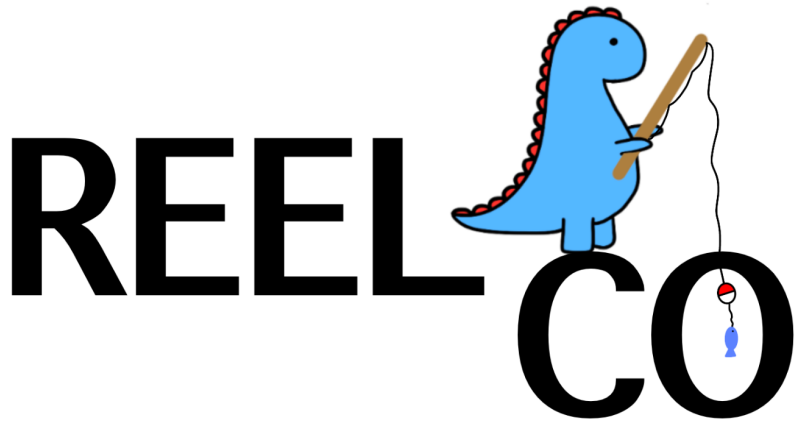
REEL CO

REEL COLOSET
Glossary

Version <1.1>



Revision History

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

Table of Contents

1. Introduction 4
   - 1.1 Purpose
   - 1.2 Scope
   - 1.3 References

2. Definitions 4
   - 2.1 ACK-NACK
   - 2.2 API
   - 2.3 CSS
   - 2.4 Field
   - 2.5 HTML
   - 2.6 HTTP
   - 2.7 IP
   - 2.8 JSON
   - 2.9 RC
   - 2.10 Snake Case
   - 2.11 SQL
   - 2.12 TCP 5

Glossary

1. Introduction

1.1 Purpose

The glossary contains the working definitions for the REEL COLOSET system. This glossary will be updated throughout the life of the project when new definitions are needed.

1.2 Scope

This glossary addresses all the terms which have specific meanings for this project. Actors are not included in the glossary as they are described in more detail in the use case definitions.

1.3 References

IEEE Std. 830-1998: IEEE Recommended Practice for Software Requirements Specifications

UPEDU : http://www.upedu.org/

Course Website : https://people.eecs.ku.edu/~saiedian/Teaching/448/

2. Definitions

2.1 ACK-NACK
Acknowledgement-Negative Acknowledgement; a message sent by the receiver to the transmitter to indicate if it has correctly or incorrectly received a data packet.

2.2 API
Application Programming Interface; a software system that allows two applications to interact with each other.

2.3 CSS
Cascading Style Sheets; programming language used to describe how HTML elements should be described.

2.4 Field
Is a box or section on the page that the consumer can interact and type into.

2.5 HTML
HyperText Markup Language; markup language that describes how elements should be displayed on a browser.

2.6 HTTP
Hypertext Transfer Protocol; application layer of the internet suite.

2.7 IP
Internet Protocol; communications protocol for network-connected computers.

2.8 JSON
JavaScript Object Notation; a data-interchange format used for representing data based on the syntax of a JavaScript object.

2.9 RC
REEL COLOSET

2.10 Snake Case
The naming convention where every word starts with a lowercase letter and each word is separated by an underscore, e.g. ‘variable_name’. In this convention, each space in English writing is represented as an underscore.

2.11 SQL
Structured Query Language; programming language used for managing and organizing databases.

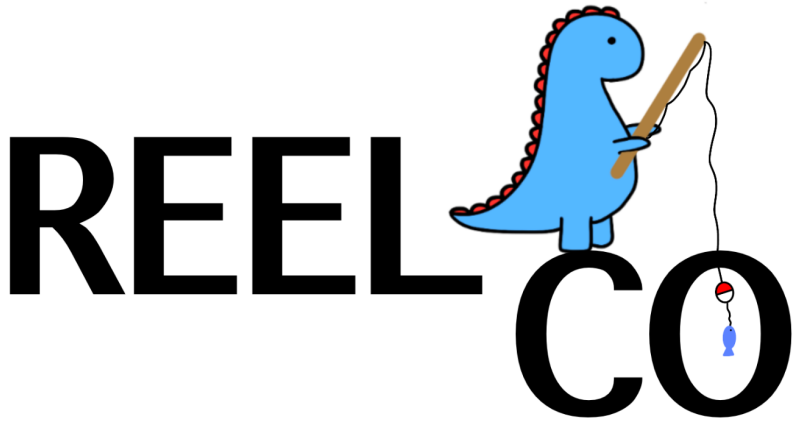
2.12 TCP

Transmission Control Protocol; a protocol to the internet suite.

