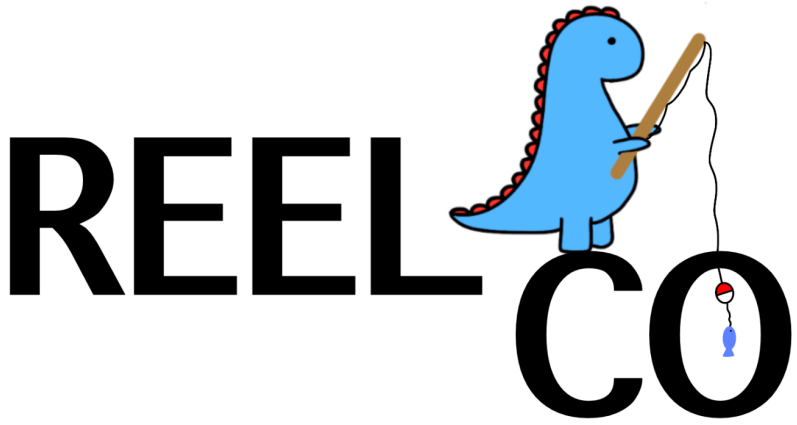
REEL CO

REEL COLOSET
Use-Case Specifications

Version <1.1>



Revision History

Date	Version	Description	Authors
<09/29/2022 - 09/30/2022>	<1.0>	First Draft	Claire Thompson, Olivia Romig, Ron Heminway, Libby Miller, Elise Lovell, Emmy Richardson
<10/02/2022>	<1.1>	Formatting Tweaks	Libby Miller & Emmy Richardson

Table of Contents

Figures

Use-Case Specifications

1. Use-Case Model

1.1 Introduction

The system involves weather-based outfit recommendations. It lets the consumer login and update what is in their closet and laundry hamper. User type is Consumer. A user accesses the system by providing a username and password previously defined by the user and stored in the database management system. The consumer uses the system through a website where they can access their virtual closet database and outfit recommendations. All data is stored in an external database.

1.2 General Actors Descriptions

1.2.1 Consumer

An actor that utilizes the services of the website.

1.2.2 API

A source to get the weather data needed for the program.

1.2.3 Database Management System

A place that safely and securely houses the closet and laundry hamper data.

1.2.4 REEL COLOSET System

A system that handles all interactions with the REEL COLOSET (RC) interface.

1.2.5 Login/Logout System

A system that handles logging the user in and out of the website.

1.3 Use-Case Model Hierarchy

1.3.1 Website Connections

- **Description**

The package is a container package, it houses Login/Logout Management and Interface Interactions.

- **Use Cases**
  - Login
  - Logout
  - Validate Query
  - Execute Query
  - Access Database
  - Confirm Login/Logout
  - Closet
  - Laundry
  - Create Account
  - Edit Profile
  - Daily Outfit
  - Login/Logout
- **Actors**
  - Consumer
  - Database Management System

- RC System

- **Relationships**

It is a container package that houses Login/Logout Management and Interface Interactions.

