Tecnológico
de Monterrey

Web Application Development

Project Proposal

ReWear

Lecturer: Msc Aldredo Salazar Velez

Heiko-Gabriel Scholz : A01677275

Heiko-Gabriel Scholz

1.a) Purpose oft he Application

Clothing is a matter for everyone. Everyone has a lot of clothes and consumes them over their lifetime. But many of the clothes we buy we do not like that much as we thought when we bought them. Or we have clothes which are in good condition but do not fit us any longer.

The consumer behaviour of clothes in the western society leads to many problems in the world. On the one hand, clothes require a lot of resources for their production like water and poisonous chemicals. On the other hand, these clothes are produced in sweatshop und unhuman conditions.

ReWear is a business where people can swap their old clothes with other people. So clothes will have a longer lifetime and the consumption of clothes decreases. For this business to work, our company requires a web application, where people can order the clothes and share their own clothes.

1.b) Scope

The scope of the project are young people between 16 and 30 years, who are aware of the dramatic damage the consumer behaviour of clothing is causing to people and the environment. The website should give every user an easy opportunity to make use of their old clothes and swap them for other used clothes. With this website people can save money, get rid of their old clothes and do good for the planets future.

1.c) Vision

The vision of this project is to create an application which invites every young person in Sweden to be part of our community of our new trend to make better use of your used clothes and to make the society more aware of the damage they cause by their consumer behaviour.

1.d)

The objective of this application is to give the community of our users the required functionality to manage all processes included for the realization of our business in a friendly and easy way.

1.e) Define the primary content of your web application

The following list gives an overview of all the functionalities our web application will include to realize ours companies scope.

Login Page

A page where already registered users can log in to our website. It is required to be a registered user to have access to our service. The user requires a username and a password to get access to our service.

Registration Page

For new users we need an opportunity to create an account for our service. The user needs to enter several personal information to create an account. If every required information is provided, the account can be created and will be saved in the database.

Profile Page

The user has a profile page where he can see his personal data and can make changes to it.

This page is also to specify the personal styling of each customer. Our service is to deliver packages of either 5 or 10 clothing products which are fitting to the users styling preferences. In this page the user selects his clothing style and his preferences for his next order .

Inventory

The idea of our business is, that before users purchase packages of clothing, they first send in some of their used clothes. In the Inventory section are all the clothes send in by a user displayed.

Order

On this page the user can chose the product he wants to purchase. He can chose between small and large packages of clothing.

1.f) Define the secondary content of your web application

About us

The idea behind ReWear is to create a community of people who share the value of sustainability and environmental friendly behaviour. On this section we provide information about our company and how it helps to save the environment.

2.a) Functionality and usability

High Priority features

- Login
- Logout
- Registration
- Select clothing preferences
- Order Package
- View previous orders
- View inventory and sent clothes
- Change profile Data

Low Priority features

- Show saved resources by ReWear
- Delete account
- Feedback section for received packages
- Display received clothes

2.b) Style and Layout

Login Page

The Login Page will consist of a big picture showing clothes and the logo of ReWear in the centre of the Page. In the top right corner there is an input field where the users can log in to our Service. In case the users want to register, there is another button which changes the login field with the register field.

Home Page

The home has our company's logo in the top right corner. Beneath that on the top of the page is a navigation bar, where the user can switch between the different services.

2.c) Functionality Requirements

Use Case Scenario	Login
Functionality	The user logs in into the system
Precondition	The user has an existing account
Basic flow	1) User enters Username 2) User enters Password

	3) User clicks login button
Alternative flow	User has a running session
Postcondition	User gets directed to the home page
Exception flow	User cannot log in

Use Case Scenario	Registration
Functionality	User creates an account
Precondition	User does not have an existing account
Basic flow	1) User enters personal Data 2) User choses password 3) User confirms password 4) User clicks create account button
Alternative flow	User leaves webpage during the registration
Postcondition	Account is created and user accesses the homepage
Exception flow	Wrong password confirmation, username already exists