REEL CO

REEL COLOSET
Software Requirements Specifications

Version <1.1>



Revision History

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

Table of Contents

1. Introduction 4
   - 1.1 Purpose
   - 1.2 Scope
   - 1.3 Definitions, Acronyms, and Abbreviations
   - 1.4 References 5
   - 1.5 Overview

2. Overall Description 6
   - 2.1 Product perspective
     - 2.1.1 System Interfaces
     - 2.1.2 User Interfaces
     - 2.1.3 Hardware Interfaces
     - 2.1.4 Software Interfaces 7
     - 2.1.5 Communication Interfaces
     - 2.1.6 Memory Constraints
     - 2.1.7 Operations
   - 2.2 Product functions
   - 2.3 User characteristics
   - 2.4 Constraints
   - 2.5 Assumptions and dependencies 8
   - 2.6 Requirements subsets

3. Specific Requirements 9
   - 3.1 Functionality
     - 3.1.1 < RC System >
     - 3.1.2 < Login/Logout System >
     - 3.1.3 < Weather API >
     - 3.1.4 < Database >
   - 3.2 Use-Case Specifications 10
   - 3.3 Supplementary Requirements

4. Classification of Functional Requirements 11

Software Requirements Specification

1. Introduction

1.1 Purpose

The purpose of this document is to describe the requirement specifications for an outfit generator based on the weather. The intended audience of this document includes the prospective developers of this outfit generator system.

1.2 Scope

The software system to be created is the REEL COLOSET, which will be referred to as "RC" throughout the rest of this document. RC allows consumers to create an account, input their city, and input the clothing items in their closet. Then RC will generate a daily outfit based on the weather of the city the consumer's account has inputted. This product is helpful for minimizing time selecting outfits as well as providing insightful advice for creating more functional outfits. This will apply to the RC Use-Case model from the Use-Case Specifications document.

1.3 Definitions, Acronyms, and Abbreviations

ACK-NACK

Acknowledgement-Negative Acknowledgement; a message sent by the receiver to the transmitter to indicate if it has correctly or incorrectly received a data packet.

API

Application Programming Interface; a software system that allows two applications to interact with each other.

CSS

Cascading Style Sheets; programming language used to describe how HTML elements should be described.

Field

A box or section on the page that the consumer can interact and type into.

HTML

HyperText Markup Language; markup language that describes how elements should be displayed on a browser.

IP

Internet Protocol; communications protocol for network-connected computers.

RC

REEL COLOSET

SQL

Structured Query Language; programming language used for managing and organizing databases.

TCP

Transmission Control Protocol; a protocol to the internet suite.

1.4 References

IEEE Std. 830-1998: IEEE Recommended Practice for Software Requirements Specifications

1.5 Overview

The rest of this document contains an overall description of the RC website system (section 2), and the specific requirements for the system (section 3). It will define the overall descriptions of each part of the document then list out all requirements needed for the product and finally classify all the functional requirements into categories.

2. Overall Description

2.1 Product perspective

This product was created to help people cut down the time it takes to create an outfit, and to reduce some of the stress involved in coming up with outfits that look stylish. Another purpose for this product involves the fact that the weather plays a big part in what a person should wear - RC takes the guesswork out of what the outfit needs in order to be consistent with the weather forecast of the day. It will use a database and an API to perform these tasks, which will interact with the RC System and the Login/Logout System.

2.1.1 System Interfaces

The RC web application to be developed is a software as a service accessible via web browser on the internet. It consists of four major components: a Login/Logout System, the RC System, a weather API, and a Database.

The Login/Logout System allows users to log into and out of the RC website. The Login/Logout System is connected to the database where user login information is stored. Once logged in, the user interacts with the RC System which allows the user to look at and edit their closet and move things to and from laundry as well as view their daily recommended outfit. The daily recommended outfit section of the RC system connects to both the database and the weather API, both of which are used to generate the outfit for the day. The database also connects to the RC System through the user info tab where the user can edit their account information that's stored in the database.

2.1.2 User Interfaces

The Login/Logout System and RC System must provide a user interface that is available through any web browser. The database and weather API do not have a user interface.

The Login/Logout System user interface shall have two fields for the user to input their username and password into. The text entered into the password field will be hidden and there will be a button to log in. There will also be a button for creating a new account. When the "create new account" button is pressed it will prompt the user for a username, password, email, and to select a city from a drop down menu. On this page there will be a button to finish making your account. It will then take the user to the RC System that is logged into their new account.

