



# COMP9900

## Information Technology Project

### Final Report

### Scent Searcher

Team Fight (T18B)

2 August 2021

---

#### Group Members

Zhan Gao	z5287301	Scrum master, full stack developer	z5287301@ad.unsw.edu.au
Yuhao Ding	z5180648	Backend developer	z5180648@ad.unsw.edu.au
Xiaoyu Dong	z5323011	Front end developer	z5323011@ad.unsw.edu.au
Yinghan Li	z5178973	Data-processing developer	z5178973@ad.unsw.edu.au
Tianyi Wu	z5263505	Data-processing developer	z5263505@ad.unsw.edu.au

# Table of Content

<b>CHAPTER 1    OVERVIEW .....</b>	<b>1</b>
1.1.    BACKGROUND .....	1
1.2.    PROJECT DESIGN.....	2
1.3.    SYSTEM ARCHITECTURE.....	3
<b>CHAPTER 2    THIRD PARTY FUNCTIONALITIES.....</b>	<b>4</b>
2.1.    JAVA.....	4
2.1.1. <i>Spring Boot</i> .....	4
2.1.2. <i>Spring Boot Security</i> .....	4
2.1.3. <i>Project Lombok</i> .....	4
2.1.4. <i>Java JWT</i> .....	5
2.1.5. <i>MyBatis</i> .....	5
2.1.6. <i>MyBatis-Plus</i> .....	5
2.2.    REACTJS.....	6
2.3.    POSTGRESQL.....	7
2.4.    REACT FILE READER.....	7
2.5.    LOGIN AND REGISTER .....	8
2.6.    MODULE .....	8
<b>CHAPTER 3    DATA SOURCE.....</b>	<b>9</b>
<b>CHAPTER 4    FUNCTIONALITIES.....</b>	<b>10</b>
4.1.    DATABASE .....	10
4.2.    LOGIN/REGISTER PAGE.....	11
4.2.1. <i>Login</i> .....	11
4.2.2. <i>Create account</i> .....	12
4.2.3. <i>Sign out</i> .....	13
4.2.4. <i>Backend</i> .....	13
4.3.    HOMEPAGE .....	14
4.3.1. <i>Navigation bar (header/logo/footer)</i> .....	14
4.3.2. <i>Search bar (search by perfume name)</i> .....	14
4.3.3. <i>Search and select bars for notes</i> .....	14
4.3.4. <i>Search result display</i> .....	15
4.3.5. <i>Backend</i> .....	16
4.4.    USER CENTER .....	17
4.4.1. <i>My profile page</i> .....	17
4.4.2. <i>My bookmarked perfumes page</i> .....	20
4.4.3. <i>Backend</i> .....	22
4.5.    PERFUME DETAIL .....	23
4.5.1. <i>Perfume detail display</i> .....	23
4.5.2. <i>Reviews</i> .....	24

<i>4.5.3. Backend</i> .....	25
<b>4.6. USER PAGE .....</b>	<b>29</b>
<i>4.6.1. Link from perfume detail page</i> .....	29
<i>4.6.2. User page</i> .....	29
<i>4.6.3. Backend</i> .....	30
<b>4.7. ADMIN .....</b>	<b>31</b>
<i>4.7.1. Backend</i> .....	31
<b>CHAPTER 5 IMPLEMENTATION CHALLENGES .....</b>	<b>33</b>
<b>5.1. DATA PROCESSING .....</b>	<b>33</b>
<b>5.2. DATA CRAWLER.....</b>	<b>34</b>
<b>5.3. THE USE OF MYBATIS-PLUS .....</b>	<b>34</b>
<b>5.4. SEARCH PERFUMES BY MATCHING NOTES .....</b>	<b>35</b>
<b>5.5. INTERFACE DESIGNED ACCORDING TO HEURISTICS CRITERIA AND DESIGN PRINCIPLES</b>	<b>35</b>
<i>5.5.1. Match between system and real word</i> .....	35
<i>5.5.2. Error message</i> .....	36
<i>5.5.3. Help prevent error</i> .....	37
<i>5.5.4. Recognition rather than recall</i> .....	37
<i>5.5.5. The visibility of system status</i> .....	37
<i>5.5.6. Aesthetic and minimalistic design</i> .....	38
<i>5.5.7. User control and freedom</i> .....	39
<b>CHAPTER 6 INSTALLATION MANUAL .....</b>	<b>40</b>
<b>CHAPTER 7 CONCLUSION.....</b>	<b>41</b>
<b>CHAPTER 8 REFERENCE .....</b>	<b>42</b>

# Chapter 1 Overview

## 1.1. Background

The sense of smell is one of the important sense organs, some people say that the sense of smell is the first sense used at birth, and its importance lies in converting chemical signals into feelings. Perfume, as a scent business card, can show personal charm and impress others. Perfume has its own attributes and characteristics, so it becomes particularly important to find a perfume that suits you. With the popularity of the Internet and the development of related technologies, perfume search websites have come into the public's attention. The main goal of such websites is to serve those people who want to explore different perfumes and the perfume notes. According to China perfume industry research white paper (iresearch 2020), about 10 million people wear perfume every day in China. It seems a huge number, but what we need to know is that China's perfume market only accounts for 2.5% of the world market. From this, it is not difficult to find that the increasing perfume market is like a trend containing huge energy.

As websites which some perfume lovers know, such as Fragrantica and Noisetime, they can quickly understand the needs of scent searchers and show satisfactory perfumes for these users. Of course, the selection of perfume notes and more comprehensive customer service can also be realized on today's perfume search website.

We have browsed some websites which search perfume, including

- Fragrantica: An online perfume encyclopedia, perfume magazine, and a communication community for perfume lovers.  
<https://www.fragrantica.com/> (*Fragrantica.com 2021*)
- Nosetime: The world's largest Chinese perfume product library, perfume review community.  
<https://www.nosetime.com/> (*NoseTime.net 2021*)
- Basenotes: Online perfume resources in the UK, including perfume databases, message boards and forums.  
<https://www.basenotes.net/> (*Basenotes.net 2021*)

The page designs of these three websites are minimalist, and scent searchers can quickly locate the object they want to search for and the perfume information they want to know. For Nosetime, although this website can locate perfume simply by its name, note and brand, users cannot search for perfume through some more complex search criteria. Although Basenotes is very useful for perfume reviews, the reviews did not show the corresponding scores. When referring to the Fragrantica, although users use perfume notes to match the desired perfume results, they cannot specify how many matching notes are allowed to be lost in the results.

When searching for a perfume of interest, the user should be able to view all its detailed information. Most websites can only show simple information such as its name and brand. When the user wants to know the description of the corresponding perfume, full set of notes, and related pictures, relevant comments, average comment ratings, and the number of likes for the perfume, the user is often not given such a comprehensive information.

For user experience, users should be able to set up bookmarks for perfumes they are interested in and be able to view them at any time. In addition, scent researchers should be notified when new perfumes are added on the website.

## **1.2. Project Design**

Our project attempts to build a website to provide users with the ability to search for perfumes, explore different perfumes and the notes associated with them. The website will be equipped with a powerful database that will not only have details of the perfumes but will also store information on notes, users, reviews and more. Users will be able to make inquiries about what they are interested in during the process.

Scent is a being with unique abilities. It can sometimes be like a carrier for people to be able to store emotions and memories. Therefore, perfume is increasingly becoming a contemporary way of life. People want to explore different fragrances and want to understand more deeply the relationship between the various notes in a perfume. But there is no platform that integrates all the fragrances on the market that allows them to search for perfumes by the combination of notes easily and quickly. Sometimes the search results are not an exact match, and the combination of notes is different from what is expected, but it is not immediately intuitive to tell the difference.

The challenges of this project are the following. First, the processing and cleaning of the data will be the difficulties. And to increase the detailed information of the perfume, we also need to store the detailed description of the perfume as well as the pictures through web crawlers. Second, the project will use the latest technology architecture and technology stack for both front-end and back-end. The presentation layer will use HTML, CSS and reactJS. The business layer will use Spring, Spring MVC and Mybatis, and PostgreSQL will be the processing tool for the data layer.