Use Case Scenario	Logout
Functionality	User logs out with his account
Precondition	User is logged in with his account
Basic flow	1.) User presses the logout button
Alternative flow	User leaves page without logging out
Postcondition	User’s session ends User gets redirected to the login page
Exception flow	

Use Case Scenario	Select styles
Functionality	User select his styles for his next order
Precondition	User is logged in with his account
Basic flow	1.) User goes to the Styles Section 2.) User fills out the input elements 3.) User presses save styles button
Alternative flow	User leaves section without saving
Postcondition	Users style profile data get updated
Exception flow	User enters invalid data

Use Case Scenario	Change user Data
Functionality	User changes his profile data
Precondition	User is logged into his account
Basic flow	1.) User goes to the profile section 2.) User presses change data button 3.) User changes the value of the input elements 4.) User presses save changes button
Alternative flow	User leaves section without pressing the save changes button
Postcondition	User profile data is changed in the database
Exception flow	User Enters invalid data to the input fields

Use Case Scenario	Select package
Functionality	User selects package type for his next order
Precondition	User is logged in with his account User has selected his styles
Basic flow	1.) User goes to product section 2.) User selects a product option 3.) User presses Order button
Alternative flow	User leaves page without selecting a style
Postcondition	User executes an order User goes back to product section to select a different offer
Exception flow	User presses order button without selecting a product

Use Case Scenario	Execute Order
Functionality	User executes his order
Precondition	User is logged into his account User has selected a product option
Basic flow	1.) User presses order button 2.) User gets order Information displayed 3.) User presses order now button
Alternative flow	User changes his product offer
Postcondition	Users order is saved in the database
Exception flow	User leaves page without executing the order

Heiko-Gabriel Scholz

Use Case Scenario	View Inventory
Functionality	User views his inventory of clothes
Precondition	User is logged into his account
Basic flow	1.) User goes to Inventory section 2.) User views clothes 3.) User presses on clothes
Alternative flow	User has does not click on clothes
Postcondition	User views detailed information about the cloth
Exception flow	User has not yet bought or send clothes

Use Case Scenario	Edit clothing information
Functionality	Changing information about the clothes of the inventory
Precondition	User has logged into his account User has sent clothes
Basic flow	1.) User goes to styling section 2.) User presses on a clothes 3.) User presses on the change data button 4.) User changes value of input fields 5.) User presses save changes button
Alternative flow	User changes data without pressing save button
Postcondition	Clothing data gets changed in the database
Exception flow	User enters invalid data to change

Use Case Scenario	Rating received clothes
Functionality	User rates the clothes he received from his orders
Precondition	User has logged into his account User has received clothes
Basic flow	1.) User goes to inventory section 2.) User selects a clothing part 3.) User presses on the rate button 4.) User selects a rating 5.) User presses save rating button
Alternative flow	User selects rating without pressing save button
Postcondition	Rating gets saved in the database Rate button disappears on the clothes
Exception flow	User presses save rating button without selecting a rating

2.c) Non-Functional Requirements

1.) The Website needs to be easy usable and all elements should be placed in easy to find positions of the webpage
2.) The interface needs to look user friendly and kind to use
3.) The system needs to protect the user's data and thus be protected against SQL injection
4.) The system should be stable and allow many users to use the service at the same time
5.) The system needs to be fast and operations need to be designed efficiently
6.) The web application shall run on all popular browsers such as Chrome, Firefox and Safari
7.) The System is stable to wrong data input and will inform the user about invalid data

3.) Specifications

3.a) Functional Specification

- Login (username, password)
  Makes an Ajax call to the Database and checks if the typed username and password fit to a registered user in a database. If it exists the user gets access to the Homepage and his session starts.
- Logout()
  Ends the users session with an Ajax call and redirects the user to the login page
- Register(firstname, lastname, username, password, passwordconfirmation, email)

  Makes an Ajax call which creates a new user object in the database. Throws an exception if an object with the username already exists in the database. Displays an error if the password and passwordconfirmation parameter are different. Displays warnings if all required inputs are no filled out.