The RC System will have 4 separate tabs for the user profile, the closet, laundry, the daily outfit. There will also be a logout button in the top right of the page. On the user profile page there will be fields for all of the user’s information, which will include their username, email, selected city, and hidden password. There will be a button at the bottom to edit their profile, which will create text boxes under each field to enter their new information, and a button at the bottom to confirm changes. The closet will include a catalog of all items in the closet with icons and names for each item. There will also be a button that allows the user to add a new item. Add will prompt them with a template to create a new item. Each item will have an option to delete it or move it to the laundry. The laundry closet will also have a catalog of items, each of which will have an option to move it back to the closet. The Daily Outfit tab will show the current city selected, the current temperature, a written description of the current weather and each of the items selected from the user’s closet that it recommends for that day. When the logout button is selected it will take the user back to the Login/Logout Interface.

2.1.3 Hardware Interfaces

All components must be able to execute on a personal computer.

2.1.4 Software Interfaces

The Software Interfaces that are used by RC includes the weather API database interface and the Account Information database interface. The Software Interfaces are what the program will interact with to generate the daily outfit recommendations and keep the laundry hamper updated. The API is where the program gets its information as to what the weather is in a particular location which is used to base the program's daily outfits recommendations on. The database is used to securely store all personal information a user has connected to the site, which includes their login information, location, email and catalog of clothing items. All details of how these documents interact with the system can be found in the Use-Case Specifications document.

2.1.5 Communication Interfaces

All systems will be able to communicate with the server using a TCP/IP connection. The Server and Database will be located on the same host in order to communicate.

2.1.6 Memory Constraints

The primary memory constraints that are demanded by the RC Web Application demands less than 100 Mb of RAM.

2.1.7 Operations

The operation of the RC and Login/Logout Systems must be easy and intuitive for the average internet user. The user will initiate most operations that the systems will carry out. These operations are logging into and out of the website, creating a new account, editing profile information, adding and removing clothes from the closet, and moving clothes from closet to laundry and vice versa. The generated daily outfit operation is an unattended operation and will not require any user input. Backup operations will occur once a week and in the case of network failure, user machine failure, and database failure, the most recent backup will be restored. This will allow for most information to be recovered.

2.2 Product functions

The main functions of RC are viewing the daily generated outfit, updating the closet, and moving clothing items from the closet to the laundry hamper and vice versa. The daily generated outfit is based on the weather for the city that the consumer inputs into their account as well as the available items in the closet. Updating the closet allows the consumer to add new clothing items and make specific notes about the item as well as deleting an item if the consumer no longer owns it. The consumer can say which clothing items they wore that day and then those will get put into the laundry hamper. The database will store all personal information entered into the user's profile as well as a catalog of the items in their closet.

2.3 User characteristics

Users are the consumers of the product who are familiar with how the internet works and can navigate to a website. Consumers are knowledgeable of what their username and password is, as well as being familiar with what articles of clothing are in their closet.

2.4 Constraints

An API call will be made to the Weather API when an outfit is generated for the consumer, which

therefore accesses an interface to another application. Audit functions in the RC System include static code analysis that will be run in the main source file to highlight any bugs or security problems. There will be a set of overhead control functions that provide a global scope for the program but will not be fully reliant on them. The program will utilize a combination of Javascript, SQL, HTML, and CSS - this means that potential intersectionality between the coding programs will need to be addressed when problems arise, especially when combining with SQL.

Regarding signal handshake protocols, ACK-NACK data acknowledgement protocols will be pivotal in ensuring data from the API as well as the data servers will not be lost or dropped during transmission. An internet and live data server is also pivotal for consumers to properly interact and log in to the web application. Therefore, username and password setup are an important layer of security to protect user data.

2.5 Assumptions and dependencies

There is the assumption that the consumer is using a non-deprecated browser and has the internet knowledge to navigate a website.

2.6 Requirements subsets

2.6.1 A badge system to reward users based on their interaction with the app (i.e., “You’ve worn these jeans 5 times!”, “You’ve done laundry 3 times!”).
2.6.2 A travel feature where a user can input where they’re going and what events they’re attending.
2.6.3 Helpful clothing tips will be generated with each Daily Outfit.
2.6.4 A notification system that will send users a daily email containing their Daily Outfit.
2.6.5 A ‘Favorite’ option where users can favorite their most beloved clothing items so that they will be generated more often in their Daily Outfits.
2.6.6 A counter keeps track of how many times an article of clothing is worn, and will notify the user to wash an article after it is worn a specified number of times.