- **Packages Owned**
  - Login/Logout Management
  - Interface Interactions

1.3.1.1 Packages Diagram

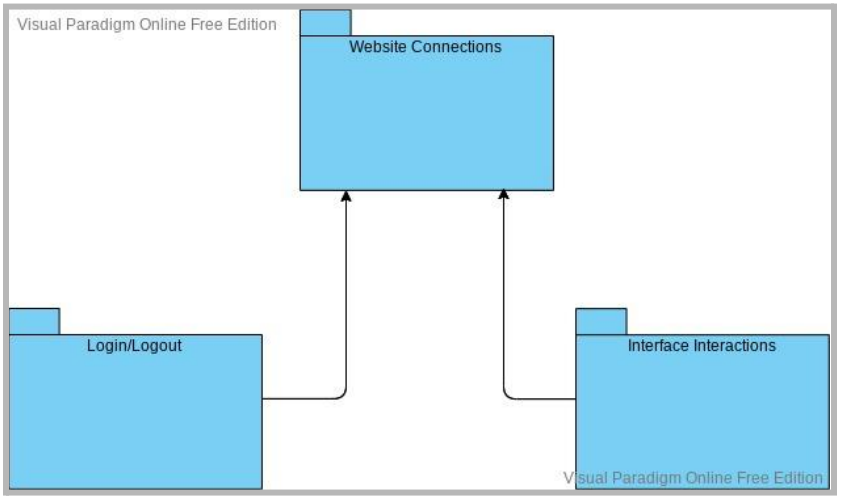


Figure 1 Website Connections/Use Packages

1.3.1.2 Login/Logout Management

- **Description**

  Lets the consumer login and logout of their accounts.
- **Use Cases**
  - Login
  - Logout
  - Validate Query
  - Execute Query
  - Access Database
  - Confirm Login/Logout
- **Actors**
  - Consumer
  - Database Management System
  - RC System
  - Weather API
  - Database
  - Login/Logout System
- **Relationships**
  - None
- **Packages Owned**
  - None

1.3.1.3 Interface Interaction Management

- **Description**
- **Use Cases**
  - Closet
  - Laundry
  - Create Account
  - Edit Profile
  - Daily Outfit
  - Login/Logout
- **Actors**
  - Consumer
  - Weather API
  - Database
  - Login/Logout System
- **Relationships**
  - None
- **Packages Owned**
  - None

1.4 Diagrams of the Use-Case Model

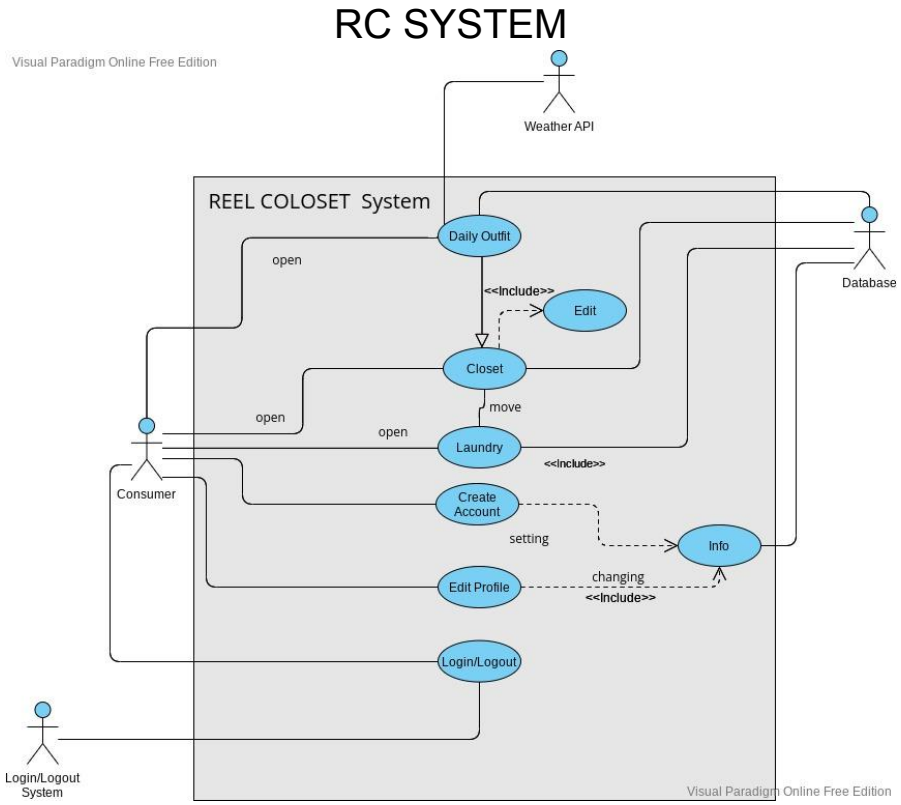


Figure 2 RC System

2. Create Account

2.1 Brief Description

The consumer creates their account and inputs their information in order to store their personal closet and laundry hamper.