- SelectStyle(type, size, color, brands)

Makes an ajax call to the backend. Checks if a users style profile is already stored in the database. In case it is not there, it creates a new row in the styles table referencing the current user. If it already exists, executes and update statement to change the users changes in the database. The brands parameter is an array of selected brands by the user.

- CreateOrder(product)
  Makes an Ajax call to the database and creates an order for the customer. Displays the order Information to the user. The product argument is either the small or the big package option. The order object in the database stores meta data about the order like the date, status, price and orderId.
- GetClothesDetail(clothes)

  Makes an Ajax call and get all stored information about the selected clothes. Shows the clothes information to the user. The clothes parameter is the ID of the selected clothes.

- ChangeClothesDetail (clothes)

  Makes an Ajax call which updates the provided information. The user has selected a clothing to update before. The clothes parameter is the previous pressed clothes.

- ChangeProfile(firstname, lastname, username, email, address)

  Makes an Ajax call which updates the profile table of the current user in the Database. The provided Parameters are optional, only one parameter is at least required to use the function.

- RateClothes(clothes, rating)

  Makes an Ajax call to the database and fills the rating row of the clothing with a rating. The clothes parameter is the Id of the selected clothes and the rating the rating of the selected clothes.

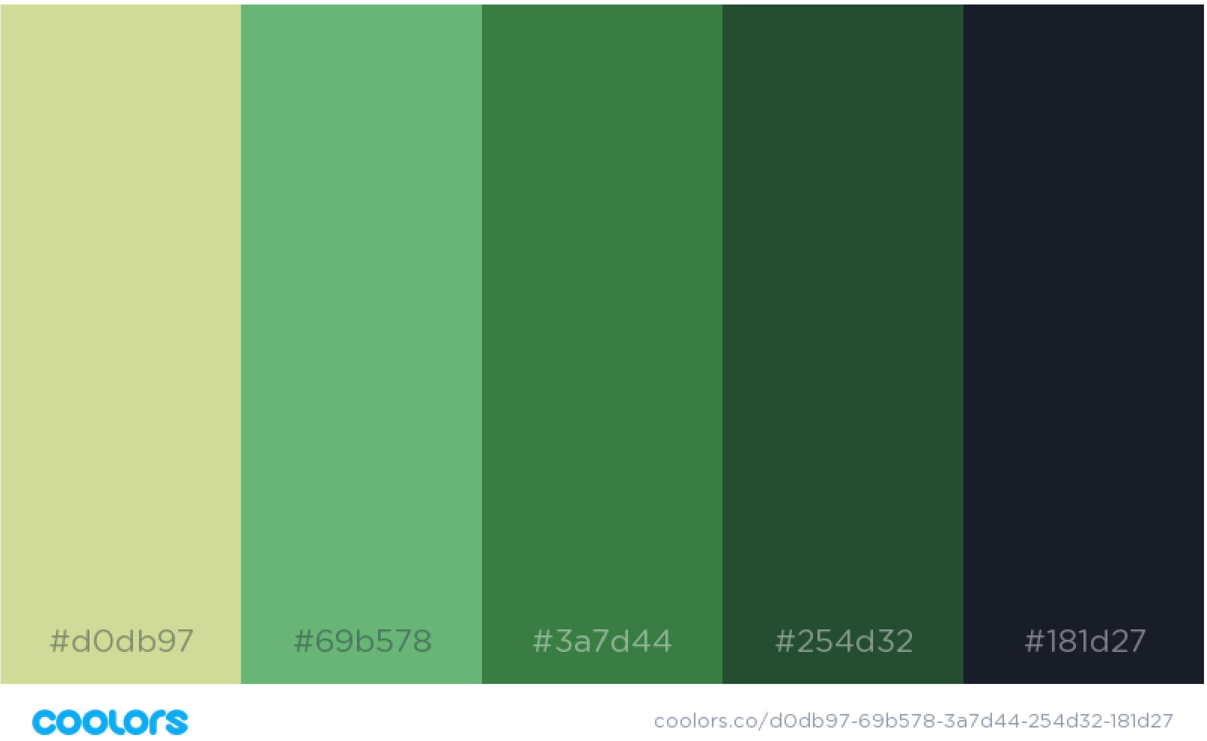
- GetOrders(username)

  Makes an Ajax call and gets all the order objects to a user. Displays all order information in a table to the user.

