#_Toc123230519)

[Use Case: Add to Cart 6](#_Toc123230520)

[CHAPTER 2 Fully Dressed Use Case 5](#_Toc123230521)

[Use Case UC1: Rent Equipment: 6](#_Toc123230522)

[Use Case UC1: Rent Agreement: 6](#_Toc123230523)

[Use Case UC1: Add to Cart: 6](#_Toc123230524)

[Chapter 3 System Sequence Diagram: 5](#_Toc123230525)

[Rent Equipment: 6](#_Toc123230526)

[Rent Agreement: 6](#_Toc123230527)

[Add To Cart: 6](#_Toc123230528)

[Zain Arshad (FA19-BSE-082) 5](#_Toc123230529)

[Use Case: Create Account 6](#_Toc123230530)

[Use Case: Login Account 6](#_Toc123230531)

[Use Case: Deliver order 6](#_Toc123230532)

[Use Case: Ship Order 6](#_Toc123230533)

[Use Case: Payment 6](#_Toc123230534)

[Use Case: Return item 6](#_Toc123230535)

[Chapter 2 Fully Dressed Use case 6](#_Toc123230536)

[Use Case UC1: Create Account 6](#_Toc123230537)

[Use Case UC3: Return Items 6](#_Toc123230538)

[Use Case UC4: Track Order 6](#_Toc123230539)

[Use Case UC5: Process sale 6](#_Toc123230540)

[Use Case UC6: Payment 6](#_Toc123230541)

[Chapter 3 System Sequence Diagram: 6](#_Toc123230542)

[Create Account 6](#_Toc123230543)

[Login: 6](#_Toc123230544)

[Return Items: 6](#_Toc123230545)

[Payment: 6](#_Toc123230546)

[Track Order: 6](#_Toc123230547)

[Process Sale: 6](#_Toc123230548)

[Chapter 4 Domain Model 6](#_Toc123230549)

[Chapter 5 Contracts: 6](#_Toc123230550)

[Mubeen Khalid SP21-BSE-015: 6](#_Toc123230551)

[Mugheer Islam Khattak SP21-BSE-017: 6](#_Toc123230552)

[Zain Arshad 6](#_Toc123230553)

[Chapter 6 Package Diagram: 5](#_Toc123230554)

[Chapter 7 Interaction Diagrams 5](#_Toc123230555)

[Zain Arshad (FA19-BSE-082): 5](#_Toc123230556)

[Login 6](#_Toc123230557)

[Chapter 8 Class Diagram 5](#_Toc123230558)

[Zain Arshad (FA19-BSE-082) 5](#_Toc123230559)

[Login: 5](#_Toc123230560)

# CHAPTER 1 PROJECT PROPOSAL

## Introduction:

### Construct ease is the e-commerce store where people can buy or rent any construction related equipment or material with ease. The “Constructease” is derived from Construction and easy.