3. Specific Requirements

3.1 Functionality

3.1.1 < RC System>

3.1.1.1 The user shall be able to open the Daily Outfit tab.
3.1.1.2 The user shall be able to open the Closet Tab.
3.1.1.3 The user shall be able to delete a clothing item from the closet.
3.1.1.4 The user shall be able to add a clothing item to the closet.
3.1.1.5 The user shall be able to open the Laundry tab.
3.1.1.6 The user shall be able to open the Profile tab.
3.1.1.7 The user shall be able to edit their profile information.
3.1.1.8 The user shall be able to move their closet inventory into Laundry.
3.1.1.9 The user shall be able to click the 'Wash Clothes' button.
3.1.1.10 The system shall generate a guest session that does not require user authentication.
3.1.1.11 The user shall be able to accept/decline the generated outfit.
3.1.1.12 The user shall be able to create an account.
3.1.1.13 The user shall be able to create separate wardrobes for different events (i.e., formal wear).
3.1.1.14 The user shall be able to favorite certain clothing items.
3.1.1.15 The system shall alert the user when their hamper is full.
3.1.1.16 The user will be able to add specific details about an article of clothing.
3.1.1.17 The initial window of the Login/Logout System shall have a password and username field as well as a button labeled login. The password field shall be hidden. There will also be a button to create a new account.
3.1.1.18 The main window will have four tabs: Daily Outfit, Laundry, Closet, and Profile. It will also include a logout button.
3.1.1.19 The Daily Outfit tab will include the temperature, city, brief description of the weather, and an outfit suggestion for the day.
3.1.1.20 The Laundry tab will have a catalog of dirty clothes and a do laundry button.
3.1.1.21 The Closet Tab will have a catalog of clothing items as well as the ability to delete a specific item or move it to the laundry basket. There will also be an add item button to create a new item.
3.1.1.22 The Profile Tab will include fields that list the user's username, selected city, email, and hidden password. There will be an edit profile button that allows the user's to change their profile information by typing in the new information and hitting the confirm changes button.

3.1.2 < Login/Logout System>

3.1.2.1 The user shall be able to load the login page.
3.1.2.2 The user shall be able to enter their username & password.
3.1.2.3 The system shall authenticate the user and open their account.
3.1.2.4 The system shall log the user out of their account.
3.1.2.5 If the authentication is successful, the user shall be directed to the RC System.
3.1.2.6 If the authentication is unsuccessful, the system shall display an 'Invalid Username/Password' message.
3.1.2.7 When the login button is pressed a query shall be sent to login to the system.
3.1.2.8 When the logout button is pressed a query shall be sent to logout of the system.

3.1.3 < Weather API >

3.1.3.1 The API shall retrieve the weather for the user's location.
3.1.3.2 The API shall be accessed every hour to keep the current weather information up to date.

3.1.4 < Database>

3.1.4.1 The database shall store user closet data.
3.1.4.2 The database shall store the user location data.
3.1.4.3 The database shall store the consumer's username.
3.1.4.4 The database shall store the user's password.
3.1.4.5 The database shall store the user's email address.
3.1.4.6 The database shall store data regarding the user's laundry basket.
3.1.4.7 The database shall empty clothes in the laundry basket after the user clicks the 'Wash Clothes' button.
3.1.4.8 The database shall keep a temporary history of what the user has worn.
3.1.4.9 The database shall generate favorites items more frequently.
3.1.4.10 The database shall respond to query requests from the program.

3.2 Use-Case Specifications

3.2.1 RC System
3.2.1.1 Closet
3.2.1.2 Laundry
3.2.1.3 Create Account
3.2.1.4 Edit Profile
3.2.1.5 Daily Outfit
3.2.1.6 Login/Logout

3.2.2 Login/Logout System
3.2.2.1 Login
3.2.2.2 Logout
3.2.2.3 Validate Query
3.2.2.4 Execute Query
3.2.2.5 Access Database

For more information regarding use-case specifications, refer to the Use-Case Specifications document.

3.3 Supplementary Requirements

For further details on the Supplementary Requirements please refer to the Supplementary Specifications document.

4. Classification of Functional Requirements

Functionality	Type
3.1.1.1 The user shall be able to open the Daily Outfit tab.	Essential
3.1.1.2 The user shall be able to open the Closet Tab.	Essential
3.1.1.3 The user shall be able to delete a clothing item from the closet.	Essential
3.1.1.4The user shall be able to add a clothing item to the closet.	Essential
3.1.1.5 The user shall be able to open the Laundry tab.	Essential
3.1.1.6 The user shall be able to open the Profile tab.	Essential
3.1.1.7 The user shall be able to edit their profile information.	Essential
3.1.1.8 The user shall be able to move their closet inventory into Laundry.	Essential
3.1.1.9The user shall be able to click the ‘Wash Clothes’ button.	Essential
3.1.1.10 The system shall generate a guest session that does not require user authentication.	Desirable
3.1.1.11 The user shall be able to accept/decline the generated outfit.	Desirable
3.1.1.12 The user shall be able to create an account.	Essential