3.b) Design Specification

The design will use many predesigned styles and layouts of bootstrap. Buttons other inputs use predefined objects from bootstrap. In case the project isn’t finished until delivery date, simple css will be used to design the remaining parts.

The whole application shall be designed simple with less distracting styles. The page will include some pictures of clothes and use the following colour pallet:



3.c) Technical specification

For the realisation of this project, I will mainly rely on technieques dealt with in class.

Front end:

HTML5 for the basic structure of the webpage,

CSS for the basic styling I cant do with Bootstrap,

Jquery for the functions and the interactions with the backend via ajax calls and modification of HTML elements,

Javascript for the use of Jquery and every functionality that is not provided by JQuery,

Bootstrap for the Layout and styling of the whole webpage

Backend:

Ajax for the communication with the PHP services,

PHP for the communication with the Database,

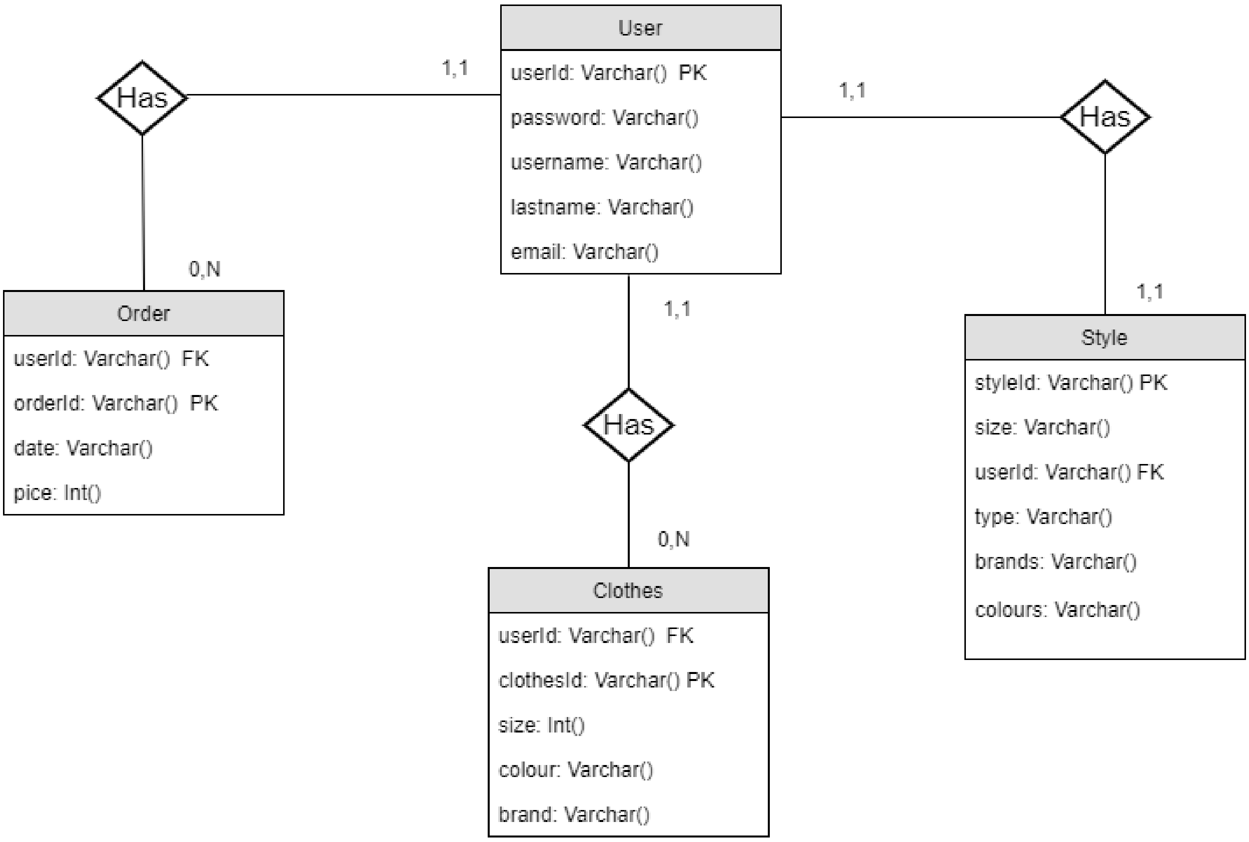
MAMP as a host for Application for testing and running,