### Vision and Business Case

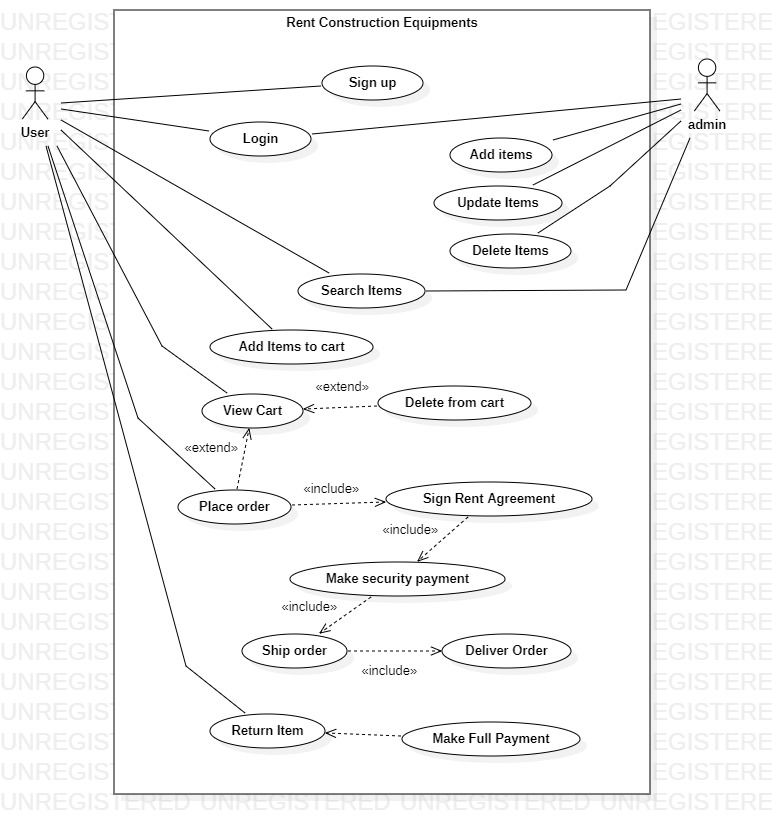
Constructease is a startup ecommerce store that sells construction related material and equipment. Rental services are also provided by Constructease. All the material and the equipment on the store is top notch with high sustainability. The products are of high quality and the startup offers money back guarantee for the customers in case they don’t like the product or are not satisfied with the product. Constructease is sold exclusively online; no retailers will be carrying any Constructease products.

## Use-Case Model

* The system must allow users to sign up before logging into the website by providing the necessary details. (To make any purchase log in is a must and before logging in, sign up is a must).
* The system must allow users to log into their accounts by entering their email and password before making any purchase. (Login will allow users to roam the website using their profile which they created).
* The system must allow the admin to add, remove and update the items at any time. (At any point in time the admin will be able to add, remove or update the products from the admin panel).
* The system must allow the admin to view the stock remaining.
* The system must allow users to view the products and roam in without logging in.
* The system must allow users to view the categories and select one according to their desired product.
* The system must allow users to search for their desired product on the site if it is available otherwise tell them that it is not available.
* The system must allow users to select multiple products and add them to the cart at once. (Multiple selections of items at one time).
* The system must allow users to view the cart at any time. (Cart will contain the items which the user selected during the scrolling phase).
* The system must allow users to write a review regarding the product.
* The system must allow the admin to view the review and take steps accordingly.
* The system must allow users to make orders for the items in the cart.
* The system must create a unique id against every id and assign that to the customer.
* The system must tell the customer to fill out a compulsory form before completing the order and doing the payment.
* The system must perform the form validation to make sure that the details are correct or not.
* The system must allow the customer to do the payment via credit card or cash on delivery (if cash on delivery is selected the customer must pay some amount at the order confirmation time).g

# CHAPTER 2 USE CASES

## Use Case Diagram



## Brief Case

### Mubeen Khalid (SP21-BSE-015)

#### Use Case: Add Items

Use case “Add Items” is a sub-function of “Manage Items” handled by the admin. The admin can add any item by using this use case to the system. The items with additional details like quantity, quality, image, and prices will be added to the system. No other user rather than the admin can have access to this.

#### Use Case: Delete Items

Case in point "Delete Items" is a sub-function of "Manage Items," which is managed by the administrator. Using this use case, the administrator can delete any object from the system. Items containing additional information such as number, quality, picture, and the price will be removed from the system. This information is only accessible to the administrator.

#### Use Case: Update items

As an instance, "Update Items" is a sub-function of the administrator-managed "Manage Items." The administrator can utilize this use case to update any item in the system. Items with extra information, such as number, quality, image, and price, can be changed in the system. Only the administrator has access to this information.

#### Use Case: Search Items

"Search Items," for example, is a sub-function of the administrator-managed "Manage Items." The administrator can utilize this use case to search the system for any required object. Items with additional information, such as number, quality, image, and price, may be found and analyzed in the system. This information is only accessible to the administrator.