3.1.1.13 The user shall be able to create separate wardrobes for different events (i.e., formal wear).	Desirable
3.1.1.14 The user shall be able to favorite certain clothing items.	Desirable
3.1.1.15 The system shall alert the user when their hamper is full.	Desirable
3.1.1.16 The user will be able to add specific details about an article of clothing.	Desirable
3.1.1.17 The initial window of the Login/Logout System shall have a password and username field as well as a button labeled login. The password field shall be hidden. There will also be a button to create a new account.	Essential
3.1.1.18 The main window will have four tabs: Daily Outfit, Laundry, Closet, and Profile. It will also include a logout button.	Essential
3.1.1.19 The Daily Outfit tab will include the temperature, city, brief description of the weather, and an outfit suggestion for the day.	Essential
3.1.1.20 The Laundry tab will have a catalog of dirty clothes and a do laundry button.	Essential
3.1.1.21 The Closet Tab will have a catalog of clothing items as well as the ability to delete a specific item or move it to the laundry basket. There will also be an add item button to create a new item.	Essential
3.1.1.22 The Profile Tab will include fields that list the user's username, selected city, email, and hidden password. There will be an edit profile button that allows the user's to change their profile information by typing in the new information and hitting the confirm changes button.	Essential
3.1.2.1 The user shall be able to load the login page.	Essential
3.1.2.2 The user shall be able to enter their username & password.	Essential
3.1.2.3 The system shall authenticate the user and open their account.	Essential
3.1.2.4 The system shall log the user out of their account.	Essential

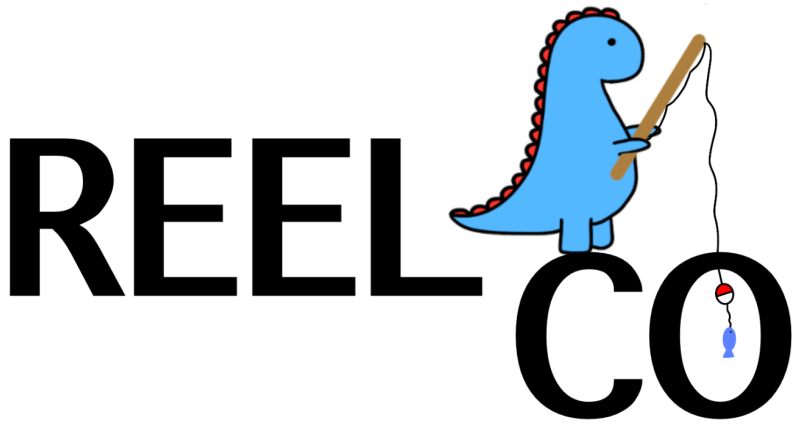
3.1.2.5 If the authentication is successful, the user shall be directed to the RC System.	Essential
3.1.2.6 If the authentication is unsuccessful, the system shall display an ‘Invalid Username/Password’ message.	Essential
3.1.2.7 When the login button is pressed a query shall be sent to login to the system.	Essential
3.1.2.8 When the logout button is pressed a query shall be sent to logout of the system.	Essential
3.1.3.1 The API shall retrieve the weather for the user’s location.	Essential
3.1.3.2 The API shall be accessed every hour to keep the current weather information up to date.	Essential
3.1.4.1 The database shall store user closet data.	Essential
3.1.4.2The database shall store the user location data.	Essential
3.1.4.3 The database shall store the consumer’s username.	Essential
3.1.4.4 The database shall store the user’s password.	Essential
3.1.4.5 The database shall store the user’s email address.	Essential
3.1.4.6The database shall store data regarding the user’s laundry basket.	Essential
3.1.4.7 The database shall empty clothes in the laundry basket after the user clicks the 'Wash Clothes’ button.	Essential
3.1.4.8 The database shall keep a temporary history of what the user has worn.	Desirable

3.1.4.9 The database shall generate favorites items more frequently.	Desirable
3.1.4.10 The database shall respond to query requests from the program.	Essential

REEL CO

REEL COLOSET
Supplementary Specifications

Version <1.1>



Revision History

Date	Version	Description	Author
<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

1. Introduction 4
   - 1.1 Purpose
   - 1.2 Scope
   - 1.3 Definitions, Acronyms, and Abbreviations
   - 1.4 References
   5
   - 1.5 Overview

2. Assumptions and Dependencies 5

3. Usability 5
   - 3.1 User's Knowledge
   - 3.2 Graphical Interface
   - 3.3 Usability Standards

4. Performance 5

5. Supportability 5

6. Design Constraints 6

7. Security 6

8. Interfaces 6

Supplementary Specifications

1. Introduction

1.1 Purpose

Supplementary specifications are for describing the requirements that we can't define well in our Use Case Models. These could include requirements like: reliability, performance, design constraints, usability, security, and many other things.

1.2 Scope

The scope of the project includes all specifications and systems included in the REEL COLOSET model and its associated requirements.

1.3 Definitions, Acronyms, and Abbreviations

API
: Application Programming Interface; a software system that allows two applications to interact with each other.