### 1.3. System Architecture

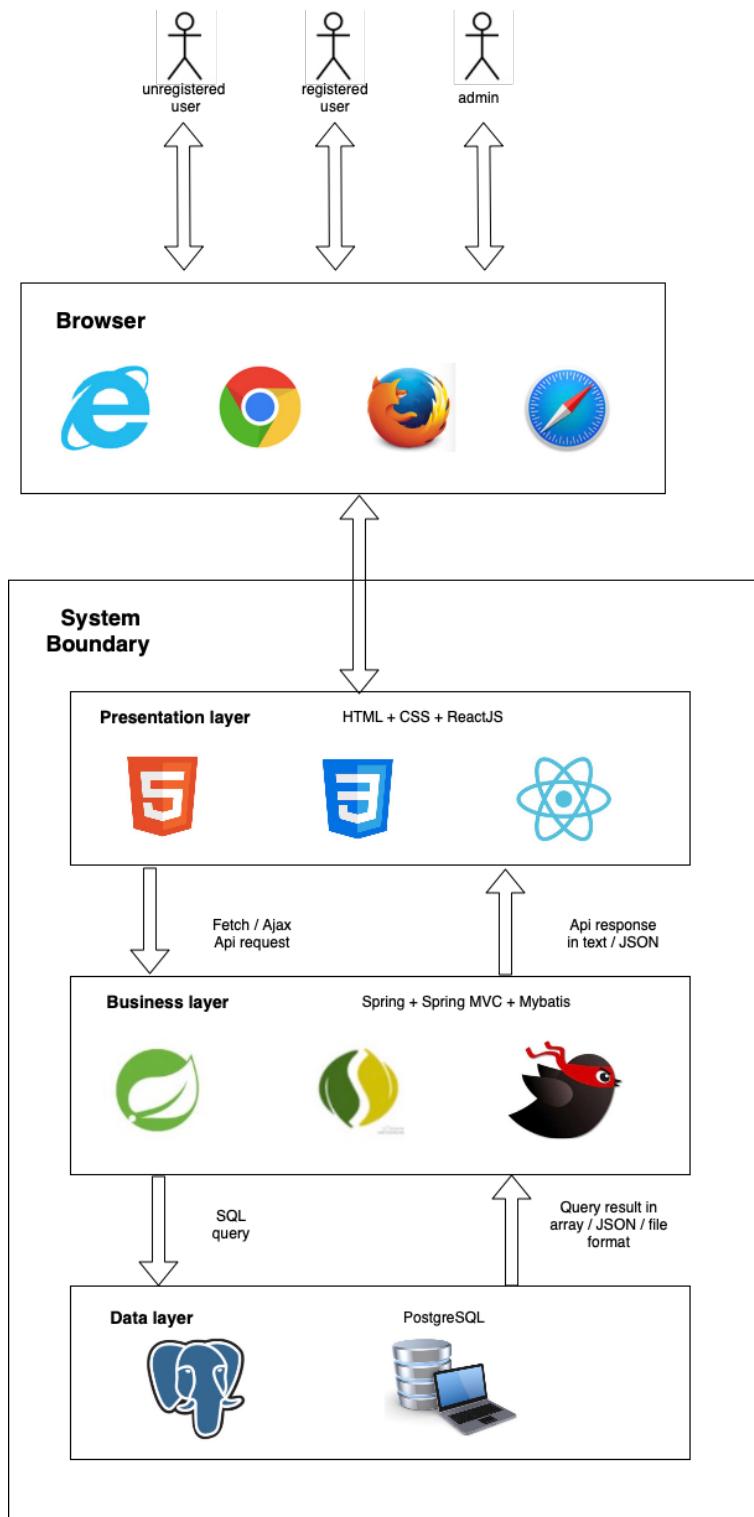


Figure 1 System architecture diagram

## **Chapter 2    Third Party Functionalities**

### **2.1. Java**

#### **2.1.1. Spring Boot**

Spring boot is a Java framework that can help create stand-alone, production-grade Spring-based applications, the primary goals are (Webb et al., 2021):

- Provide a radically faster and widely accessible getting-started experience for all Spring development.
- Be opinionated out of the box but get out of the way quickly as requirements start to diverge from the defaults.
- Provide a range of non-functional features that are common to large classes of projects (such as embedded servers, security, metrics, health checks, and externalized configuration).
- Absolutely no code generation and no requirement for XML configuration.

Spring boot permits redistribution and private use. There are no license issues using this library.

#### **2.1.2. Spring Boot Security**

Spring Security is a framework that provides authentication, authorization, and protection against common attacks. With first class support for both imperative and reactive applications, it is the de-facto standard for securing Spring-based applications (Alex et al., 2021).

Spring boot security permits redistribution and private use. There are no license issues using this library.

#### **2.1.3. Project Lombok**

Project Lombok is a java library that automatically plugs into the editor and build tools, spicing up the java project. It helps inject getter or equals method automatically with one annotation, so it can automate logging variables (Project Lombok, 2021).

Project Lombok permits redistribution and use in source and binary forms, with or without modification. There are no license issues using this library.

#### **2.1.4. Java JWT**

JJWT aims to be the easiest to use and understand library for creating and verifying JSON Web Tokens (JWTs) on the JVM and Android. JJWT is a pure Java implementation based exclusively on the JWT, JWS, JWE, JWK and JWA RFC specifications and open source under the terms of the Apache 2.0 License (jjwt-api 0.11.2 javadoc (io.jsonwebtoken), 2021).

Java JWT permits use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so. There are no license issues using this library.

#### **2.1.5. MyBatis**

MyBatis is an excellent persistence layer framework that supports custom SQL, stored procedures, and advanced mapping. MyBatis eliminates almost all JDBC code and the work of setting parameters and obtaining result sets. MyBatis can configure and map primitive types, interfaces and Java POJOs (Plain Old Java Objects) as records in the database through simple XML or annotations (introduction | MyBatis-Plus, 2021).

#### **2.1.6. MyBatis-Plus**

MyBatis-Plus is an enhancement tool for MyBatis (open a new window). On the basis of MyBatis, it only enhances and does not change. It is born to simplify development and improve efficiency (introduction | MyBatis-Plus, 2021).

It has the following characteristics:

- No intrusion: only enhance and do not change, the introduction of it will not affect the existing project, it is as smooth as silk.
- Low loss: basic CURD will be automatically injected at startup, performance is basically no loss, direct object-oriented operation.
- Powerful CRUD operation: built-in general Mapper, general service, only a small amount of configuration can realize most of the CRUD operations of a single table, and a more powerful condition builder to meet various usage requirements.
- Support Lambda from call: through Lambda expressions, you can easily write all kinds of query conditions, no need to worry about writing wrong fields.
- Supports automatic generation of primary keys: supports up to 4 primary key strategies (including a distributed unique ID generator-Sequence), which can be configured freely, which perfectly solves the primary key problem.
- Support Active Record mode: support Active Record form call, entity classes

only need to inherit model class to perform powerful CRUD operations.

- Support custom global general operations: support global general method injection (Write once, use anywhere).
- Built-in code generator: use code or Maven plug-in to quickly generate Mapper, Model, Service, Controller layer code, support template engine, and more custom configurations for you to use.
- Built-in paging plug-in: based on MyBatis physical paging, developers do not need to care about specific operations. After configuring the plug-in, writing paging is equivalent to ordinary List query.
- Paging plugin supports multiple databases: support MySQL, MariaDB, Oracle, DB2, H2, HSQL, SQLite, Postgre, SQLServer and other databases.
- Built-in performance analysis plug-in: sql statement and its execution time can be output. It is recommended to enable this function during development and testing to quickly detect slow queries.
- Built-in interception plug-in: provides intelligent analysis and blocking of delete and update operations of the entire table, and can also customize interception rules to prevent misoperations.

## 2.2. ReactJS

ReactJS is a JavaScript library for building user interfaces, it has three main features (React – A JavaScript library for building user interfaces, 2021):

- Declarative

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

Declarative views make your code more predictable and easier to debug.

- Component-Based

Build encapsulated components that manage their own state, then compose them to make complex UIs.

Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

- Learn Once, Write Anywhere

We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.

React can also render on the server using Node and power mobile apps using React Native.

ReactJS permits use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so. There are no license issues using this library.

## 2.3. PostgreSQL

PostgreSQL is an open-source client/server relational database management system (PostgreSQL 2021). In addition to this, it adds support for JSON data, full-text search capabilities and other extensions. Relational database MySQL is currently the most popular open-source database, but in comparison PostgreSQL also has obvious advantages (PostgreSQL - Wikipedia 2021).

- PostgreSQL is extremely stable, and the probability of users experiencing server-level database loss is extremely low.
- PostgreSQL is a multi-storage engine, while MySQL is a single-storage engine.
- MySQL only supports standard types of data, while PostgreSQL supports many advanced data types, such as array. PostgreSQL also supports Boolean types, JSON and other NoSQL features.
- PostgreSQL is multi-process and MySQL is multi-threaded.
- PostgreSQL has a software for administration, pgAdmin, which is free to use and easy to operate (pgAdmin - PostgreSQL Tools 2021).

## 2.4. React file reader

It is used in admin page to process the uploaded local image to be a base 64 image string (react-file-reader, 2021).

We set ‘base64’ is false and ‘multipleFiles’ is true, and filetypes is only for png, jpeg and jpg. Therefore, React File Reader accepts validated uploading multiple images and then return a JS Object including both the base64 files and the HTML5 FileList.

Its MIT license permits users redistribution and reuse code for any purpose, even in the case of the proprietary software. If we want to make distribution or make changes, we need to include the original copy of the MIT license in our project. We have a license file to show the copyright.

## 2.5. Login and register

In this part, we review the work of from Jason Watmore about implementing user registration and login functionality with React Hooks and Redux (React Hooks + Redux - User Registration and Login Tutorial & Example | Jason Watmore's Blog, 2021).

Its MIT license permits users redistribution and reuse his work if we include the original copy of the MIT license in our project.

## 2.6. Module

Module is a python file. Generally, ‘import’ is used to import the python file when writing a program. The purpose is to use the methods defined in the imported module to achieve code reuse. The modules used in the crawler program are as follows:

- Requests

Requests is a powerful module related to network requests, used to request information from the server, which is also the basis of the crawler program.

- Time

Time is a time-related module, used to hibernate the crawler to prevent the crawler from being too fast to cause the website to terminate the process.

- Tqdm

Tqdm is a module related to the progress bar, used to visually display the progress of the crawler.

- Json

Json is a module that exchanges format with data, used to standardize the format of the acquired data information.

- Random

Random is a module related to random numbers, used to obtain random numbers as a tool for calling api.

- MD5

MD5 is a module related to encryption algorithm, as an algorithm tool for calling api key.

## **Chapter 3 Data Source**

Our data is sourced from Kaggle, a competition platform for data modeling and data analysis. Companies and researchers can post data on the site, and data analysts, statisticians, and data mining experts can download data from the site and compete to come up with the best models of the data (Kaggle - Wikipedia 2021). Kaggle dataset provides high quality data (Find Open Datasets and Machine Learning Projects | Kaggle 2021).

The data for this project is mainly from the Fragrances - Notes and User Rating dataset (Fragrances - Notes and User Rating 2021). The data provides detailed information on more than 50,000 perfumes and notes, and is divided into top, middle and bottom notes according to the three main notes of perfumes. This dataset on perfumes is relatively complete and can meet the needs of our project website to build the database.