# CHAPTER 2 Fully Dressed Use Case

| Use Case UC1: Add Items |
| --- |

**Scope**: Construction Equipment Renting Store

**Level**: Subfunction

**Primary** **Actor**: Admin

**Stakeholders and Interests**:

* Owner of the system wants to add items for sale/rent on a system with ease so that he/she will be able to sell his/her products.
* Customer: Wants items to view on the application.

**Preconditions**:

* The user must Sign up.
* The user must log in to the admin panel before adding any items.

**Success Guarantee** (or Postconditions): The items would have been added to the respective categories, the list is updated the and item is visible for sale.

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

* The user requests the system to show what items have been added to it.
* The system retrieves the list of items updated and displays it to the user.
* The user selects a category (buy/rent) to add any item.
* The system confirms that the item has been added.

**Extensions (or Alternative Flows):** If at any time the system is unable to retrieve, record, or provide details then the system informs the user of the problem, attempts to record the time and nature of the failure and the use case ends.

**Special Requirements:**

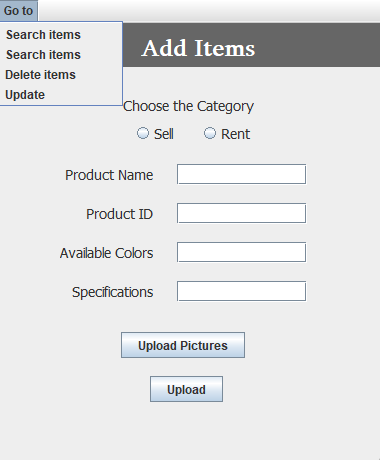
* Colors and sizes of text fonts used must provide - or be able to provide - for the visually impaired (e.g., color blindness).
* Items must be uploaded with high-quality pictures for the quality variations for the user to view

**Frequency of Occurrence:** Could be nearly continuous.

**Open Issues:**

* Do we have a full list of items for every category?
* Do we have a full list of items for every category?

**Screen Shots:**



|  |
| --- |

| Use Case UC1: Search Items |
| --- |

**Scope**: Construction Equipment Renting Store

**Level**: Subfunction

**Primary** **Actor**: Admin

**Stakeholders and Interests**:

Owner of the system: Wants to search items available on the system with ease so that he/she will be able to confirm whether the product is available or not.

**Preconditions**:

* The user must Sign up.
* The user must log in to the admin panel before adding any items.

**Success Guarantee** (or Postconditions): The items available on the system would be visible to the user.

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

* The user requests the system to show whether the searched items are available or not.
* The system retrieves the list of items updated and displays it to the user.
* The user selects a category (sell/rent) to search for any item.
* The system confirms that the items are available.

**Extensions (or Alternative Flows):**

**Special Requirements:**

* Colors and sizes of text fonts used must provide - or be able to provide - for the visually impaired (e.g., color blindness).
* All data related to the item selected for deletion must not be visible to the user.

**Frequency of Occurrence:** Could be nearly continuous.

**Open Issues:**

* Do any items have minimum capacity issues?
* Do we have a full list of items for every category?

**Screen Shots:**

**Graphical user interface

Description automatically generated**

| Use Case UC1: Update items |
| --- |

**Scope**: Construction Equipment Renting Store

**Level**: Subfunction

**Primary** **Actor**: Admin

**Stakeholders and Interests**: Owner of the system wants to update items for sale/rent on a system with ease so that he/she will be able to edit product details.

**Preconditions**:

* The user must Sign up.
* The user must log in to the admin panel before adding any items.

**Success Guarantee** (or Postconditions): The items would have been updated in the respective categories, the list is updated and so the items.

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

* The user requests the system to show what items have been updated.
* The system retrieves the list of items updated and displays it to the user.
* The user selects a category (buy/rent) to update any item.
* The system confirms that the item has been updated.

**Extensions (or Alternative Flows):** If at any time the system is unable to retrieve, record, or provide details then the system informs the user of the problem, attempts to record the time and nature of the failure and the use case ends.

**Special Requirements:**

* Colors and sizes of text fonts used must provide - or be able to provide - for the visually impaired (e.g., color blindness).
* All data related to the item selected for modification must be visible to the user.