MySQL as my Database storage system

4.) System Architecture

Like every modern application, I am also relying on the three tier architecture where the application is split into 3 logical parts.

1. The presentation layer, including all elements concerning the presentation and style of the webpage. Mainly including the HTML, CSS and Bootstrap elements.
2. The application layer for everything relating operations and providing functionality, like event handlers, functions and the ajax calls. This Layer mainly consists of Javascript, Qjuery, Ajax and PHP.
3. The data Layer where our MySQL database stores all sent information by the presentation layer. The entity relationship model for our database is displayed in the following diagram.



4.3) Architectural patter

For the implementation of this application I will orientate my architecture to the Model View Controller (MVC) pattern.

This patter divides the logical parts in 3 components.

1.) Model : The model describes the data objects handled by our application. In this application the model will be the customers of our service. It manages the data directly to the database and executes request from the controller layer.

2.) View : The view describes the presentation layer of our application. It manages how the request from the controller are displayed on the webpage.

3.) Controller: The controller handles all interaction from the user and sends request to the view and model to get the requested result for the user. In our case he might ask for detailed information about a clothing from the database and then send it to the view where it gets displayed to the user.

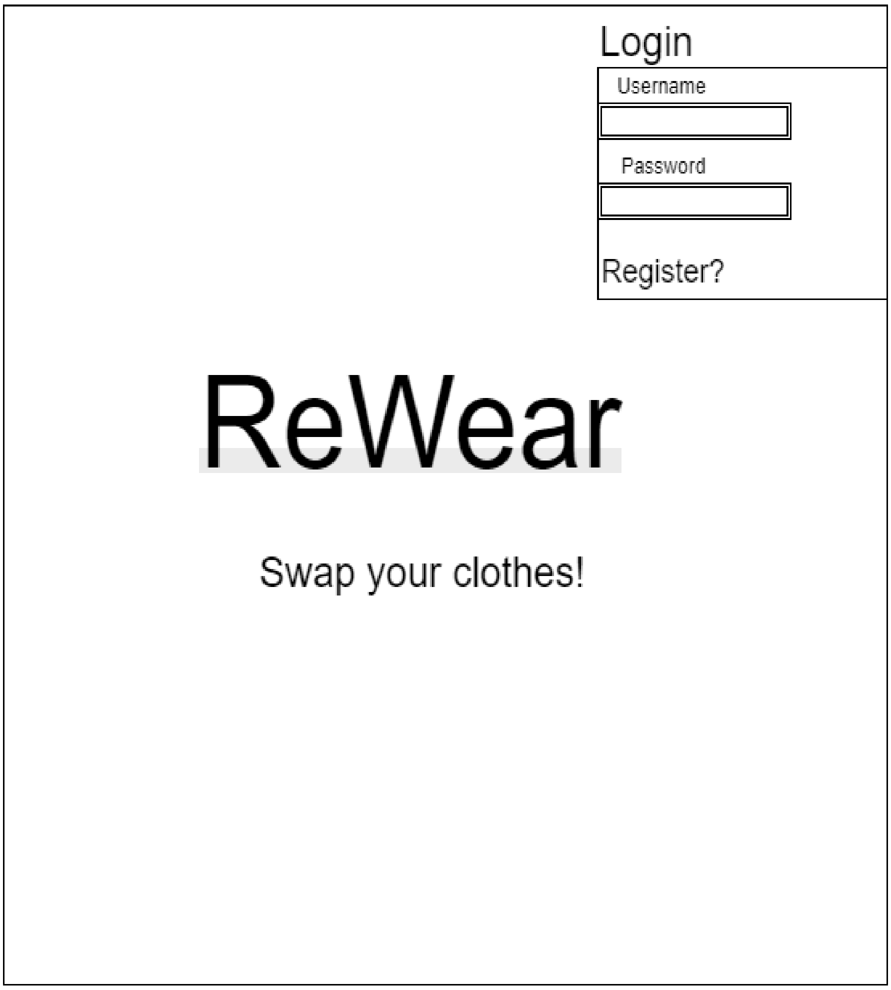
5.) Ressources

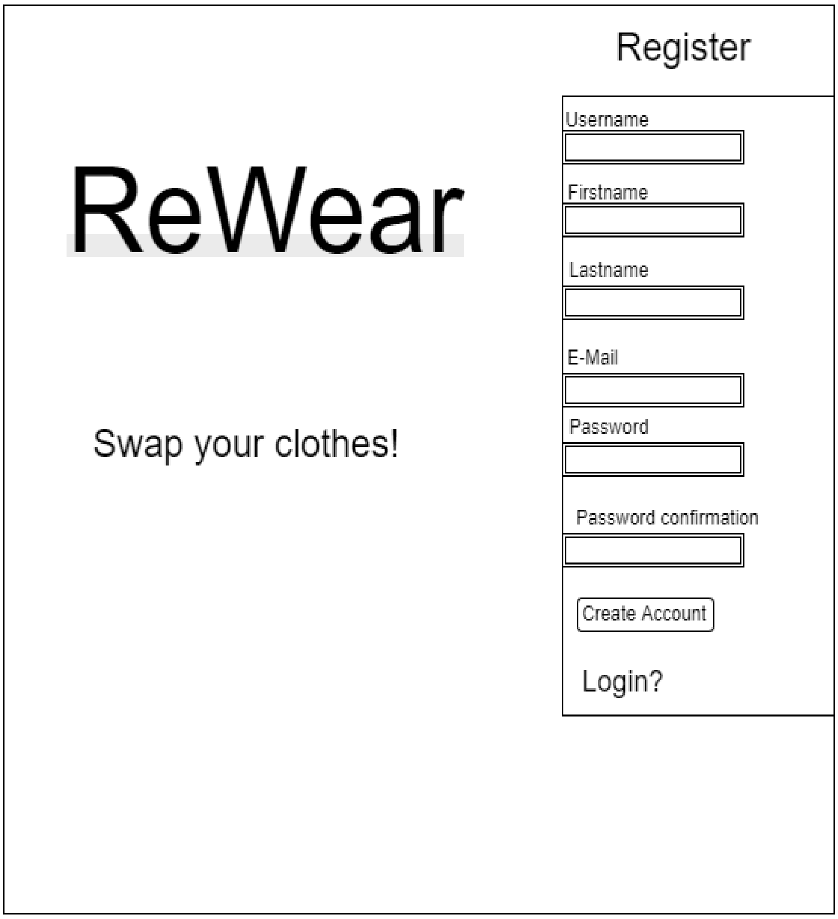
https://medium.com/towards-data-science/10-common-software-architectural-patterns-in-a-nutshell-a0b47a1e9013