# Chapter 4 Functionalities

## 4.1. Database

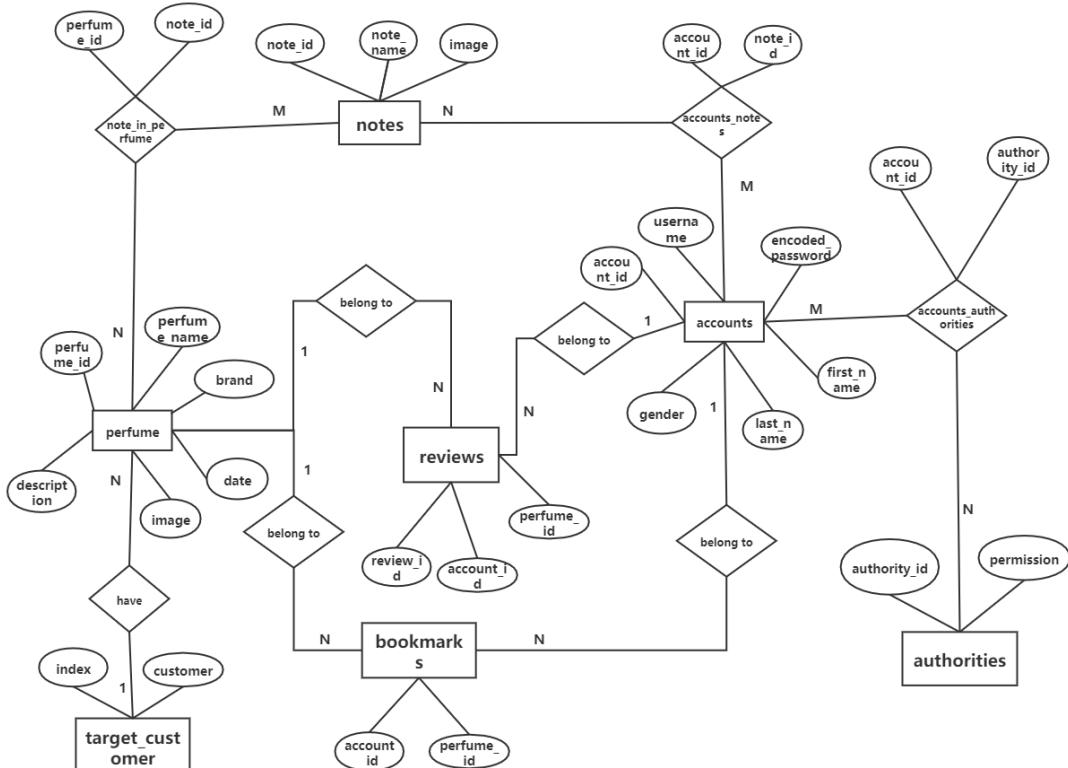


Figure 2 ER diagram

The ER diagram entity type, and its attributes are as follows:

- authorities (authority\_id, permission)
- accounts (account\_id, username, encoded\_password, first\_name, last\_name, gender)
- target\_customer(index, customer)
- perfume (perfume\_id, perfume\_name, brand, date, image, description, target)
- notes (note\_id, note\_name, image)
- bookmarks (account\_id, perfume\_id)
- reviews (review\_id, account\_id, perfume\_id)

The relationship is obtained as shown in the figure above:

Authorities and accounts are in a many-to-many relationship, that is, an authority can have multiple accounts, and an account can have multiple authorities. The relationship is: N:M

Attributes of the contact type:

accounts\_authorities(account\_id, authority\_id)

The relationship between target\_customer and perfume is one-to-many, that is, a target\_customer can have multiple perfumes. The relationship is 1:N

Notes and perfume have a many-to-many relationship, that is, a note can have multiple perfumes, and a perfume can have multiple notes. The connection is: N:M

Attributes of the contact type:

note\_in\_perfume (perfume\_id, note\_id)

Accounts and notes are in a many-to-many relationship. One account has multiple notes, and one note has multiple accounts. The connection is: N:M

Attributes of the contact type:

accounts\_notes (account\_id, note\_id)

Accounts and perfume have a pair-to-many relationship. One account has multiple perfumes, and one perfume has multiple accounts. The connection is: N: M

Attributes of the contact type:

bookmarks (account\_id, perfume\_id)

Accounts and perfume have a many-to-many relationship. One Account can comment on multiple perfumes, and one perfume can have multiple Accounts comments. Contact is N:m

Attributes of the contact type:

reviews (review\_id, account\_id, perfume\_id)

## 4.2. Login/register page

### 4.2.1. Login

The screenshot shows a login form titled "Welcome to Scent Searcher!". It includes a message for existing users, two input fields for a username and password, a "Login" button, and links for account creation and password recovery.

Welcome to Scent Searcher!

If you are an existing user, please login your account :)

Don't have an account yet?

[Create Your Own Account](#) in Scent Searcher!

Figure 3 Register form

Existing users can login to scent searcher in this page. In the bottom of the page, it provides link for new users to register an account. The input of password needs at least 6 characters to submit, and it will alert if users violate this rule, which is helpful for preventing error.

#### 4.2.2. Create account

### Create your own account in Scent Searcher!

First Name  
ikari

Last Name  
nagisa

Username  
 !  
Username is required

Gender  
Gender Diverse

password  
  
! Please match the format requested.  
Must contain at least 6 or more characters

Figure 4 Create account form

Each inputs has its own validation requirement, for example, password needs to be at least 6 characters. And a new user can only register new username which does not exist in the database. If input violate the requirement, under the input box, it will show alert message to ask user to alter to make a validate input.

The screenshot shows a registration form titled "Create your own account in Scent Searcher!". It includes fields for First Name, Last Name, Username, and a gender selection dropdown. The gender dropdown is open, displaying five options: "Choose your gender...", "Female", "Male", "Prefer Not To Disclose", and "Gender Diverse". Below the dropdown are "Register" and "Cancel" buttons.

Figure 5 Drop-down menu of selecting gender

This drop-down gender selection menu can help user to recognize the option they need.

If users found they have registered an account in scent searcher, they can click the ‘cancel’ button and then go back to the login page.

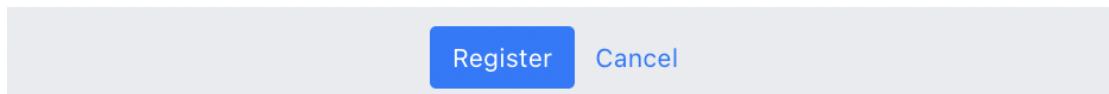


Figure 6 Register and cancel button in create account page

#### 4.2.3. Sign out

Button in header navigation can make user sign out.

#### 4.2.4. Backend

Login and register functions are implemented using Spring Boot Web and Security framework, in the form of Restful APIs. The APIs require the frontend to send authorization information (e.g., username and password) as JSON string, then the backend will communicate with the database using Spring Data JPA framework, checking if the authorization information is correct or create new account. After this, the backend will generate a JWT token for temporary authorization and send it back to the frontend for further access.

## 4.3. Homepage

### 4.3.1. Navigation bar (header/ logo/ footer)

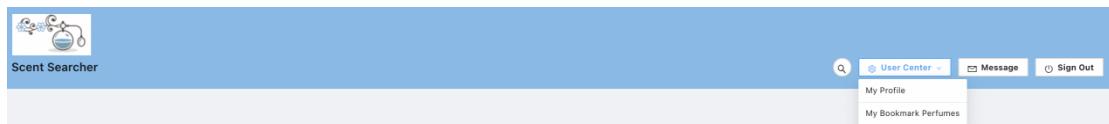


Figure 7 Header navigation bar



Figure 8 Footer navigation bar

When user click the logo of Scent Searcher and each navigation button in in footer and header, it will direct them to respective page.

These buttons in navigation bar provide more links to users, therefore, they can know where they are and where they want to go.

There are two subpages in user center, so a clean and well-structured drop-down menu is made to help to navigate users.

### 4.3.2. Search bar (search by perfume name)



Figure 9 Search bar (search by perfume name)

The first component of the home page is a search bar which can search perfumes by perfume name (Figure 9). After clicking the search button, the input string is sent to the backend to return a list of perfumes that contain the input string as part of the perfume name (ignoring cases).

### 4.3.3. Search and select bars for notes

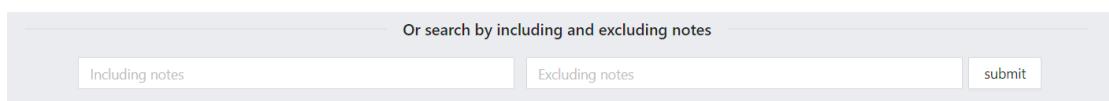


Figure 10 Search and select bars for notes

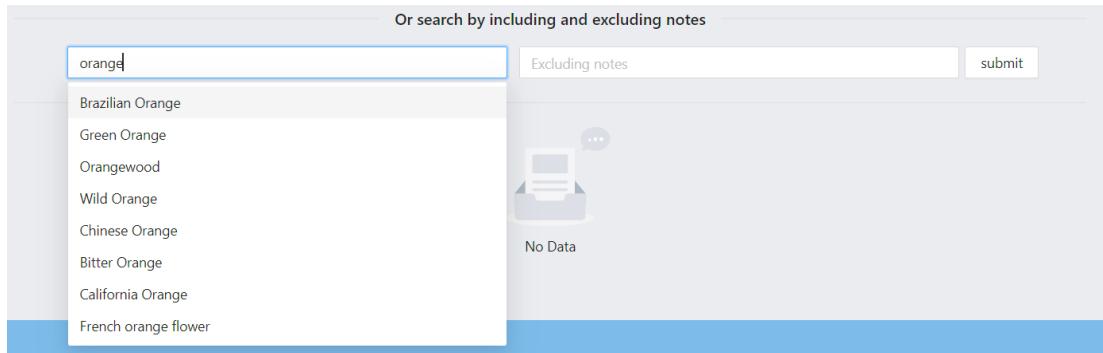


Figure 11 Search results for notes after entering a string

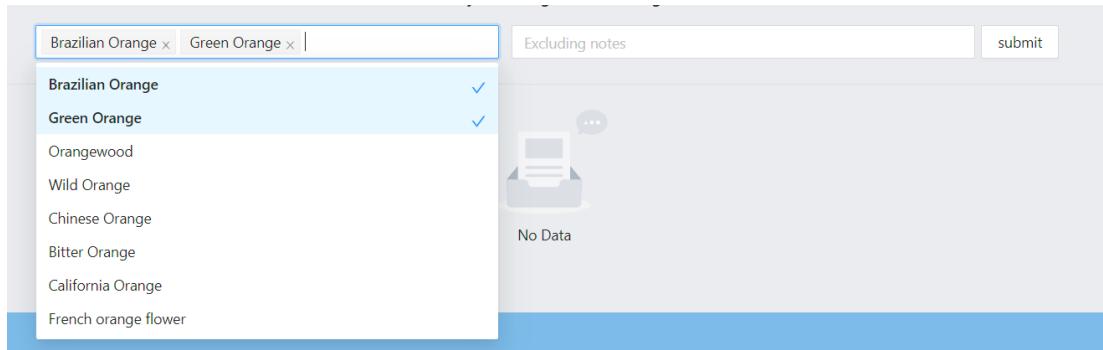


Figure 12 Select Search results for notes

The second component of the home page are two search and select bars for notes, one for notes that must be included in the result perfumes and the other one for the notes that must be excluded in the result perfumes (Figure 10). The two search and select bar both support async search functions, which means that after a short delay of entering a string, the notes that contains the input string as part of the note name (ignoring cases) will be displayed under the search bar, and the results can be added as selected notes after clicking (Figure 11 and Figure 12).