**Frequency of Occurrence:** Could be nearly continuous.

**Open Issues:**

* do any items have minimum/maximum capacity issues?
* do we have a full list of items for every category?

**Screen Shots:**

**Graphical user interface, application

Description automatically generated**

|  |
| --- |

| Use Case UC1: Delete Items |
| --- |

**Scope**: Construction Equipment Renting Store

**Level**: Subfunction

**Primary** **Actor**: Admin

**Stakeholders and Interests**: Owner of the system wants to delete items for sale/rent on a system with ease so that he/she will be able to delete/remove products.

**Preconditions**:

* The user must Sign up.
* The user must log in to the admin panel before adding any items.

**Success Guarantee** (or Postconditions): The items would have been deleted from the respective categories, the list is updated the and item is not visible.

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

The user requests the system to show what items have been deleted from it.

The system retrieves the list of items updated and displays it to the user.

The user selects a category (buy/rent) to delete any item.

The system confirms that the item has been deleted.

**Extensions (or Alternative Flows):** If at any time the system is unable to retrieve, record, or provide details then the system informs the user of the problem, attempts to record the time and nature of the failure and the use case ends.

**Special Requirements:**

* Colors and sizes of text fonts used must provide - or be able to provide - for the visually impaired (e.g., color blindness).
* All data related to the item selected for deletion must not be visible to the user.

**Frequency of Occurrence:** Could be nearly continuous.

**Open Issues:**

* do any items have minimum capacity issues?
* do we have a full list of items for every category?

**Screen Shots:**

**Graphical user interface, application

Description automatically generated**

# Chapter 3 System Sequence Diagram:

## Add items:

Diagram, table

Description automatically generated

## Search Items:

Table

Description automatically generated

## Update Items:

Table

Description automatically generated

## Delete items:

Table

Description automatically generated

# Mugheer Islam Khattak (SP21-BSE-017)

## Use Case: Rent Equipment

Case in point, "Rent Equipment" is a sub-function of the system user-managed "Select Categories". Using this use case, the system user will be directed to the equipment section where the user will be able to select any Equipment for rent by fulfilling the terms and conditions followed by that equipment and then add it to the cart, and then followed by order completion. In the system, items lying under the Rent Equipment category will be located and viewed by the user. Both the administrator and any system user have access to this information.

## Use Case: Rent Agreement

Case in point, "Rent Agreement" is a sub-function of the system user-managed "Rent Equipment". Using this use case, the system user will have to accept the terms and conditions followed by that equipment. In the system, items lying under the Rent Equipment category will require the completion of the rental agreement before proceeding to the cart. Both the administrator and any system user have access to this information.

## Use Case: Add to Cart

“Add to cart” is the main function of the system where the number of items selected by the user will be added. The user will not be able to make any order before proceeding with to the cart. The customer will have access to this in case he/she wants to purchase or rent anything.

# CHAPTER 2 Fully Dressed Use Case

## Use Case UC1: Rent Equipment:

**Scope**: Construction Equipment Renting Store

**Level**: Subfunction

**Primary** **Actor**: System User

**Stakeholders and Interests**:

* User of the system wants to select the desired category of items available for rent on the system with ease so that he/she will be able to rent the equipment.

**Preconditions**:

The user must select a category: Rent Equipment.

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

* The user requests the system to show items by choosing the desired category.
* The system retrieves the list of items and displays it to the user.
* The system confirms that the item is available.

**Extensions (or Alternative Flows):**

If at any time the system is unable to retrieve, record, or provide details then the system informs the user of the problem, attempts to record the time and nature of the failure and the use case ends.

**Special Requirements:**

* Colors and sizes of text fonts used must provide - or be able to provide - for the visually impaired (e.g., color blindness).
* All data related to the item selected for Renting must be visible to the user.

**Frequency of Occurrence:** Could be nearly continuous.

**Open Issues:**

Do we have a full list of items for every category?