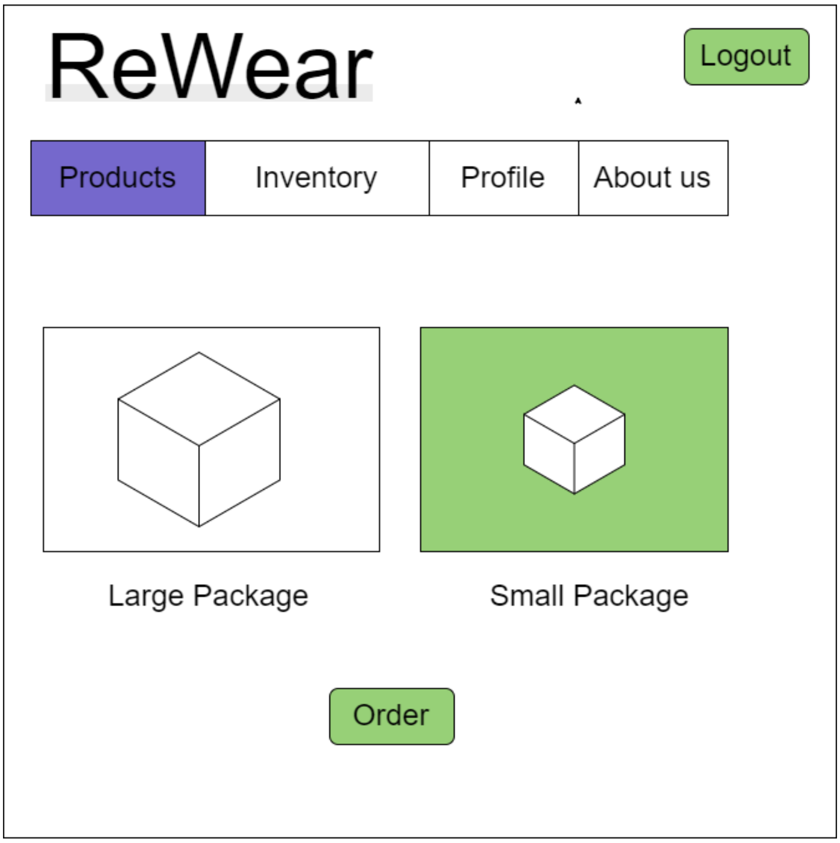
https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