#### 4.3.4. Search result display

After submitting the searching criteria (either by perfume names or notes), the results will be displayed under the search bar, as a list of cards that contains the image, brand and perfume name of every result (Figure 13). There are 4 results in a row, and a maximum number of 20 results are displayed at one time. There is a more button at the bottom of the page, and after clicking the button, another maximum of 20 results (if there are more results) will be displayed. After clicking the card at any place, the browser will be redirected the details page of that perfume (as discussed below).

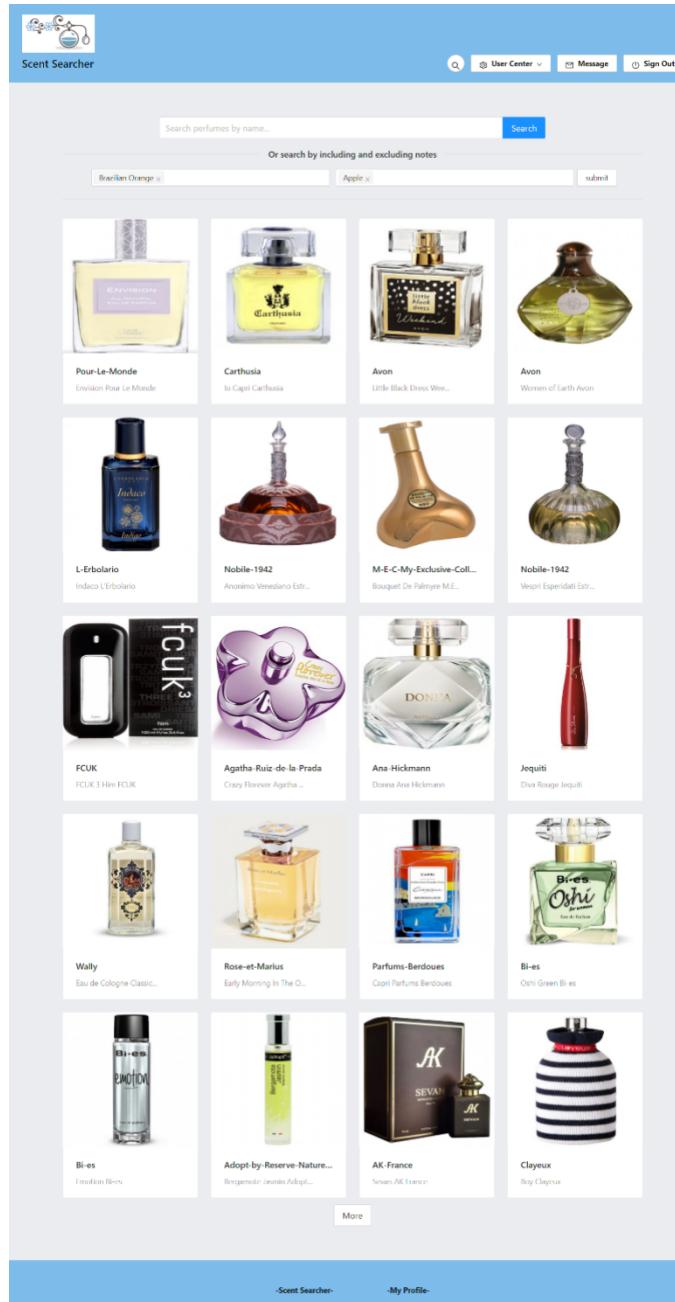


Figure 13 Search result display

#### 4.3.5. Backend

The search by perfume name function is implemented using Spring Boot Web framework, in the form of Restful API. The API takes a URL parameter as input string, then the backend will communicate with the database using Spring Data JPA framework, return all perfumes that contains the input string as part of its perfume name (ignoring cases), then return the results to the frontend in JSON format.

The search by notes function is implemented using Spring Boot Web framework, in the form of Restful API. The API takes a request body in JSON format which contains a list of note names that must be included, along with a list note names that must be excluded, as an input. Then the backend will communicate with the database using Spring Data JPA framework, return all perfumes that match the searching criteria, then return the results to the frontend in JSON format.

## 4.4. User center

In the drop-down menu of user center, it navigates users to two submenu, one is My profile page, and another one is My bookmarked perfumes page.

### 4.4.1. My profile page

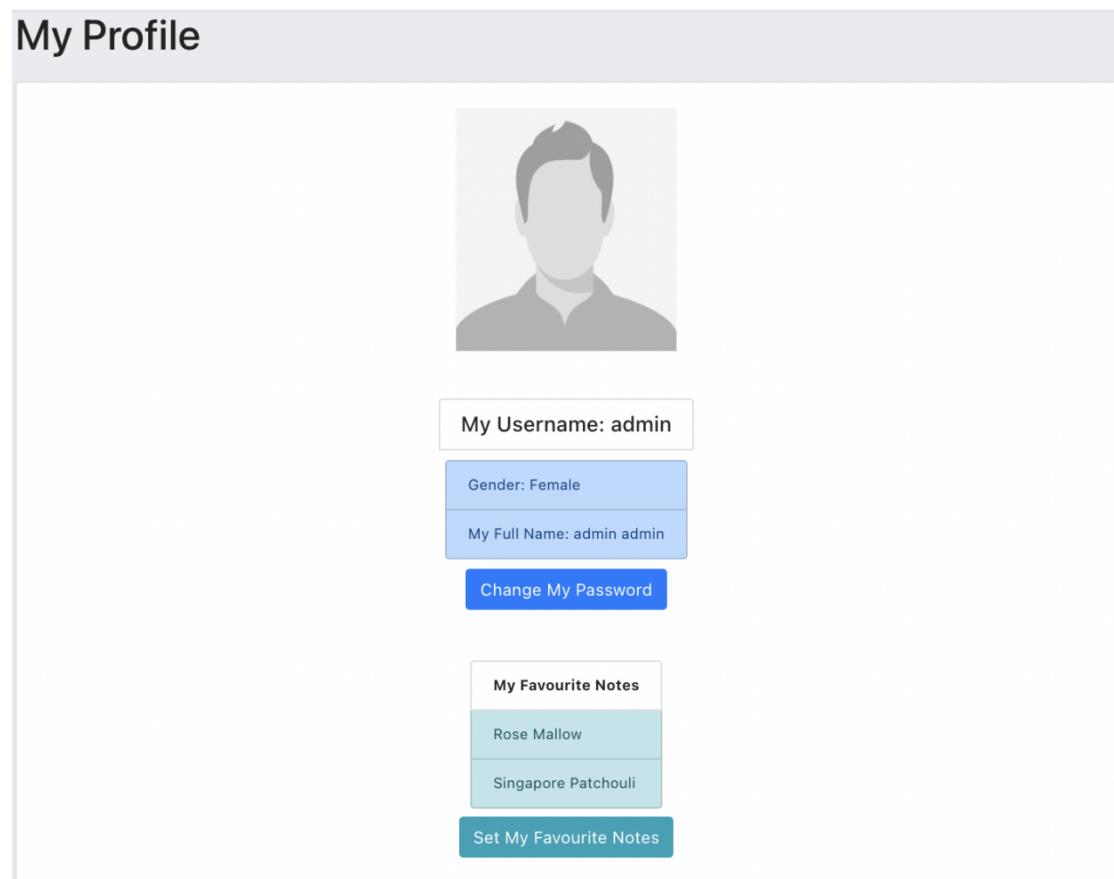


Figure 14 My profile page

In personal profile page, after calling api, it shows the user' details (username, gender, full name, favourite notes).

Users can change password and set favourite notes in profile page. There will be 2 modals of these two sub functions.

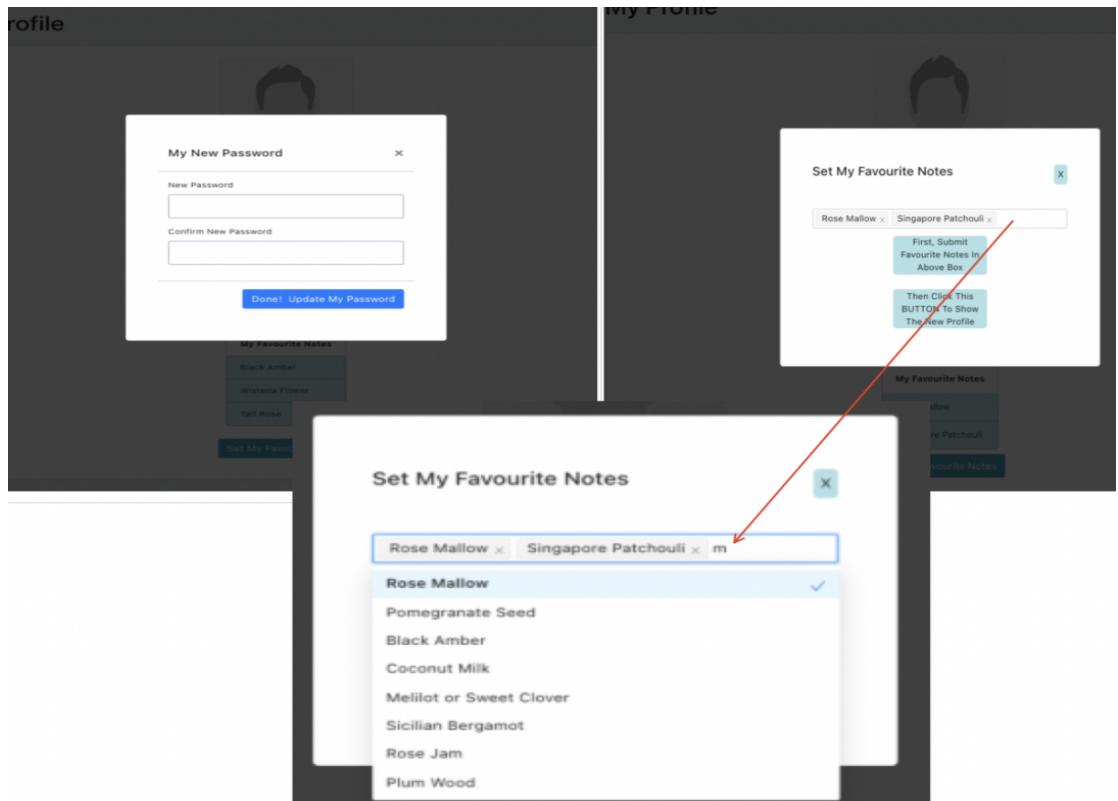


Figure 15 My profile and its function

#### 4.4.1.1. Change my new password

After several validation for passwords, users can update their passwords.