CSS
: Cascading Style Sheets; programming language used to describe how HTML elements should be described.

HTML
: HyperText Markup Language; markup language that describes how elements should be displayed on a browser.

HTTP
: Hypertext Transfer Protocol; application layer of the internet suite.

JSON
: JavaScript Object Notation; a data-interchange format used for representing data based on the syntax of a JavaScript object.

Snake Case
: The naming convention where every word starts with a lowercase letter and each word is separated by an underscore, e.g. 'variable_name'. In this convention, each space in English writing is represented as an underscore.

SQL
: Structured Query Language; programming language used for managing and organizing databases.

TCP
: Transmission Control Protocol; a protocol to the internet suite.

RC
: REEL COLOSET

1.4 References

UPEDU : http://www.upedu.org/

Course Website : https://people.eecs.ku.edu/~saiedian/Teaching/448/

1.5 Overview

This supplementary specifications document lists and describes all of the non-functional requirements that are required for RC to operate efficiently.

2. Assumptions and Dependencies

This supplementary specifications document assumes that the user is familiar with the use of websites, is using a non-deprecated browser, and the data received from both the weather API and the user is accurate.

3. Usability

3.1 User's Knowledge

Targeted users are familiar and comfortable with using web browsers. The product should be intuitive and user-friendly and should require no training or help to navigate and use the site.

3.2 Graphical Interface

The consumer will only interact with the product through the graphical interface. All system functionalities must be accessible by keyboard and/or mouse. The user must be able to click tabs at the top to access any other portion of the website, e.g. accessing the closet tab from the daily outfit tab, as well as type their account information on the login screen or the profile tab.

3.3 Usability Standards

The interface should be intuitive and easy to learn with buttons, headings, and help/error messages that are simple to understand. The interface should also have a low perceived workload so that it does not look intimidating, demanding or frustrating.

4. Performance

The application shall run on as long as the consumer is able to run a TCP connection to the HTTP. The web application will execute itself on an internet browser and response time for all tasks shall be inferior to 3.0 seconds.

5. Supportability

The consumer can access the website from any non-deprecated browser. The snake case naming convention shall be utilized when creating all variables and file names in the project.

6. Design Constraints

The programming languages used will include JavaScript, HTML, and CSS for all web content and SQL for all database queries. When generating the daily outfit, the program shall parse a JSON response from the weather API.

7. Security

The user provides a username, password, and city location, and the system has a database which includes all of the items in their closet. All of these will need to be protected - hackers may attempt SQL injection to bypass authentication to obtain sensitive information. This private information can include the person's city or someone's password that the hacker could try to use elsewhere, which is a breach of privacy.

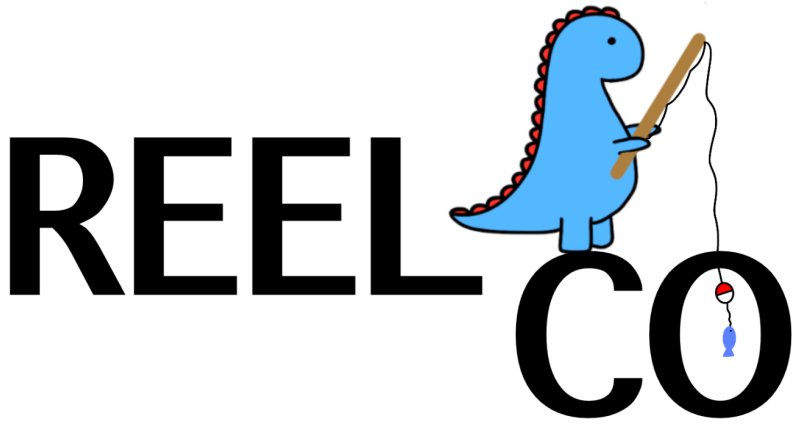
8. Interfaces

The User Interfaces for the RC website includes the graphical user interface, this will be what the consumer interacts with on the website. The Software Interfaces that are used by RC includes the weather API database interface and the Account Information database interface. The Software Interfaces are what the program will interact with to generate the daily outfit recommendations and keep the laundry hamper updated.