## Use Case UC1: Rent Agreement:

**Scope**: Construction Equipment Renting Store

**Level**: Subfunction

**Primary** **Actor**: System User

**Stakeholders and Interests**:

* User of the system wants to select the desired category of items available for rent on the system with ease so that he/she will be able to rent the equipment by fulfilling the given agreement.

**Preconditions**:

The user must select any equipment for rent.

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

* The user requests the system to show the agreement required to rent the selected equipment.
* The system confirms that the item is available for rent.
* The user selects the quantity of equipment to rent.

**Extensions (or Alternative Flows):**

If at any time the system is unable to retrieve, record, or provide details then the system informs the user of the problem, attempts to record the time and nature of the failure and the use case ends.

**Special Requirements:**

* Colors and sizes of text fonts used must provide - or be able to provide - for the visually impaired (e.g., color blindness).
* The agreement related to the item selected for the rental purpose must be visible to the user.

**Frequency of Occurrence:** Could be nearly continuous.

**Open Issues:**

Do we have a full list of items for every category?

Do we have the quantity of the equipment chosen by the user for rent?

## Use Case UC1: Add to Cart:

**Scope**: Construction Equipment Renting Store

**Level**: Main function

**Primary** **Actor**: System User

**Stakeholders and Interests**:

* User of the system wants to select the number of items chosen to rent or purchase on the system with ease so that he/she will be able to rent/purchase the items.

**Preconditions**:

the user must have selected any material for purchase or equipment for rent.

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

* The user selects the quantity of material or equipment.
* The system grants the quantity required by the user and adds it to the cart.

**Extensions (or Alternative Flows):**

If at any time the system is unable to retrieve, record, or provide details then the system informs the user of the problem, attempts to record the time and nature of the failure and the use case ends.

**Special Requirements:**

* Colors and sizes of text fonts used must provide - or be able to provide - for the visually impaired (e.g., color blindness).

**Frequency of Occurrence:** Could be nearly continuous.

**Open Issues:**

Do we have a full list of items for every category?

Do we have the quantity of the item?

# Chapter 3 System Sequence Diagram:

## Rent Equipment:

Table

Description automatically generated

## Rent Agreement:

Table

Description automatically generated

## Add To Cart:

Table

Description automatically generated

# Zain Arshad (FA19-BSE-082)

## 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 Post conditions): 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 Post conditions): 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 Post conditions):

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 Post conditions):

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**:

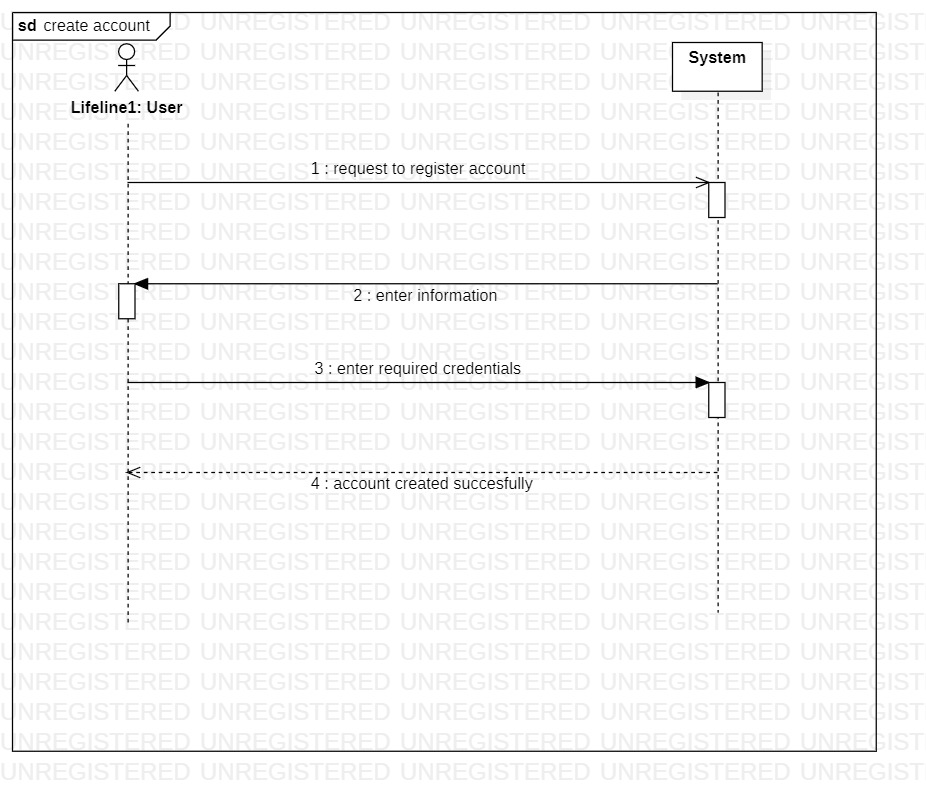
1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment 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 System Sequence Diagram:

### Create Account



### Login:

Table

Description automatically generated with low confidence

### Return Items:

Diagram, table

Description automatically generated with medium confidence

### Payment:

Table

Description automatically generated

### Track Order:

Table

Description automatically generated

### Process Sale:

Table

Description automatically generated

## Chapter 4 Domain Model

Diagram

Description automatically generated

## Chapter 5 Contracts:

## Mubeen Khalid SP21-BSE-015:

|  |  |
| --- | --- |
| Contract CO 1: Add Items | |
| Operation | addItems() |
| Cross References: | Use Cases: Add Items |
| Preconditions: | The admin should be logged in. |
| Postconditions: | SalesLineItem instance was created.  Sli was associated with view products.  Sli.quantity became quantity. |

|  |  |
| --- | --- |
| Contract CO 2: Delete Items | |
| Operation | deleteItems() |
| Cross References: | Use Cases: Delete Items |
| Preconditions: | The admin should be logged in. |
| Postconditions: | SalesLineItem instance was deleted.  Sli association was broken with view products.  Sli.quantity became quantity. |

|  |  |
| --- | --- |
| Contract CO 3: Update Items | |
| Operation | updateItems() |
| Cross References: | Use Cases: Update Items |
| Preconditions: | The admin should be logged in. |
| Postconditions: | SalesLineItem instance has already been created.  Sli was already associated with view products.  Sli.quantity became quantity.  sli was associated with a new ProductDescription, based on itemlD match (association formed). |

|  |  |
| --- | --- |
| Contract CO 4: Search Items | |
| Operation | searchItems() |
| Cross References: | Use Cases: Search Items |
| Preconditions: | The user/admin log in is not mandatory. |
| Postconditions: | SalesLineItem instance has already been created.  Sli was associated with view products based on itemID match. |

## Mugheer Islam Khattak SP21-BSE-017:

|  |  |
| --- | --- |
| Contract CO 1: Proceed to Rental | |
| Operation | proceedToRental() |
| Cross References: | Use Cases: Rent Equipment |
| Preconditions: | The user must be logged in.  Item to rent must be added into cart. |
| Postconditions: | Rent agreement instance was created.  RA was associated with specific product added in cart. |

|  |  |
| --- | --- |
| Contract CO 2: Rent agreement | |
| Operation | rentAgreement |
| Cross References: | Use Cases: Rent Agreement |
| Preconditions: | The user must be logged in.  Item to rent must be added into cart.  The user must select proceedtoRental. |
| Postconditions: | Rent agreement instance was already created.  RA was associated with orderId. |

|  |  |
| --- | --- |
| Contract CO 1: Add to cart | |
| Operation | addtocart(id,name) |
| Cross References: | Use Cases: Add to cart |
| Preconditions: | The user must be logged in.  Item to rent must be selected to add into cart. |
| Postconditions: | Cart instance was created.  SalesLineItem instance was created.  Sli was associated with cart.  Sli.quantity became quantity. |