New Password
909090
Confirm New Password
1
<b>Confirm password does not match password</b>

New Password
9090
<b>Password must be at least 6 characters</b>

Figure 16 Change password form

#### 4.4.1.2. Set favorite notes

As shown in Figure 10, the search note bar is reusable from homepage. But it contains users' current favorite notes first, users can delete any notes they don't like anymore. once users input letter for searching notes, it will return notes contain that word or letter in a dropdown menu for displaying choices. Then users can select any notes they like and set the set of notes as their favourite notes. And then the 'My favorite notes' in their profile page will be updated.

#### 4.4.2. My bookmarked perfumes page

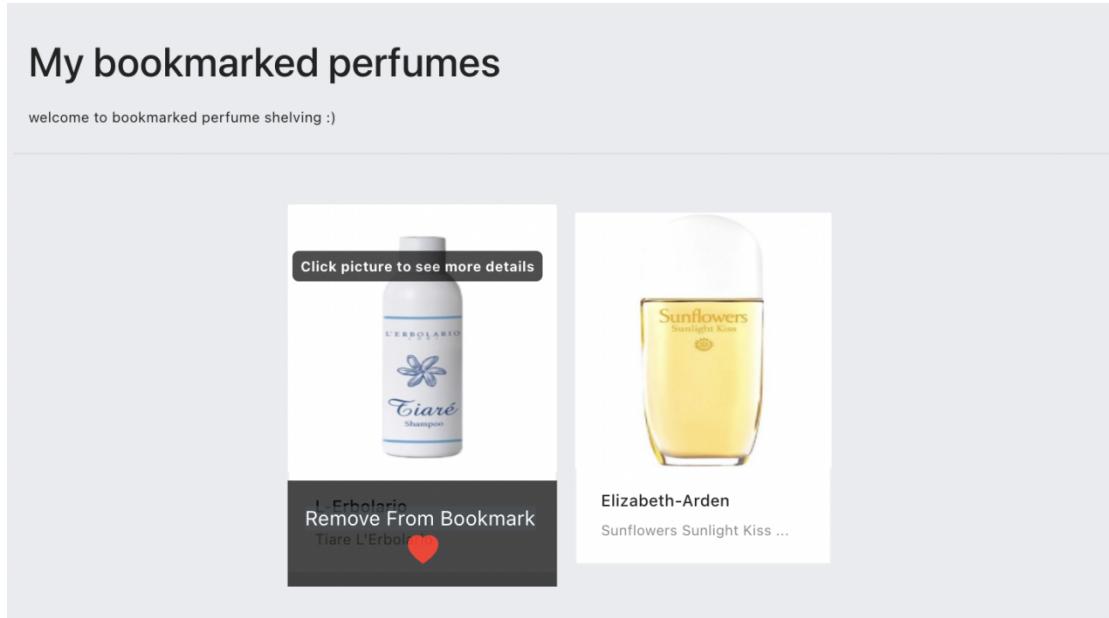


Figure 17 My bookmarked perfumes page

This page shows all perfumes user bookmarked, and when mouse hover on the perfume card, that card will pop out a little bit, which increase the visibility.

##### 4.4.2.1. A tooltip message

A tooltip message of 'click picture to see more' will appear when cursor hover on perfumes, and after user click on the perfume picture, it will lead users to perfume details page.

##### 4.4.2.2. Bookmark function

Users can click the 'remove from bookmark' component to remove that perfume from bookmark, and then the bookmark page will be updated immediately. The heart colour is red if that perfume is in user's bookmark, while the text is 'add to bookmark' and heart colour is grey if it is not.

There will be an alert message feedback once user update his bookmark successfully. The bookmark function is also reused in each perfume detail page.



Figure 18 Bookmark text and heart color changed according to the states

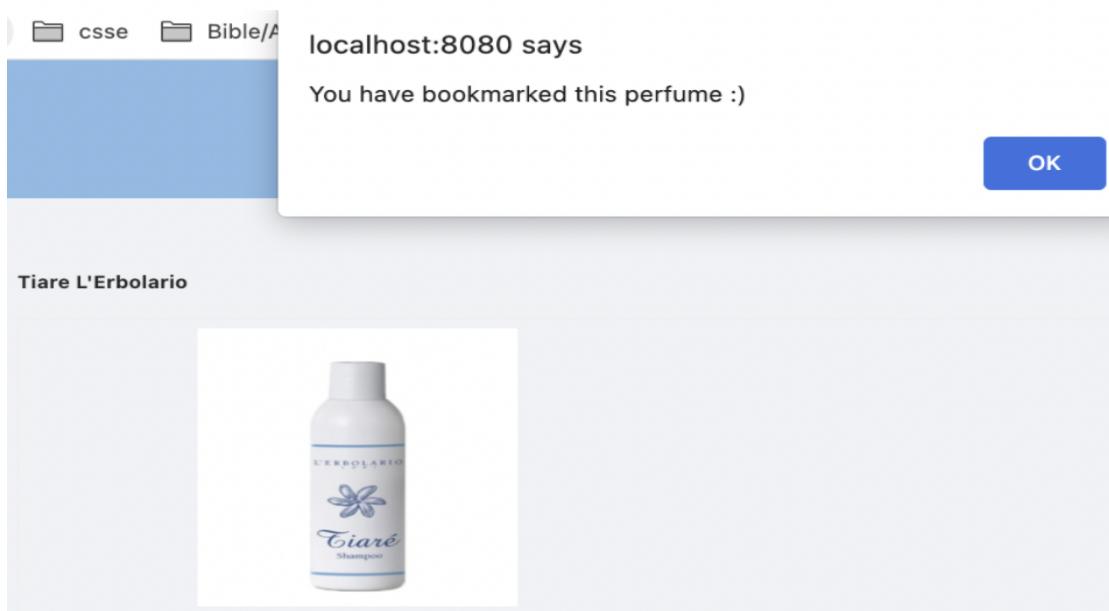


Figure 19 Success message alert of bookmark function

Once removed from the bookmark, the perfume will be removed from users my bookmarked perfumes page immediately.

#### **4.4.3. Backend**

For the operation of the profile page, it is mainly divided into viewing and modifying the profile and operating the perfume bookmarks.

First, when entering the page, the front-end will send the user's id to the backend, and the backend will query specific information belonging to the user based on the id, including username, gender, full name, etc. Then the back end will send this information packaged and sent to the front end in json format.

Users can also modify their own passwords. The front-end will package the user's id and the password that the user wants to modify and send to the back end. The back end will find the record belonging to the user according to the id and update the original password with the new password to complete the update. operate

## 4.5. Perfume detail

### 4.5.1. Perfume detail display

The perfume detail page can be accessed by clicking the search result card on the home page, after clicking, the browser is redirected to the detail page, and the ID of the perfume is added to the URL, which can be accessed later by the React framework (Figure 20). The page shows all the information of a perfume, including the perfume name, the image of the perfume, the brand, the production year, the target user, the description and the including notes (Figure 21).

localhost:8080/perfumes/2051

Figure 20 Example URL of perfume detail page

The screenshot shows a web application interface for a perfume detail page. At the top, there is a header bar with the text "localhost:8080/perfumes/2051". Below the header, the main content area displays the following information:

- Perfume Name:** Acqua di Gio Blue Edition Pour Homme Giorgio Armani
- Image:** A blue bottle of the perfume.
- Add To Bookmark:** A button with a heart icon.
- Table (Brand Information):**

Brand	Giorgio-Armani
Production Year	1996
Target User	for men
- Description:** A detailed paragraph about the perfume, mentioning its launch in 2014, its Mediterranean theme, and its key ingredients.
- Notes:** A list of fragrance notes including Amber, Mandarin Orange, Persimmon, Patchouli, Neroli, Nasturcia, Rosemary, Water Notes, and Jasmine.
- Reviews:** A section showing "0 reviews" with a "No Data" message and a "Add review" button.

At the bottom of the page, there are navigation links: "-Scent Searcher-", "-My Profile-", and "-Admin Page-".

Figure 21 Perfume detail page

#### 4.5.2. Reviews

Under the details of the displayed perfume, there is a component showing all reviews of the displayed perfume, including if the perfume is liked or not by the author of the review, a rating out of 5 and a review text (Figure 22).

The current user can also add a review to the displayed perfume, by clicking the button of “Add review”, after clicking the button, a modal will appear for the current user to enter the review details, including if the perfume is liked or not by the author of the review, a rating out of 5 and a review text (Figure 23).

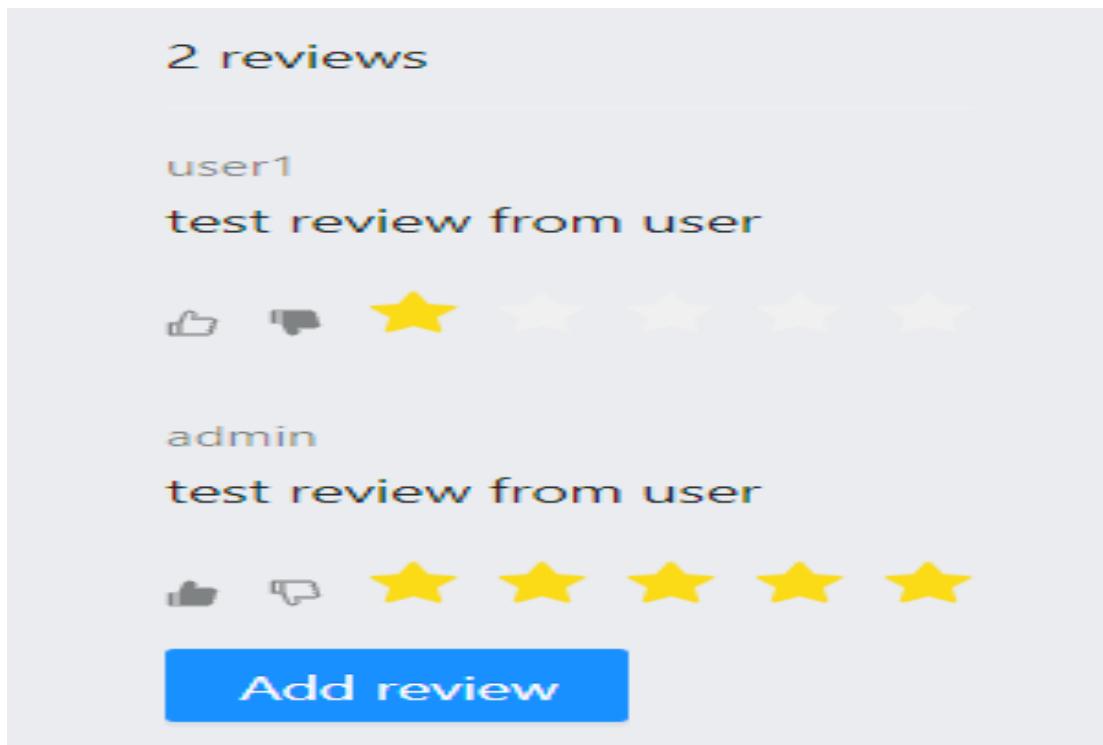


Figure 22 Review list

New Review X

---

\* Like or not:  Like  Dislike

\* Rating:

\* review:

Submit

Figure 23 Modal for add a new review

#### 4.5.3. Backend

The frontend will send the displayed perfume ID to the backend when the page is loading, then the backend will return the details of the required perfume back to the frontend, including perfume name, the image of the perfume, the brand, the production year, the target user, the description and the including notes and related reviews, in the JSON format.