2.2 Flow of Events

2.2.1 Basic Flow

The consumer loads the website and then navigates to the create account function. Once clicked the create account will prompt the user to input their email, user name, and password.

2.2.2 Alternative Flows

None

2.3 Special Requirements

2.3.1 < Having an Email >

The consumer needs to have an email to create an account.

2.4 Preconditions

2.4.1 < Internet Connection >

The consumer must be able to access the internet.

2.5 Postconditions

2.5.1 < Account Creation >

The consumer now has an account that they can access.

2.6 Extension Points

2.6.1 < Set Information - Include >

Once an account is created the consumer is prompted to add their information.

2.7 Relationships

The Create Account use case is connected to the consumer and has setting information included.

2.8 Use-Case Diagrams

Refer to Use-Case Diagram (Figure 2).

2.9 Other Diagrams

None.

3. Daily Outfit

3.1 Brief Description

The daily outfit is generated utilizing the weather API and the closet data stored on the database management system.

3.2 Flow of Events

3.2.1 Basic Flow

After the user selects the generate daily outfit option, the outfit is generated by combining the weather information from the API and the data stored in the consumer's closet. The outfit is displayed on the browser.

3.2.2 Alternative Flows

None.

3.3 Special Requirements

None.

3.4 Preconditions

3.4.1 < Pre-Existing Closet Information >

The consumer must have a closet so that an outfit can be generated.

3.5 Postconditions

3.5.1 < Consumer now has a weather approved outfit generated in browser >

The consumer will receive their outfit suggestion.

3.6 Extension Points

3.6.1 < Weather information - Include >

The Daily Outfit needs the weather information from the API to generate an accurate outfit.

3.7 Relationships

Daily outfit is connected to the consumer and includes the weather data.

3.8 Use-Case Diagrams

Refer to Use-Case Diagram (Figure 2).

3.9 Other Diagrams

None.

4. Laundry

4.1 Brief Description

The consumer's laundry is tracked in the database based on what clothes they wear each day.

4.2 Flow of Events

4.2.1 Basic Flow

The consumer inputs what they have worn into the application, and the information is stored in the database. Depending on the preferences that the consumer set, the application will send a reminder to the consumer to tell them to wash a certain item of clothing.

4.2.2 Alternative Flows

None.

4.3 Special Requirements

None.

4.4 Preconditions

4.4.1 < Pre Existing Closet >

In order to have the laundry feature function properly the consumer needs to have an accurate closet that
the laundry feature can use.

4.5 Postconditions

4.5.1 < Laundry Hamper will be updated >

The laundry hamper will be updated based on what the consumer has worn.

4.6 Extension Points

None.

4.7 Relationships

The laundry is connected to the database that houses the accounts, and laundry is connected to closet
by a move relationship to move the clothes from the laundry hamper to the closet.

4.8 Use-Case Diagrams

Refer to Use-Case Diagram (Figure 2).

4.9 Other Diagrams

None.

5. Edit Profile

5.1 Brief Description

This function lets the consumer edit the city they want clothing recommendations for as well as edit the originally stated profile information.

5.2 Flow of Events

5.2.1 Basic Flow

The consumer navigates to the edit profile tab and then clicks on the edit profile function. Once there the consumer will be prompted to update the information they want. Then the consumer will save their updates, and navigate back to the homepage.

5.2.2 Alternative Flows

None.

5.3 Special Requirements

None.

5.4 Preconditions

5.4.1 < Having a pre-existing account >

In order to edit the consumer's profile the consumer must first have a pre-existing account.

5.5 Postconditions

5.5.1 < Having an updated profile >

Now the consumer has a profile that is updated to their specifications.

5.6 Extension Points

5.6.1 < Information - Include >

Edit Profile includes the information that is housed in the database for the accounts.

5.7 Relationships

Edit Profile is connected to the consumer, changes, Login function, and the information in the account.