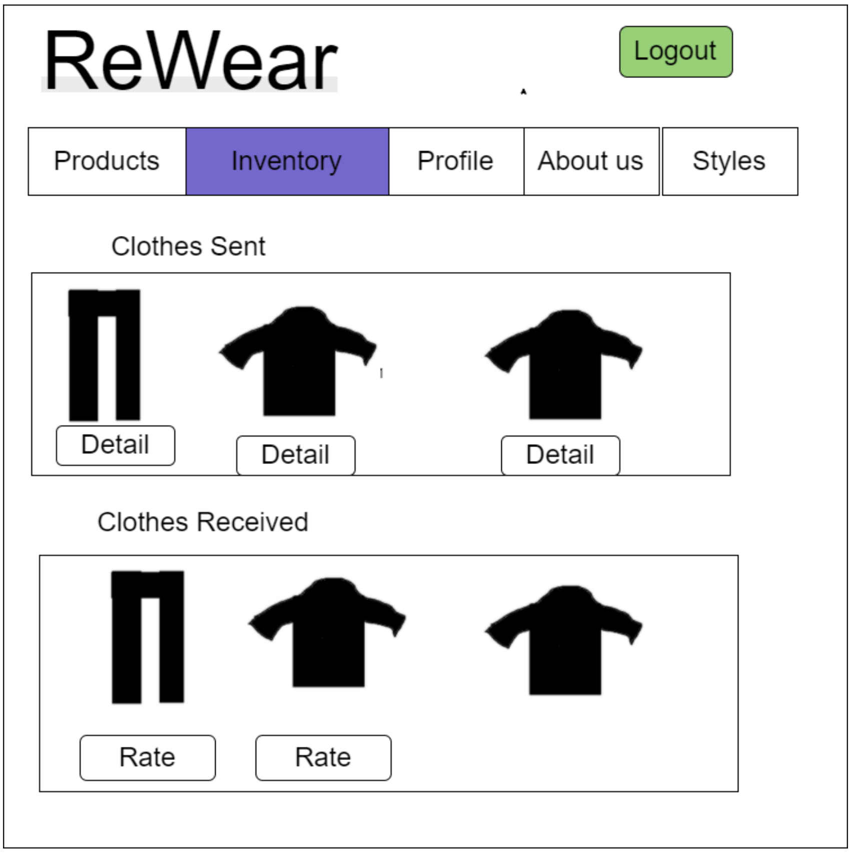
https://coolors.co/

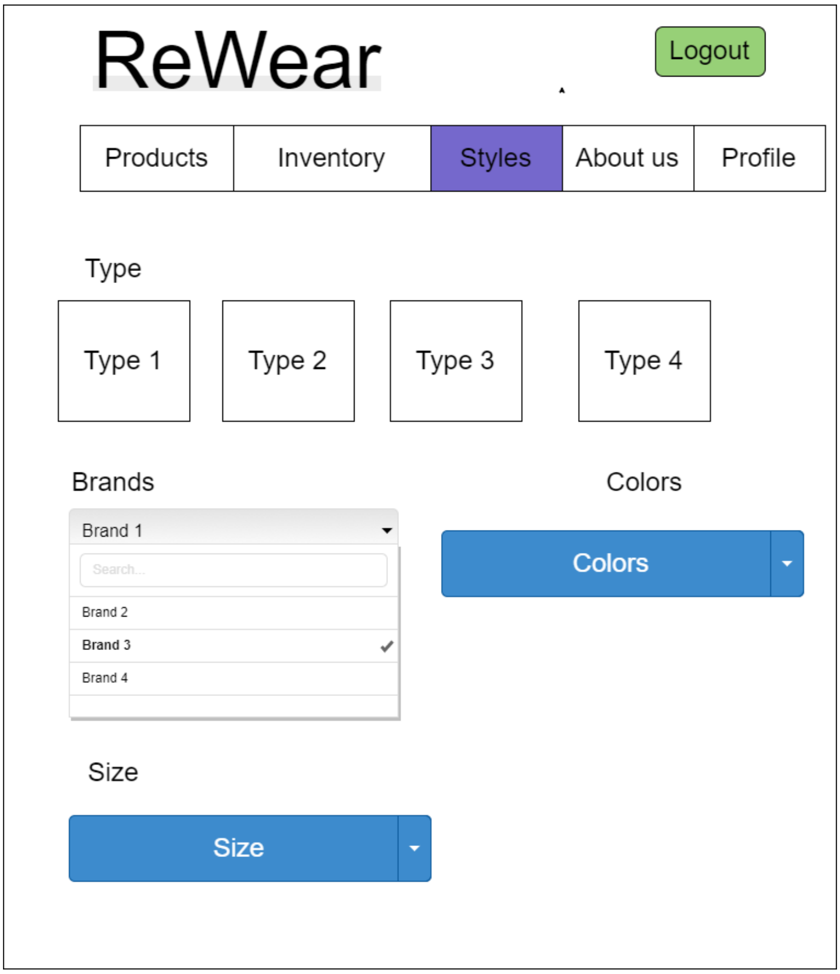
6.) Prototype




Register
ReWear
Swap your clothes!
Username
Firstname
Lastname
E-Mail
Password
Password confirmation
Create Account
Login?


ReWear
Logout
Products
Inventory
Profile
About us
Large Package
Small Package
Order


ReWear
Logout
Products
Inventory
Profile
About us
Styles
Clothes Sent
Detail
Detail
Detail
Clothes Received
Rate
Rate


ReWear
Logout
Products
Inventory
Styles
About us
Profile
Type
Type 1
Type 2
Type 3
Type 4
Brands
Colors
Brand 1
Search...
Brand 2
Brand 3
Brand 4
Colors
Size
Size