When a user adds a new review, the information will be sent to backend in the JSON format, then the backend will insert the new record into the database, which can be accessed later.

**GET** /id/{accountId}/

**Response Class (Status 200)**  
OK

Model	Example Value
{ "code": "string", "data": {}, "msg": "string" }	

**Response Content Type** \*/\*

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
accountId	1	accountId	path	integer

**Response Messages**

HTTP Status Code	Reason	Response Model
401	Unauthorized	
403	Forbidden	
404	Not Found	

[Try it out!](#) [Hide Response](#)

**Curl**

```
curl -X GET --header 'Accept: application/json' 'http://localhost:8866/id/1/'
```

**Request URL**

```
http://localhost:8866/id/1/
```

**Response Body**

```
{
  "data": {
    "accountId": 1,
    "username": "11111",
    "encodedPassword": "12213123",
    "firstName": "1231",
    "lastName": "1312",
    "gender": "123213"
  },
  "code": "0",
  "msg": "操作成功!"
}
```

**Response Code**

```
200
```

Figure 24 Backend of User center

Next is about the content of the perfume bookmark. Users can view all their favorite perfumes, which will make the front end send a request to the back end to send the user's id to the back end, and the back end can query all the pairs in the bookmark table through the id. It should record the id and return the corresponding perfume id to the front end in the form of a list, and then query the specific information of the perfume through the id of the perfume.

The screenshot shows the API documentation for a GET request to `/getBookmark/{accountId}`. The response class is Status 200 OK. The response model is defined as:

```
{
  "code": "string",
  "data": {},
  "msg": "string"
}
```

The response content type is set to `*/*`. A parameter `accountId` is defined with a value of `1`, which is of type integer and path. Response messages include 401 Unauthorized, 403 Forbidden, and 404 Not Found. There is a 'Try it out!' button and a 'Hide Response' link. The curl command to execute the request is shown as:

```
curl -X GET --header 'Accept: application/json' 'http://localhost:8866/getBookmark/1'
```

The request URL is listed as `http://localhost:8866/getBookmark/1`.

### Response Body

```
{
  "data": [
    2,
    4
  ],
  "code": "0",
  "msg": "操作成功!"
}
```

### Response Code

200

Figure 25 Backend of User center

Users can also add and delete bookmarks. If the user likes a perfume, by clicking the button, the front end will package the perfume id and user id and send it to the backend, and the backend will add a corresponding record in the bookmark table. In this way, the function of adding a bookmark is completed.

If the user clicks the button again, the bookmark will be deleted. Similarly, the front-end sends the perfume id and user id to the backend, and the backend queries the bookmark table for the record, and then deletes it.

**POST /addBookmark**

**Response Class (Status 200)**  
OK

Model Example Value

```
{
  "code": "string",
  "data": {},
  "msg": "string"
}
```

Response Content Type

Parameters

Parameter	Value	Description
accountId	1	
perfumeId	3	

Response Messages

HTTP Status Code	Reason	Response Model
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

[Try it out!](#) [Hide Response](#)

Curl

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: app
```

Request URL

```
http://localhost:8866/addBookmark?accountId=1&perfumeId=3
```

Response Body

```
{
  "data": null,
  "code": "0",
  "msg": "操作成功!"
}
```

Response Code

```
200
```

**DELETE /deleteBookmark**

**Response Class (Status 200)**  
OK

Model Example Value

```
{
  "code": "string",
  "data": {},
  "msg": "string"
}
```

Response Content Type

Parameters

Parameter	Value	Description
accountId	1	
perfumeId	3	

Response Messages

HTTP Status Code	Reason	Response Model
204	No Content	
401	Unauthorized	
403	Forbidden	

[Try it out!](#) [Hide Response](#)

Curl

```
curl -X DELETE --header 'Accept: application/json' 'http://localhost:8866/del
```

Request URL

```
http://localhost:8866/deleteBookmark?accountId=1&perfumeId=3
```

Response Body

```
{
  "data": null,
  "code": "0",
  "msg": "操作成功!"
}
```

Response Code

```
200
```

Figure 26 Backend of User center

## 4.6. Other user's page

### 4.6.1. Link from perfume detail page

If a review in the perfume detail page is created by another user, the username on the review can be clicked to redirect to the author's page (Figure 27).

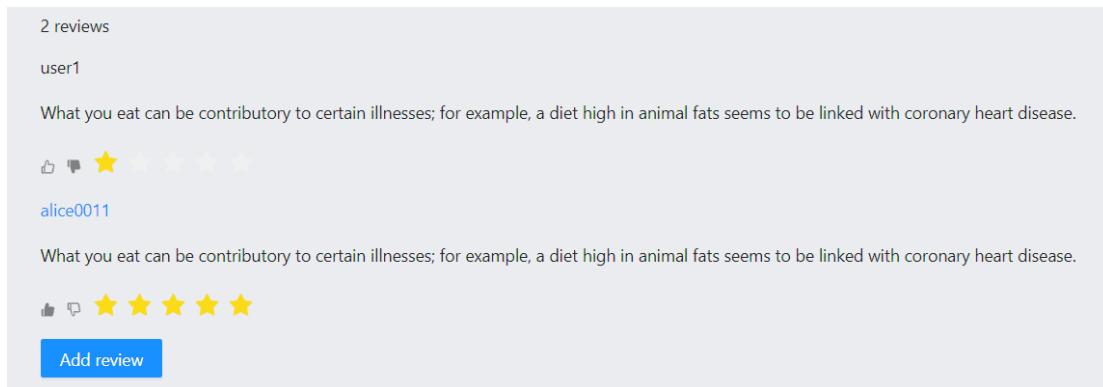


Figure 27 Link in the review component

### 4.6.2. User page

In the user page, there is a component showing the review author's account basic information including the username, the name and the gender. Below the basic information are all the bookmarked perfumes of the author (Figure 28).

Then the current user can directly add bookmarks to all the perfumes displayed in other user's page or showing that the perfume is already bookmarked by the current user (Figure 29).

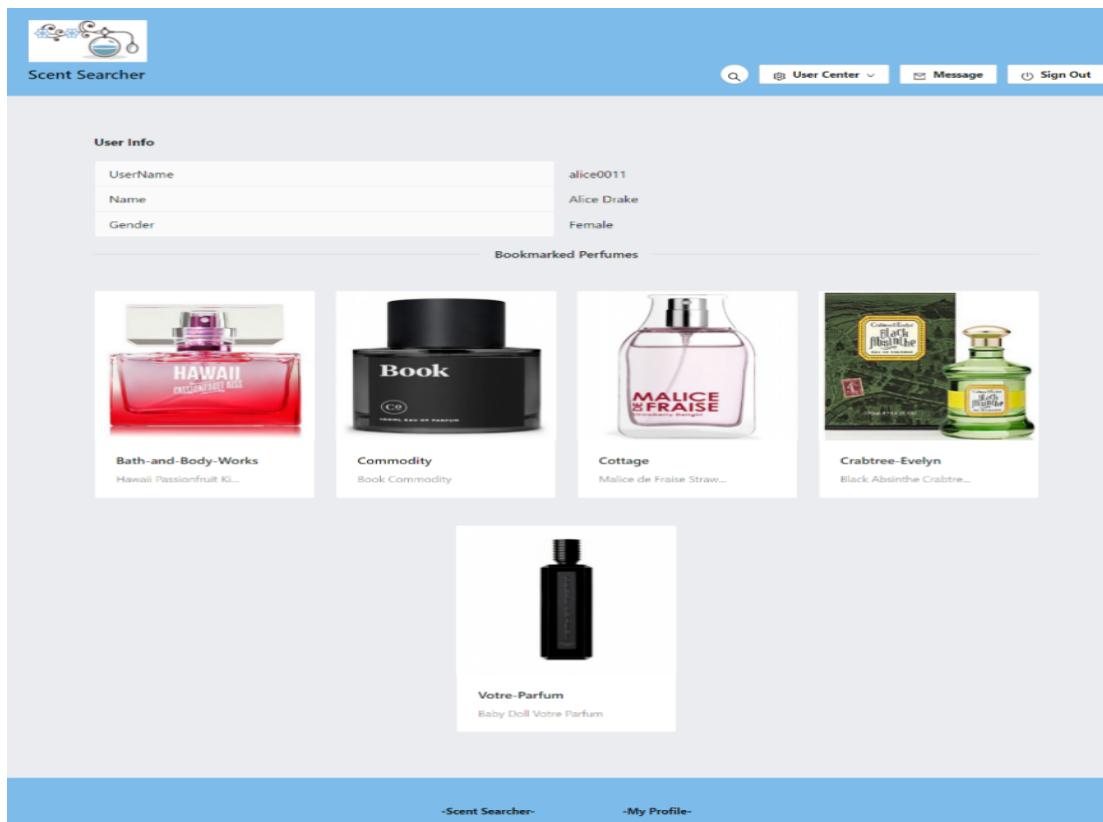


Figure 28 User page

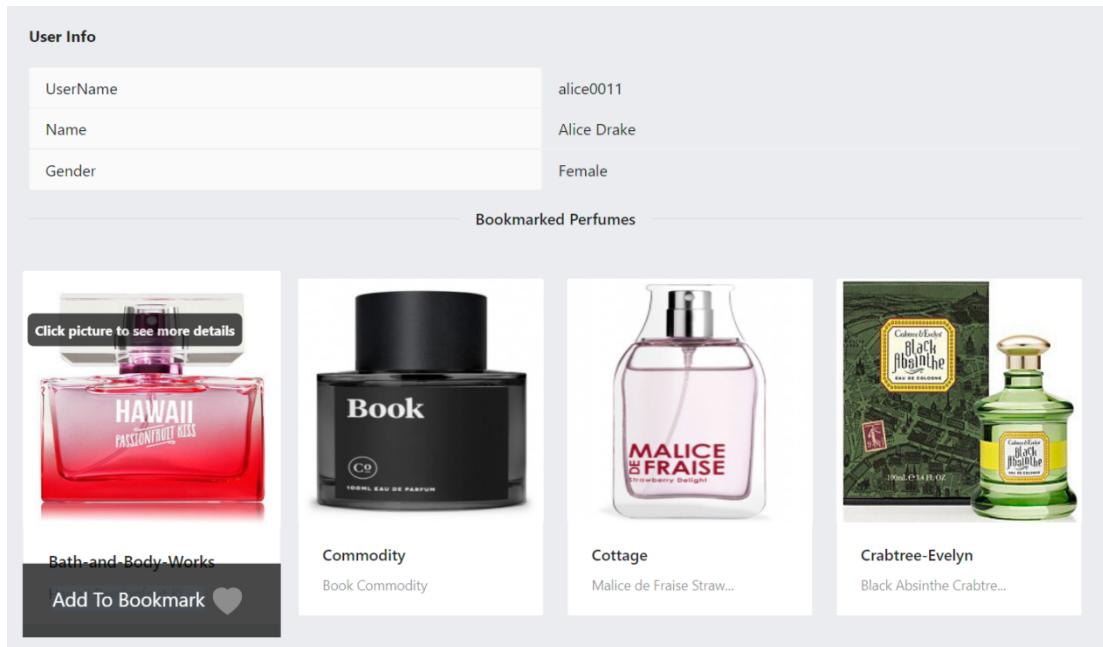


Figure 29 Add bookmark in another user's page

### 4.6.3. Backend

The frontend will send the account ID to the backend when the page is loading, then the backend will return the details of the user back to the frontend, including the

username, first name, last name and gender, along with the bookmarked perfumes, in the JSON format.

## 4.7. Admin

The screenshot shows a web form titled "Hi, Admin :) ~Edit A Perfume~". At the top right is a blue button labeled "Upload Perfume's Image". Below it are five input fields with validation requirements: "Perfume Name", "Perfume Brand", "Date of Release", "Target Audience", and "Description Of Perfume". Each field has a red asterisk indicating it is required. Below these fields is a section titled "Insert notes to the perfume:" containing a text input field with placeholder text "insert notes to the perfume...". At the bottom right is a blue button labeled "Submit Perfume's detail".

Figure 30 Admin page

Only admin or the website facilitator who has the authority can access this page. It is for uploading new perfume's data. The image upload button only allows admin update file ends with jpeg, png or jpg. Each input has its own validation requirement. Once add a perfume successfully, it will show an alert. The note insert bar in this page is reusable from homepage. After adding a perfume successfully, users can search this new perfumes and implement all function to it.

### 4.7.1. Backend

The administrator has the authority to add a new perfume. By entering various information about the new perfume, and then clicking the submit button, the perfume name, brand, release date, customer type, description, and a list of note id will be packaged into a json string and sent to the backend, and the backend will first split the content of the list from it, integrate other information into a new record, store it in the perfume table, and then traverse the note id in the list to form a record of note id and perfume id are stored in the note\_in\_perfume table in turn.

**POST /addPerfumeDetails**

**Response Class (Status 200)**  
OK

Model Example Value