5.8 Use-Case Diagrams'

Refer to Use-Case Diagram (Figure 2).

5.9 Other Diagrams

None.

6. Closet

6.1 Brief Description

The consumer can input the items in their closet for the daily outfit to pick from items in their actual closet.

6.2 Flow of Events

6.2.1 Basic Flow

The consumer can input what is in their closet by selecting the general type of a clothing item (i.e. sweater, t-shirt, jeans, etc…) and then add specific notes for their piece of clothing. The consumer can also remove items that they no longer have.

6.2.2 Alternative Flows

None.

6.3 Special Requirements

None.

6.4 Preconditions

6.4.1 < Having an Account >

The consumer needs to have an account in order to have the closet feature.

6.5 Postconditions

6.5.1 < Updated Closet >

After adding or removing a piece of the closet is updated to accurately reflect the consumer's closet.

6.6 Extension Points

6.6.1 < Edit closet - Include >

Includes the function to edit the closet after the initial closet setup.

6.7 Relationships

The Closet is connected to the consumer and the database of account information, and is also connect to
laundry by a relationship to move clothing items back and forth.

6.8 Use-Case Diagrams

Refer to Use-Case Diagram (Figure 2).

6.9 Other Diagrams

None.

7. Login/Logout

7.1 Brief Description

Login/Logout facilitates the consumer to execute their desired event when leaving or entering their account through the login/logout system.

7.2 Flow of Events

7.2.1 Basic Flow

The user indicates that they want to either login or logout of their account. This will send them to the Login/Logout System to fill in the requested information

7.2.2 Alternative Flows

None.

7.3 Special Requirements

User who is logged in cannot interact with Login and vise versa

7.4 Preconditions

7.4.1 < Having an Account >

A consumer must have an account to access the Login/Logout System.

7.5 Postconditions

None.

7.6 Extension Points

None.

7.7 Relationships

Login/Logout is connected to the consumer and thus can take the consumer to the Login/Logout System

7.8 Use-Case Diagrams

Refer to Use-Case Diagram (Figure 2).

7.9 Other Diagrams

None.