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

1. Use-Case Model 6
   - 1.1 Introduction
   - 1.2 General Actors Descriptions
   - 1.3 Use-Case Model Hierarchy
   - 1.4 Diagrams of the Use-Case Model 8

2. Create Account 9
   - 2.1 Brief Description
   - 2.2 Flow of Events
   - 2.3 Special Requirements
   - 2.4 Preconditions
   - 2.5 Postconditions
   - 2.6 Extension Points
   - 2.7 Relationships
   - 2.8 Use-Case Diagrams
   - 2.9 Other Diagrams

3. Daily Outfit 10
   - 3.1 Brief Description
   - 3.2 Flow of Events
   - 3.3 Special Requirements
   - 3.4 Preconditions
   - 3.5 Postconditions
   - 3.6 Extension Points
   - 3.7 Relationships
   - 3.8 Use-Case Diagrams
   - 3.9 Other Diagrams

4. Laundry 11
   - 4.1 Brief Description
   - 4.2 Flow of Events
   - 4.3 Special Requirements
   - 4.4 Preconditions
   - 4.5 Postconditions
   - 4.6 Extension Points
   - 4.7 Relationships
   - 4.8 Use-Case Diagrams
   - 4.9 Other Diagrams

5. Edit Profile 12
   - 5.1 Brief Description
   - 5.2 Flow of Events
   - 5.3 Special Requirements
   - 5.4 Preconditions
   - 5.5 Postconditions
   - 5.6 Extension Points

5.7 Relationships
5.8 Use-Case Diagrams
5.9 Other Diagrams

6. Closet 13

6.1 Brief Description
6.2 Flow of Events
6.3 Special Requirements
6.4 Preconditions
6.5 Postconditions
6.6 Extension Points
6.7 Relationships
6.8 Use-Case Diagrams
6.9 Other Diagrams

7. Login/Logout 14

7.1 Brief Description
7.2 Flow of Events
7.3 Special Requirements
7.4 Preconditions
7.5 Postconditions
7.6 Extension Points
7.7 Relationships
7.8 Use-Case Diagrams
7.9 Other Diagrams

8. Login 15

8.1 Brief Description
8.2 Flow of Events
8.3 Special Requirements
16
8.4 Preconditions
8.5 Postconditions
8.6 Extension Points
8.7 Relationships
8.8 Use-Case Diagrams
8.9 Other Diagrams

9. Logout 17

9.1 Brief Description
9.2 Flow of Events
9.3 Special Requirements
9.4 Preconditions
9.5 Postconditions
9.6 Extension Points
9.7 Relationships
9.8 Use-Case Diagrams
9.9 Other Diagrams

Figures

Figure 1 Website Connections/Use Packages……..7
Figure 2 REEL COLOSET System……8
Figure 3 REEL COLOSET Login/Logout System……15

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

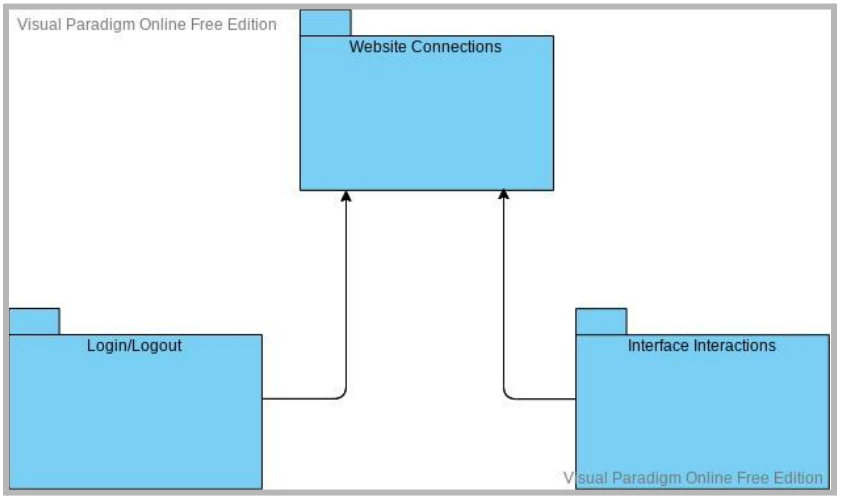


Figure 1 Website Connections/Use Packages

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

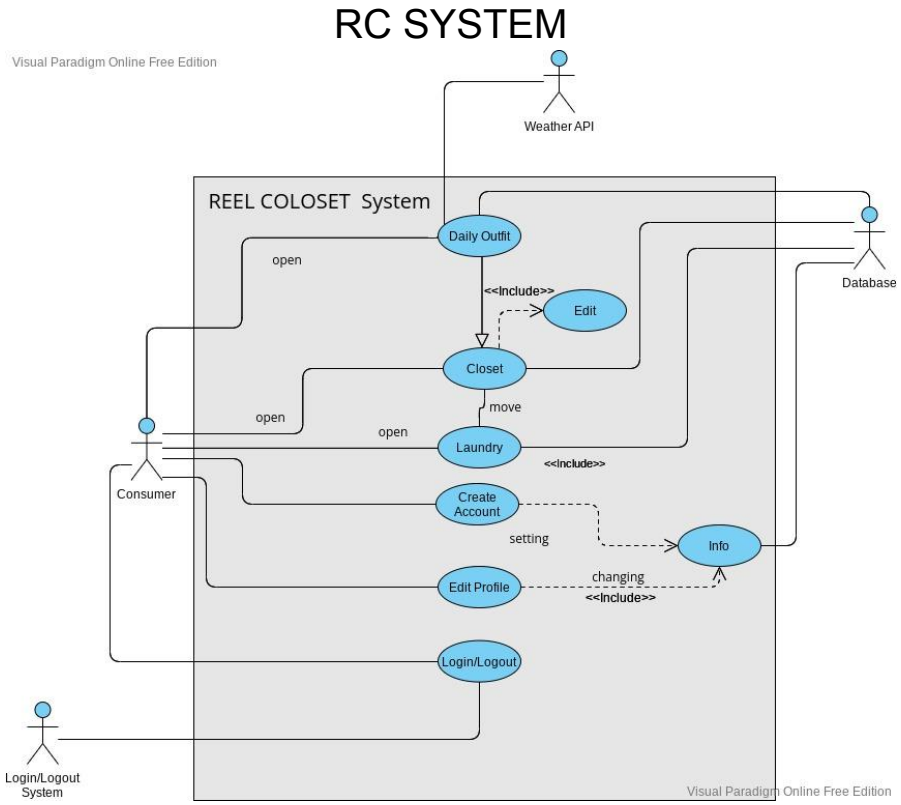


Figure 2 RC System

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

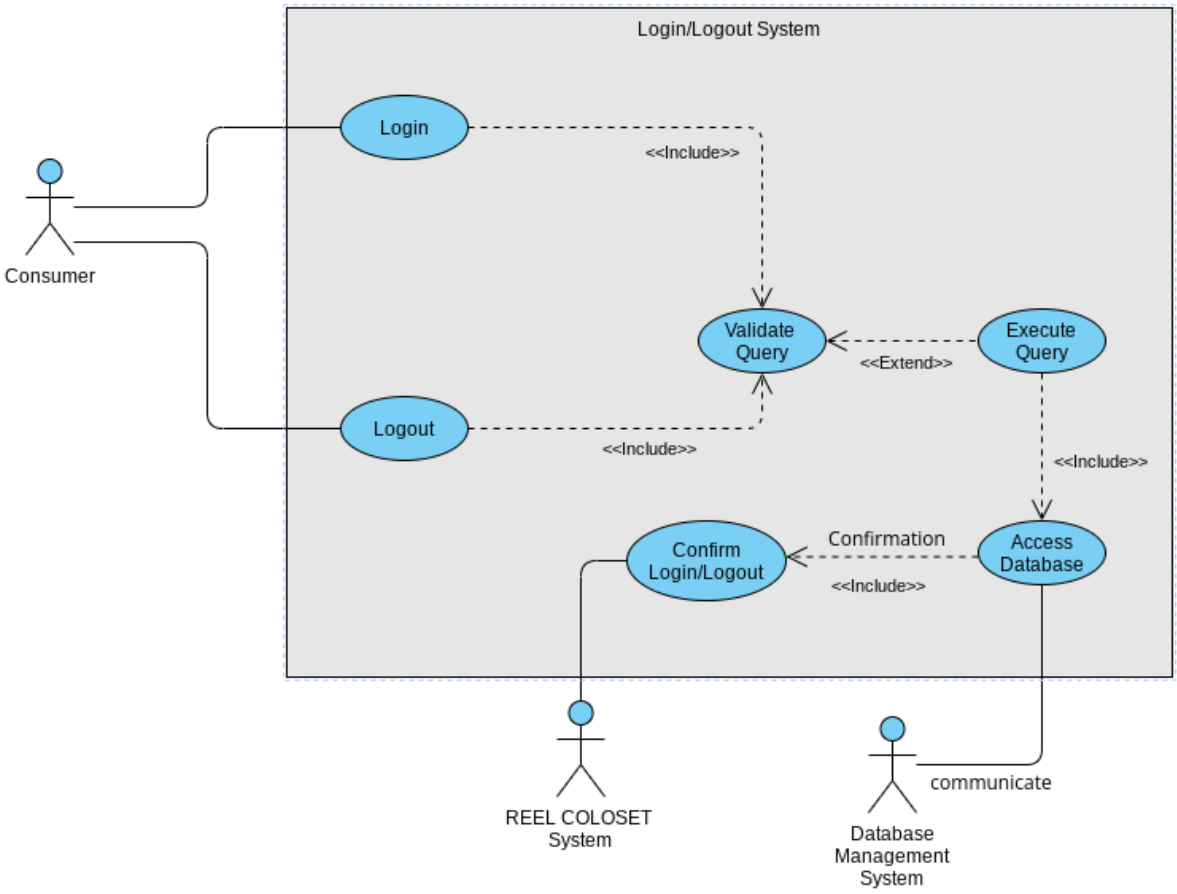


Figure 3 RC Login/Logout System

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.