```
{
  "code": "string",
  "data": {},
  "msg": "string"
}
```

Response Content Type

**Parameters**

Parameter	Value	Description	Parameter Type
perfumeDetails	{ "brand": "12345", "description": "12345", "image": "12345", "date": 2021, "notes": [ 1.3 ]	perfumeDetails	body

Parameter content type:

**Response Messages**

HTTP Status Code	Reason	Response Model
201	Created	
401	Unauthorized	
403	Forbidden	
404	Not Found	

[Try it out!](#) [Hide Response](#)

**Request URL**

**Request Headers**

```
{
  "Accept": "*/*"
}
```

**Response Body**

```
{
  "data": null,
  "code": "0",
  "msg": "操作成功！"
}
```

**Response Code**

Figure 31 Backend of Admin Page

# Chapter 5 Implementation Challenges

## 5.1. Data processing

Data processing is divided into two main parts, one for data cleaning and the other for converting all cleaned data into insertable SQL insert statements.

Data cleansing is the re-examination of the original data to ensure data quality. The following issues of data are mainly addressed.

- Data integrity

For the project database, the perfume name and perfume ID must be present; other attributes are not required for now.

The original data was incomplete with few descriptions and images of the perfume. Therefore, a data crawler is needed to solve such problems. This section will be explained separately later in this chapter.

- Data uniqueness

It is important to ensure that perfumes are not duplicated. However, it is still possible that the names of different perfumes are duplicated and only the brand, release date and other attributes can be compared. If all the comparable attributes are the same, then only one of the perfumes is kept.

- Data legitimacy

It is necessary to determine whether all the data is consistent with common sense. For example, we need to determine whether the release date is consistent with common sense.

- Dirty data

There are many duplicate data and many special characters. Therefore, there is a need for detailed observation and analysis of the data and some trial and error.

There is also some data processing for this project. For example, splitting the new attribute: target customer from the original data "perfume title".

When converting data to insertable SQL, there are still some problems. Since the perfume name and perfume brand may contain special characters, an insert statement that simply converts form data to a string type will not work in PostgreSQL. We need to consider how to convert the special characters, especially the quotation marks. The solution was found by reviewing the documentation, which required adding escape characters. For example, when converting a string, the single quotes are converted to two single quotes, one of which takes on the function of escaping in the sql insert statement.

## **5.2. Data crawler**

Before inserting perfume-related data into the table, the first step is to obtain the target perfume data. One of the biggest difficulties is the acquirement of pictures and descriptions of perfumes. Because of the huge number of perfumes, manual input is unrealistic. So using a crawler program to get this information is the final choice. But the choice of the website is another challenge. Choosing Fragrantica.com at the beginning, but it was changed to nosetime.com. The reason is that the anti-crawler mechanism of the former website is very powerful, which makes it impossible to get a lot of perfume pictures and descriptions smoothly. After changing the target website, the perfume picture and description were obtained smoothly. However, due to the limitation of the website area, the perfume description obtained is in Chinese. In order to solve this problem, the function of Chinese translation to English has been added to the program. After solving these problem one by one, the last challenge is how to store the resulting pictures. Images stored in the cloud or converted to base64 encoding are the solutions at first, but problems of these two ways are the same. The amount of data stored in the cloud is very large. If it is converted to base64 encoding, the volume of the content after encoding is also large. It will become larger and require extra work to encode and decode. In the end, storing the picture as a URL to reduce the amount of data was selected. That's all the challenges about data acquirement.

## **5.3. The use of Mybatis-Plus**

For the backend, ‘Mybatis-Plus’ is used to connect to the database and perform various operations such as adding, deleting, modifying, and checking the database. It is very convenient because its service layer and mapper layer inherit many encapsulated methods, which can be called directly. At the same time, for this reason, many methods are still useless. In order to solve the various bugs that appear in the function development, the testing and application of different methods have become an important means to overcome the difficulties. At the same time, a variety of annotations both help and hinder us in the process of completing the function. It may be that two annotations with similar functions have different details that may cause problems, such as ‘@Autowired’ and ‘@Resource’. Through searching information and practical application, the problems have now been solved, and different annotations can be used in appropriate occasions.

## **5.4. Search perfumes by matching notes**

There are two ways to implement the function of search perfumes by matching notes, The first one is as below:

- a) Get a list of perfumes that match all the including notes
- b) Then get another list of perfumes that match all the excluding notes
- c) Convert the two lists to sets then calculate the difference, the difference is the final result

The other implementation is as below:

- a) Get a list of perfumes that match all the including notes
- b) Loop all the perfumes in the above list and exclude if it contains any note in the excluding notes list
- c) After the loop what remains is the final result

These two implementations each has its own advantages and disadvantages, depending on different circumstances:

- a) If the database is much larger than the matching results, getting two lists in the first method will significantly increase the calculation time compared with the second method
- b) If the returned matching list is expected to have a large number of perfumes, looping the list will take too much time than SQL query to the database

In our application, the database is expected to be larger than the returned list of matching perfumes in most cases, so the second implementation is chosen.

## **5.5. Interface designed according to heuristics criteria and design principles**

Design website according to heuristics criteria helps our project to achieve the usability goals (effective, efficient, utility, learnability, memorability).

### **5.5.1. Match between system and real world**

The magnifying glass/mail/shutdown button icon buttons in the header navigation bar shown in Figure 14 and the red/grey heart in the bookmark function shown in Figure 17 are in accordance with the criteria of heuristics of human computer interaction, which can make the system match with the real world.

### 5.5.2. Error message

The visible error messages can help users recognize, diagnose, and recover from errors.

The screenshot shows a form for adding a perfume. The fields and their corresponding error messages are:

- \* Perfume Name: Please input the Perfume Name
- \* Perfume Brand: Please input the Perfume's Brand
- \* Date of Release: Please input the date of release
- \* Target Audience: Please input the perfume's target
- \* Description Of Perfume: Please input the perfume's description

Below the form, there is a note field labeled "Insert notes to the perfume:" with the placeholder "insert notes to the perfume...". A blue button at the bottom right says "Submit Perfume's detail".

Figure 32 Error message in admin page

The screenshot shows a login page with the heading "Welcome to Scent Searcher!". It includes a note for existing users: "If you are an existing user, please login your account :)" and a registration link: "Create Your Own Account in Scent Searcher!".

The login fields are filled with "user1" and ".....". An error message box appears over the password field, stating: "Please match the format requested. Must contain at least 6 or more characters". Below the password field is a link: "Don't have an account yet?".

At the top of the page, a red bar displays the error message: "login failed, please check your password or username."

Below the error bar, the login form is shown again with the same fields and a "Login" button.

Figure 33 Error message of login page

We implement the error message bar in many functionalities (for example, admin page, register page, login page, etc.), and these error messages can also improve the visibility and feedback of scent searcher, which is very helpful for users.

### **5.5.3. Help prevent error**

As show in Figure 16, the error message can also help prevent error.

### **5.5.4. Recognition rather than recall**

The select displaying choice for users in the search notes bar implemented in homepage, admin page, set my favorite notes function and the drop-down gender selection menu are all designed in accordance with “Recognition rather than recall”, which can help user recognize the options they need and increase the usability.

### **5.5.5. The visibility of system status**

After updating data, adding a perfume or bookmark (or remove), if API returns a 200 status, there is a success alert message about user’s status, which improve the visibility of system status. Users can get informed about the state and what is going on through those feedback.

localhost:8080 says

You have updated your password successfully:)

OK

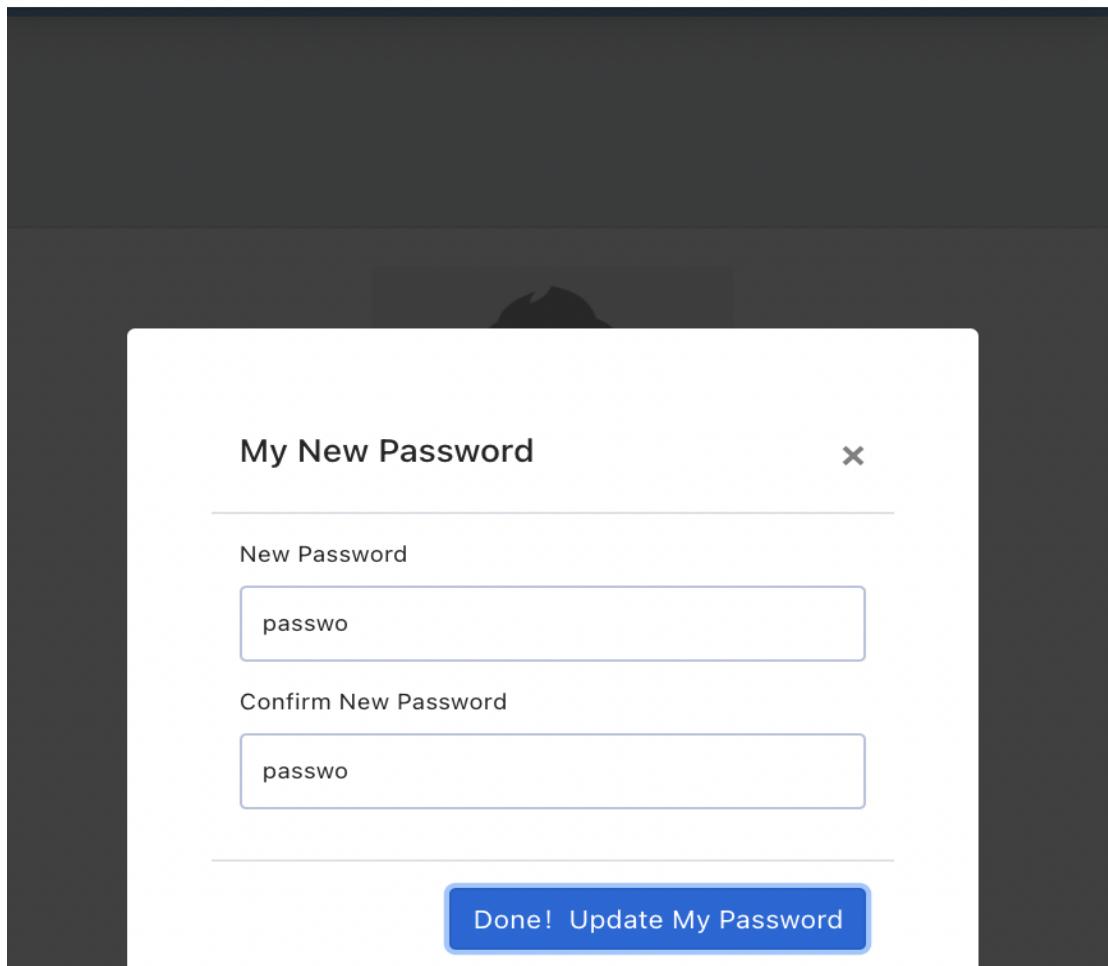


Figure 34 Successful alert message after updating password

#### 5.5.6. Aesthetic and minimalistic design

The colour scheme of our project is white/blue (powderblue, skyblue, dodgerblue, lightblue, lightseagreen). Font color is white or black. The input form, select box, and layout are all designed according to minimalistic to make users feel satisfying and enjoyable as the picture and other details of perfumes will be more important than the website design, and we hope the design of website will not distract users' attention. In this way they can focus more about perfumes. It is also helping users' usage of working memory when they are reviewing scent searcher.

### **5.5.7. User control and freedom**

As shown in register and cancel button in create account page (Figure3.2). In the create account page, the user is able to cancel the register, which gives users freedom to change their mind.

## **Chapter 6     Installation Manual**

The backends are already deployed to AWS and all URLs for RESTful APIs in the frontend are set.

To run this application, simply pull the code in the [FrontEnd\_AWS] branch and run the command below:

**npm install**

After install all the requirements, to run the application in Windows system, please use command below:

**npm start**

To run the application in Linux system, please use command below:

**npm run start\_mac**

## **Chapter 7 Conclusion**

For the past 10 weeks, we have successfully completed the project, Scent Searcher. This is the first time that we have cooperated to develop an online network platform. All functions are completed by the efforts of each team member. The completion of this project was a breakthrough for us, and it also put an end to the two-year study career in UNSW.

When we look back, we successfully cooperated to solve many problems and learned a lot of valuable experience. At the same time, we also found there are still many areas that need to be improved, and there are also many good ideas that cannot be realized due to time constraints. But this is not a pity, because this is another beginning for us, allowing us to embrace our future work with creativity and inspiration.

Finally, we would like to thank all the team members for their dedication. We wish everyone could find our favorite job and improve our technical level after graduation. Thanks also to the mentor and tutor who taught and helped us. Thank you for drawing a successful conclusion to our study career.

## Chapter 8 Reference

Iresearch and Eternal 2020, *China Perfume Industry Research White Paper 1.0*, iresearch, China.

Fragrantica.com. 2021. *Perfumes and Colognes Magazine, Perfume Reviews and Online Community—Fragrantica.com*. [online] Available at: <<https://www.fragrantica.com>> [Accessed 2 August 2021].

Nosetime.com. 2021. *NoseTime*. [online] Available at: <<https://www.nosetime.com>> [Accessed 2 August 2021].

Basenotes.net. 2021. *Independent online guide to perfume — Basenotes.net*. [online] Available at: <<https://www.basenotes.net>> [Accessed 2 August 2021].

Jasonwatmore.com. 2021. *React Hooks + Redux - User Registration and Login Tutorial & Example | Jason Watmore's Blog*. [online] Available at: <<https://jasonwatmore.com/post/2020/03/02/react-hooks-redux-user-registration-and-login-tutorial-example>> [Accessed 2 August 2021].

Pgadmin.org. 2021. *pgAdmin - PostgreSQL Tools*. [online] Available at: <<https://www.pgadmin.org>> [Accessed 1 August 2021].

En.wikipedia.org. 2021. *PostgreSQL - Wikipedia*. [online] Available at: <<https://en.wikipedia.org/wiki/PostgreSQL>> [Accessed 1 August 2021].

En.wikipedia.org. 2021. *Kaggle - Wikipedia*. [online] Available at: <<https://en.wikipedia.org/wiki/Kaggle>> [Accessed 1 August 2021].

Kaggle.com. 2021. *Find Open Datasets and Machine Learning Projects | Kaggle*. [online] Available at: <<https://www.kaggle.com/datasets>> [Accessed 1 August 2021].

Kaggle.com. 2021. *Fragrances - Notes and User Rating*. [online] Available at: <<https://www.kaggle.com/sagikeren88/fragrances-and-perfumes>> [Accessed 1 August 2021].

Reactjs.org. 2021. *Getting Started – React*. [online] Available at: <<https://reactjs.org/docs/getting-started.html>> [Accessed 1 August 2021].

Phillip Webb, S., 2021. *Spring Boot Reference Documentation*. [online] Docs.spring.io. Available at: <<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>> [Accessed 2 August 2021].

Webb, P., Syer, D., Long, J., Nicoll, S., Winch, R., Wilkinson, A., Overdijk, M., Dupuis, C., Deleuze, S., Simons, M., Pavić, V., Bryant, J., Bhavé, M., Meléndez, E. and Frederick, S., 2021. *Spring Boot Reference Documentation*. [online] Docs.spring.io. Available at: <<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>> [Accessed 31 July 2021].

- Alex, B., Taylor, L., Winch, R., Hillert, G., Grandja, J., Bryant, J., Meléndez, E., Cummings, J., Syer, D. and Stein, E., 2021. Spring Security Reference. [online] Docs.spring.io. Available at: <<https://docs.spring.io/spring-security/site/docs/current/reference/html5/>> [Accessed 31 July 2021].
- Projectlombok.org. 2021. Project Lombok. [online] Available at: <<https://projectlombok.org/>> [Accessed 31 July 2021].
- Javadoc.io. 2021. jjwt-api 0.11.2 javadoc (io.jsonwebtoken). [online] Available at: <<https://javadoc.io/doc/io.jsonwebtoken/jjwt-api/latest/index.html>> [Accessed 31 July 2021].
- Reactjs.org. 2021. React – A JavaScript library for building user interfaces. [online] Available at: <<https://reactjs.org/>> [Accessed 31 July 2021].
- Postgresql.org. 2021. PostgreSQL: About. [online] Available at: <<https://www.postgresql.org/about/>> [Accessed 31 July 2021].
- npm. 2021. *react-file-reader*. [online] Available at: <<https://www.npmjs.com/package/react-file-reader>> [Accessed 2 August 2021].
- Jasonwatmore.com. 2021. *React Hooks + Redux - User Registration and Login Tutorial & Example | Jason Watmore's Blog*. [online] Available at: <<https://jasonwatmore.com/post/2020/03/02/react-hooks-redux-user-registration-and-login-tutorial-example>> [Accessed 2 August 2021].
- Mp.baomidou.com. 2021. *introduction | MyBatis-Plus*. [online] Available at: <<https://mp.baomidou.com/guide/>> [Accessed 2 August 2021].