Login/Logout System

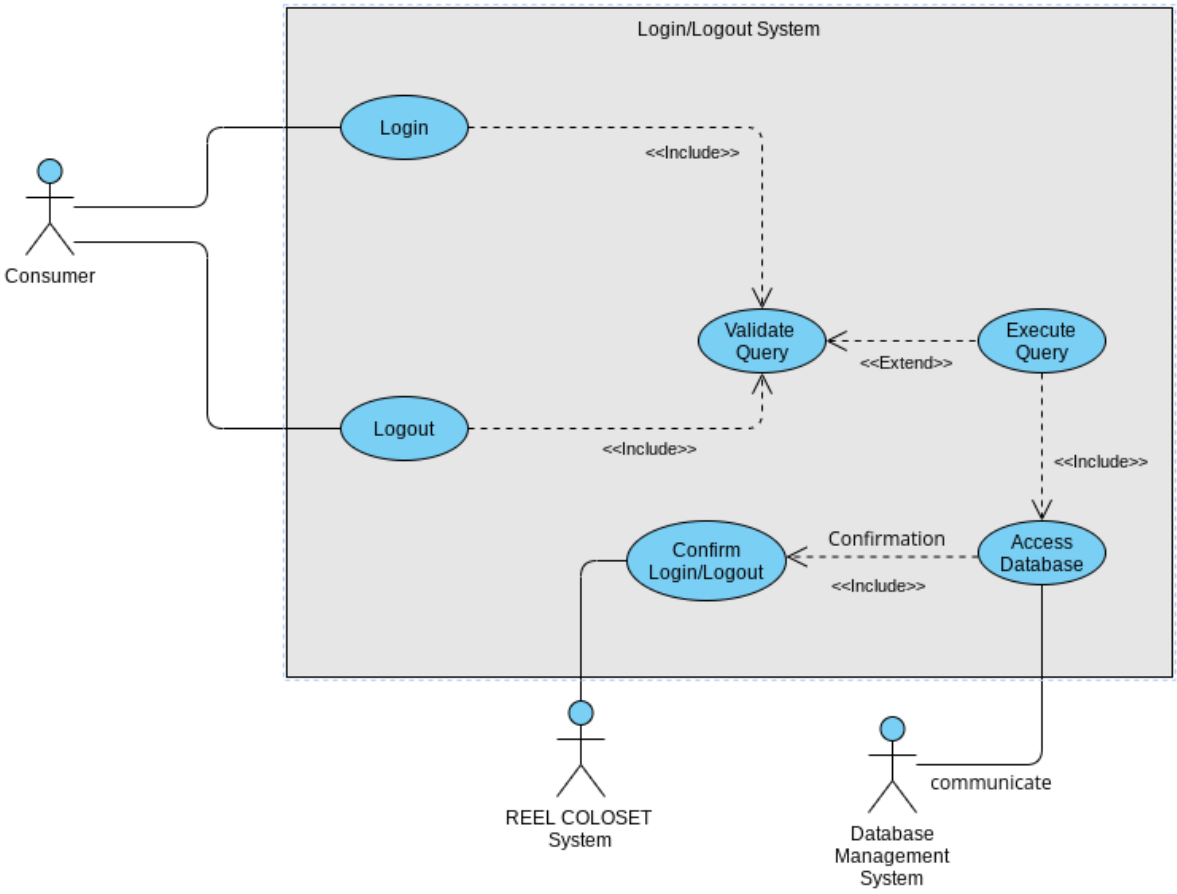


Figure 3 RC Login/Logout System

8. Login

8.1 Brief Description

Allows all consumers to access their RC accounts where their closet, laundry, and profile information are kept. Consumers will be prompted to enter their username and password to access their account and if successful will be able to interact with the website.

8.2 Flow of Events

8.2.1 Basic Flow

The consumer provides their username and password to the Login Interface and then submits it to the login process. The account is validated through a database check and then the results are returned to the consumer. If the correct username or password is entered then the user is sent to the REAL COLOSET System.

8.2.2 Alternative Flows

None.

8.3 Special Requirements

None.

8.4 Preconditions

8.4.1 < *Pre-Existing account* >

The consumer must already have an account to login.

8.4.2 < *Logged Out* >

The user isn't already logged in.

8.4.3 < *Possess user login information* >

The consumer must know their respective username and password associated with their account.

8.5 Postconditions

8.5.1 < *Account Validation* >

The username and password is checked and then if it exists the account is pulled up for the consumer.

8.6 Extension Points

8.6.1 < *Validate Query - include* >

The application validates the entered username and password.

8.7 Relationships

Login is connected to the consumer, and includes Validate Query, Execute Query, Access Database, and
Confirm Login/Logout.

8.8 Use-Case Diagrams

Refer to Use-Case Diagram (Figure 3).

8.9 Other Diagrams

None.

9. Logout

9.1 Brief Description

Allows the consumer to end their session with the website.

9.2 Flow of Events

9.2.1 Basic Flow

The consumer initiates the logout request through their GUI. Then after processing the request the system disconnects and logs the consumer out of the system.

9.2.2 Alternative Flows

None.

9.3 Special Requirements

None.

9.4 Preconditions

9.4.1 < Logged In already >

The consumer needs to be logged in in order to be logged out of the system.

9.5 Postconditions

9.5.1 < System Response >

The system will send a message to the client signally they have been logged out successfully, and is redirected to the homepage.

9.6 Extension Points

< *Validate Query - include* >

The application validates the entered username and password.

9.7 Relationships

Logout is connected to the Consumer, and includes Validate Query, Execute Query, Access Database, and
Confirm Login/Logout.

9.8 Use-Case Diagrams

Refer to Use-Case Diagram (Figure 3).

9.9 Other Diagrams

None.