## Zain Arshad

|  |  |
| --- | --- |
| 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 |

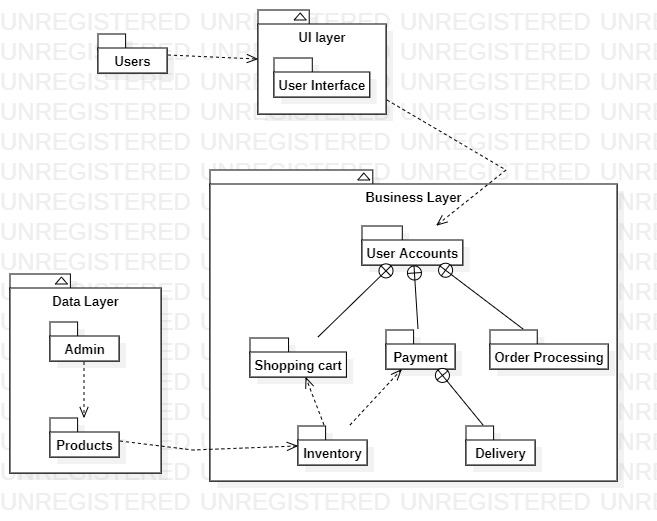
|  |  |
| --- | --- |
| Contract CO 9: Provide tracking details | |
| Operation | Provide tracking details |
| Cross reference | Use Cases: Track order |
| Preconditions | User must have a tracking id |
| Postconditions | Order track successfully |

|  |  |
| --- | --- |
| 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 |

# Chapter 6 Package Diagram:

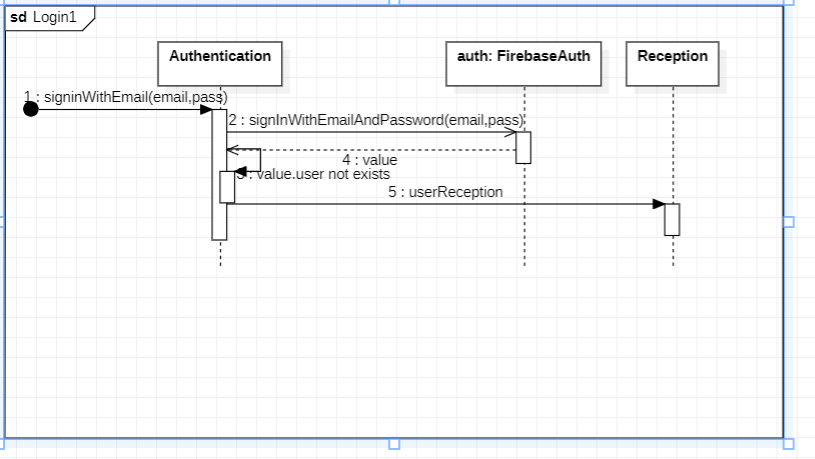


# Chapter 7 Interaction Diagrams

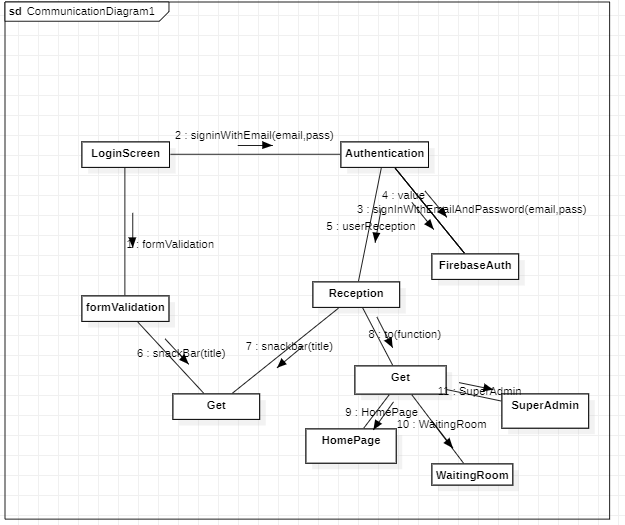
## Zain Arshad (FA19-BSE-082):

### Login

#### Sequence Diagram:



#### Communication Diagram:



# Chapter 8 Class Diagram

## Zain Arshad (FA19-BSE-082)

### Login:

