



UNIVERSITY OF
PLYMOUTH

PUSL3190 Computing Individual Project

Draft Report

‘MySizing’ Mobile Application for
Material related Size Recommendations
With React-Native and SpringBoot

Supervisor: Ms. Hirushi Dilpriya

Name: Gunathilaka M Chamodi
Plymouth Index Number: 10819486
Degree Program: BSc (Hons) Software Engineering

Abstract

The worldwide online shopping sales has been rapidly increased in comparison to the past few years, for all the sectors including apparel sector. Although apparel and clothing sales have shown a major growth in the online shopping, the return rates of these sectors also have been highly increased mostly among women. To overcome this issue, creating a system that recommends a best fitting clothing size for women for specific clothing brands would be great. Thus, creating a mobile application where the user can enter the chest size and select a clothing item according to the preferred material with its fitting size would be great solution. Furthermore, user can buy the selected item and can see the stock availability. This application uses React-native as the frontend development and java Spring Boot as the backend development with MySQL database.

Table of Contents

Abstract	2
List of Figures and Tables.....	4
Main body of report	5
Introduction.....	5
Background, objectives & deliverables	7
Objectives	7
Literature review.....	8
Method of approach	9
Gantt Chart.....	9
Programming Languages	9
Testing	9
Version Control.....	11
Code Structure.....	11
Third Party Components and Libraries	13
Use case diagram.....	14
Class Diagram of Proposed System	15
ER Diagram	16
High-level Architectural Diagram.....	17
Requirements.....	18
Functional Requirements.....	18
Non-Functional Requirements	18
Hardware / Software Requirements	19
Networking Requirements (Optional)	19
End-project report	20
Algorithms / Models.....	20
Project post-mortem.....	23
Conclusion	24
Reference List.....	25

List of Figures and Tables

Figure 1:Gantt Chart	9
Figure 2: Postman API request body showing the test HTTP requests	10
Figure 3: MySQL database table of showing how the test post requests are being updated ..	10
Figure 4: Postman API's response to the testing request.	10
Figure 5: GitHub Repo branches	11
Figure 6: React Native Code Structure	11
Figure 7: Config.js file	12
Figure 8: Use Case diagram.....	14
Figure 9: Class Diagram.....	15
Figure 10: ER Diagram	16
Figure 11: High-level Diagram.....	17
Figure 12: Frontend UI of the user feedback page	20
Figure 13: Code set of creating the mathematical model for Cotton clothing brand.....	21
Figure 14: Showing use of AsyncStorage to temporary store logged in user's username.....	22
Figure 15: Code snippet of getting the logged in user's username and chest width.....	22

Main body of report

Introduction

According to a survey conducted by Coresight Research, within US apparel clothing brands and retailers, discovered that the average return rate of the online apparel sales in US is about 24.4% (Coresight & Zheng, 2023). The reasons for high return rates in online clothing are, user body measurements are not matching with the specific clothing clothing brand's size chart measurements and the clothing item's specifications such as material. This developed system focuses on a specific clothing brand with its previous purchases history from users. This approach is more suitable since each clothing brand's size chart measurements could be different from one another. So, it would be better to get each specific clothing brand's purchase history data separately and develop the model to recommend the user fitting size according to the clothing brand. Thus, this system could be improved with adding more clothing brands and its purchase history.

Materials can be stretchable or non-stretchable and should be highly considered when buying the product. According to the questionnaire survey that conducted for this project, 62.5% of respondents stated that the material is important when purchasing a clothing item or an apparel as shown in figure 1 below. Thus, 31.3% of respondents stated that it is very important. None of the respondents say that the material consideration is not important.

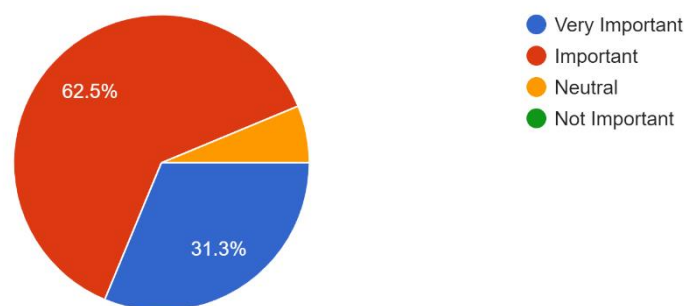


Figure 1: Pie chart showing the importance of clothing materials

Cotton, Polyester, Linen, etc., materials can be considered as non-stretchable and usually require a bigger size of the product. While stretchable materials like Spandex blend fabrics such as Polyester/ Spandex blend or elastane need a smaller size in comparison to non-stretchable fabrics.

In a study that collected data from the in-person interviews were asked about online apparel shopping and its feedback. Both 'product quality' and 'return and exchanges' topics were mentioned 18 times each. Which mentioned more often than the other topics (Yin & Xu, 2021).

Since the quality of the product matters, the developing system provides the user with a feature to select the preferred materials of the clothing according to their likings. Which gives a certainty about the product quality. It can be fully 100% cotton/ polyester/ linen or else it can be a spandex blend material.

According to Statista, comparing to other retail product categories such as shoes, bags and accessories, the highest returned rate which is 26% is for the clothing items among all the categories (Dopson, 2023). Since not all the customers are happy with the online clothing purchasing, developing a system that can recommend the best fitting size before buying the actual product would be a better solution.

The developing system is called 'Material-related size recommendations', can recommend the perfectly fitting size for the user by getting one user specific body measurement such as user's chest size. Then that measurement would be compared with the previous user's purchase history to make that size recommendation. In here it only focuses about a specific clothing brand and can use this exact solution to expand this system while collaborating with several other clothing brands.

This system collects the data about customer's order history through a page where user can enter their chest width, material of the clothing they bought, and the fitted size when do both physical and online shopping. Using all these data, system can give the user a material-related size recommendation.

Data from several surveys found that men have more trust in online shopping (purchases) than women who reported have lesser trust (Kumar & Singh, 2014). Therefore, the developing system would be mainly focused on women and the age range of 20-54. Since women found difficulties with the materials related sizes mostly.

The developing system uses a mobile application which provides user-friendly user interfaces (UIs) for the user to get the recommended fitting size of the clothing item. In addition to that some more features are available in this application such as user feedback which could improve the reliability of the system.

Through the react native mobile application, the user can enter the user-specific body measurement which is the chest size, and that data would be sent to the related database tables to give the best fitting size. And technologies as react native for the mobile application frontend, MySQL database with java spring boot for the backend would be use as a full- stack.

Background, objectives & deliverables

Objectives

Main objectives

1. Recommend the best fitting clothing size for each user according to the user chest width considering the material attributes.
2. Target users are the women who's in between age range of 20-54 for this system.
3. Ability to select a clothing item according to the preferred clothing material.
4. Displays the current stock availability of an item.
5. Able to purchase and make the payment through PayPal gateway.
6. Provide a user feedback system which can improve the reliability of the model.
7. Improves online shopping experience.

Literature review

Using 'FashionFit' system, user can virtually try on any unclothing branded clothing item from any retailers using AI. The 3D pose maps the user's body and Neural body fit model creates 3D models of the body according to the body shapes. Then able to virtually try on the clothing using GANs framework in machine learning (Hashmi, et al., 2020). In that system it uses 3D body pose to map the user's body based on an image.

A system where gathers the data about user's previous purchases, and with the specific clothing item's details, predicts the suitable size of for user using the GBM classifier as reduces the return rates (Abdulla, et al., 2019). By analyzing a collection of data, system would be able to tell the size of clothing that is recommended for the user and without fully need of the visualization.

Research about 'Avatar manager system', developed a system where the user can try on the clothing item, he/she likes on an avatar when do online shopping. Men and women can select a suitable body type which matched for each user from the given options. Then after should select the preferred body size such as small or large or extra-large, etc. In the end of this process user can adjust the avatar's body measurements compared to the user's body measurements. Then it would display how the selected clothing item would fit on according to the user's body measurements (Polke & Kumari, 2018).

THE FIT (Korea) implemented an AI-based system in online shopping malls, that recommends the suitable shoe size for the customer. This application collects the data about customer's order history and compares that data with the customer's actual foot size and recommends a shoe size from currently available shoe products (Yuan, et al., 2021).

According to a system that implemented using a Hierarchical Bayesian model. It can recommend a size for the product by checking the highest probability about customer keeping the product without returning it and according to their preference through using a survey conducted about millions of purchase data (Romain Guigourès, 2018).

Method of approach

Gantt Chart

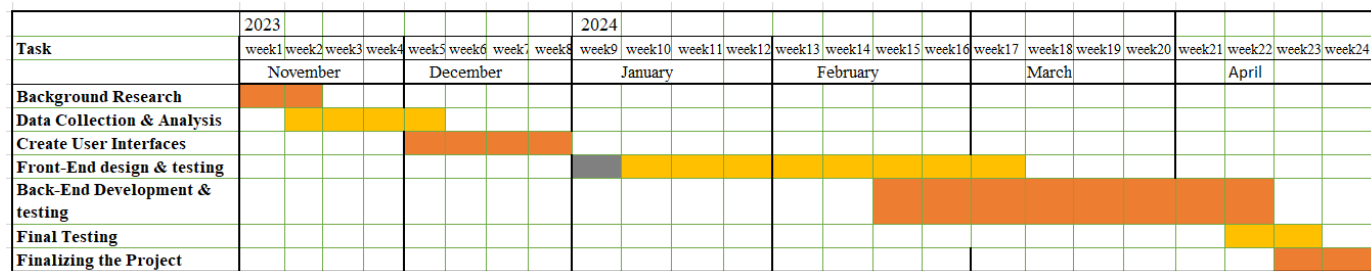


Figure 2:Gantt Chart

This developed system used a hybrid approach methodology, where it is a combination of waterfall and agile methodologies. As the Gantt chart shows above in the figure 2, in the week nine the project was on hold, then started back in 10th week where the frontend design was first started and continued straightly till week 14. This can be considered as waterfall approach where UI changes not applied later in the project. In 15th week, have started agile approach and separated the work into iterations. Thus, in an iteration the frontend and the backend of a feature is developed while do testing for that specific feature. Likewise, the iterations approach in agile have executed until the week 22. Then after final testing for the whole application is done. Finalizing the project is done at week 24 (end of the April).

Programming Languages

React-native framework is used to develop the frontend of this mobile application, while for the backend it uses Java Spring Boot framework with MySQL database. This can be considered as a great stack for developing a mobile app. MySQL is hosted on a remote server using port 3306 which use to communicate to a network. For java spring boot it is using the tomcat server with port 8080 to run that its local server and the React's metro is running on the port 8081.

Testing

Uses Postman application as a tool to test the HTTP requests and its responds. Under collections have created a new collection called 'Sizerecom'. As an example a new POST request is send to the endpoint and that request would be sent in a JSON format as shows in the figure 17 below,

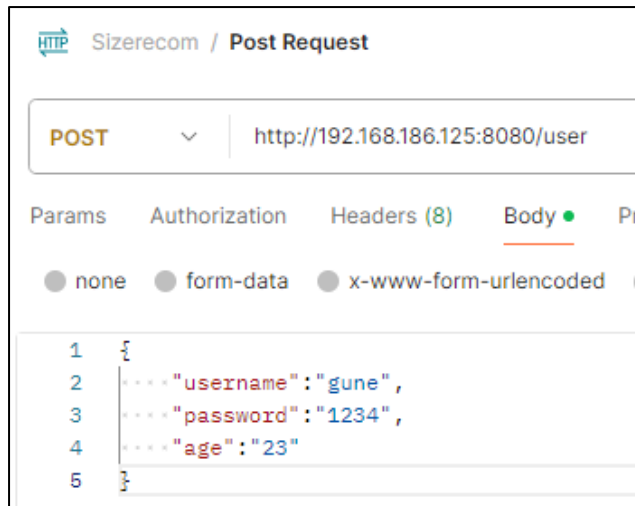


Figure 3: Postman API request body showing the test HTTP requests

When the request is sent, it would update the MySQL database tables as in the figure 3 below. So, then it could test the APIs in this application as it uses 'PostMapping' annotation in java spring boot to handle and send the POST requests.

	id	age	password	username
▶	1	22	qwer	Chamodi
	2	23	1234	gune
✱	NULL	NULL	NULL	NULL

Figure 4: MySQL database table of showing how the test post requests are being updated

A new GET request is made to get (retrieve) the data and view all users in endpoint of '/allusers'. As can see in the figure 4, the ID is auto incrementing by one using the '@GeneratedValue' annotation with '@Id' annotation. The Id then used as the primary key.



Figure 5: Postman API's response to the testing request.

Version Control

For every project it is important to use good amount of version control, to keep backups and for code maintainability. As for this developing application, the data is committed/ saved on two different branches as 'testdev' and main as in the figure 6 below. First each feature's backend part and frontend part would be committed to the testdev subbranch, then after if both frontend and backend of that feature is working fine, it would be committed to the main branch from the 'testdev' branch.

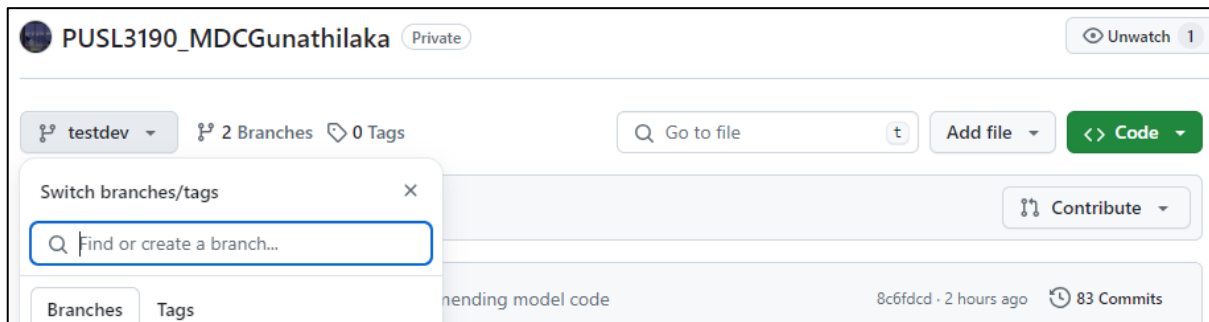


Figure 6: GitHub Repo branches

Code Structure

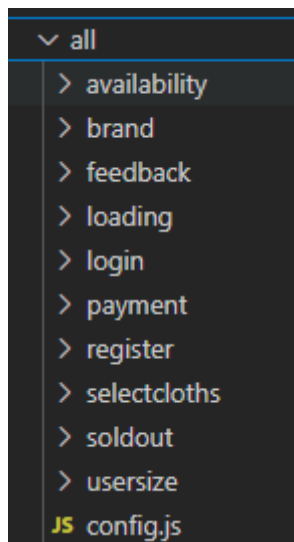


Figure 7: React Native Code Structure

In the react-native frontend folder, a folder called 'all' is there as in figure 6, where it has sub folders named after each feature of this developed application. The 'config.js' is the file that contains the endpoint path such as 'http://<IP>:8080'. This path is allocated to a variable and that variable would be imported in every file to avoid code duplication and improve flexibility as shown below in figure 7.

```
all > JS config.js > ...  
1 export const BASE_URL = 'http://192.168.1.59:8080';|
```

Figure 8: Config.js file

When talking about the java spring boot backend, the advantage of using this is having a clear code structure where it is easy to maintain. Spring boot's especially is to create microservices. In this developed application it has allocated a feature into an iteration. Thus, as in the provided figure 9 below controller, model, repository folders have created and in the model folder it has all the codes to create the database model into the MySQL database. In the controller folder it has mentioned about the ends points with the related codes to perform the tasks of each feature. Furthermore, if has to find any field using its field name, have to mention it in a repository interface connected to its model.

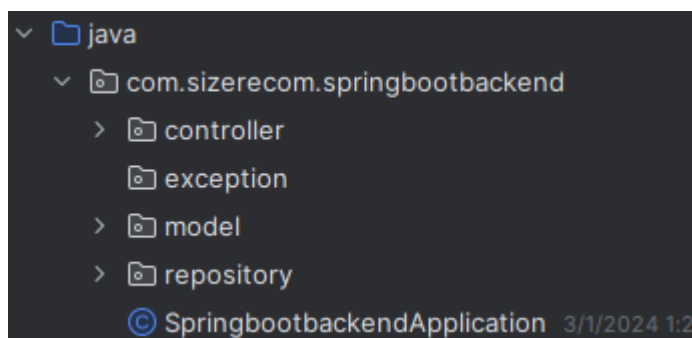


Figure 9: Spring boot Code Structure

Third Party Components and Libraries

For the mobile application have used third party components and libraries to add more additional functionalities to the system while creating the application well organized by enhancing the user experience also.

1. @react-navigation/native
 - This can navigate the user from one screen to another screen using the navigate (‘’) method.
2. @react-navigation/stack
 - This creates a stack, so if put ‘A’ screen on top and ‘B’ screen under that, this library can stack them and would go to the screen ‘A’ first then navigates to the screen ‘B’.
3. npx react-native link react-native-gesture-handler react-native-screens react-native-svg
 - Since the ‘Button’ component in react-native does not include many options when it comes to editing or styling the component, in here it uses a component called ‘TouchableOpacity’ which can easily be styled and act as a button.
4. react-native-screens library
 - uses to create a stack where navigate between the screens.
5. @react-navigation/native @react-navigation/bottom-tabs
 - to create the bottom navigation bar in most of the pages
6. @react-native-picker/picker
 - To create the dropdown list in the ‘Material.jsx’ page
7. Axios library
 - uses this library to make http requests from the react-native app to the server.

Use case diagram

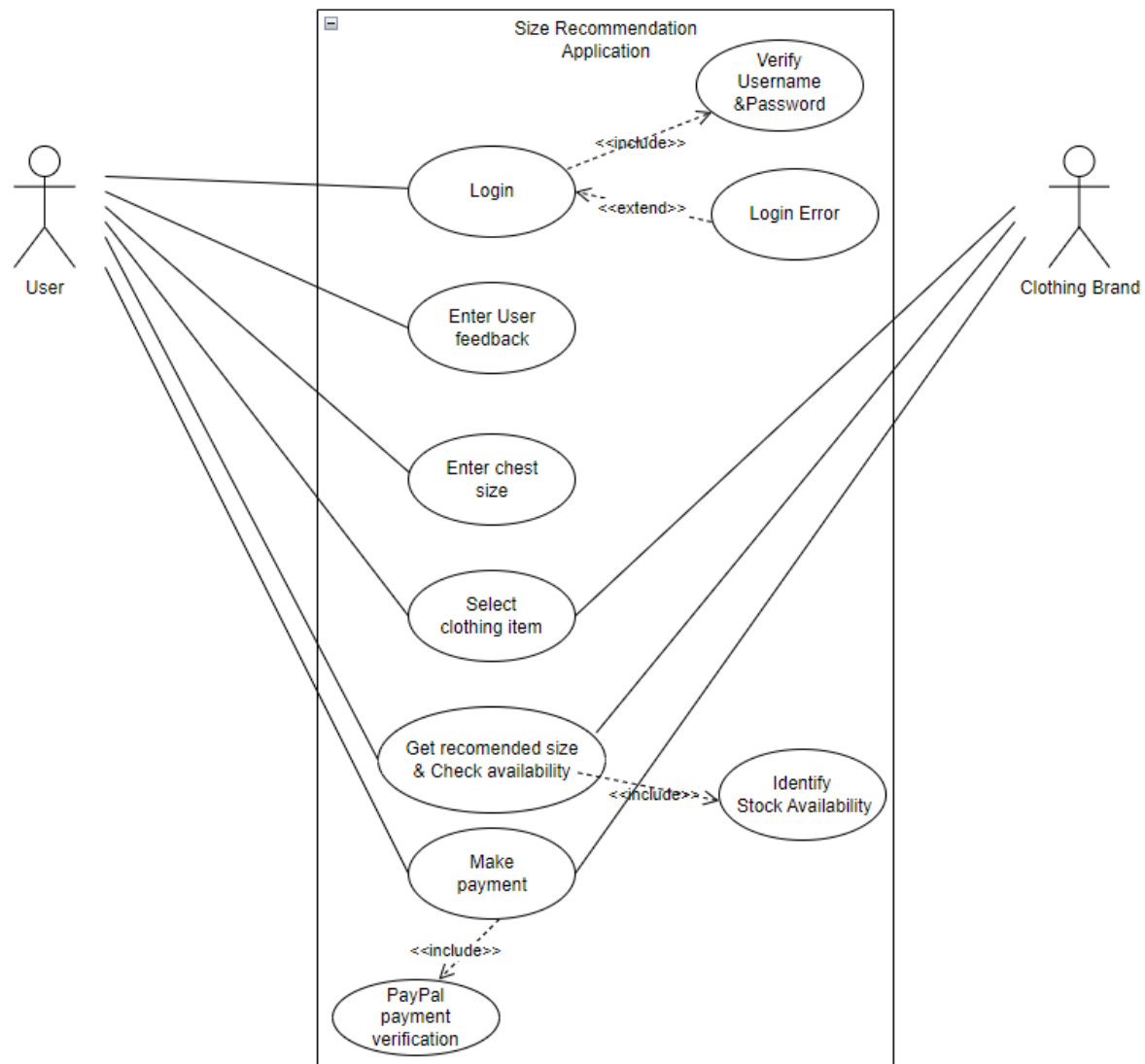


Figure 10: Use Case diagram

The system name is 'Size Recommendation system' and actors are external objects that can be listed as primary actors, secondary actors. The primary actor in the developing system is the user/ customer which displays on the left side of the container/ system in the figure 10 shows above. These people can use the system to get the size recommendations.

In the other hand the secondary actor in here is the clothing clothing brand shown in the right side of the figure 10. It reacts when the user/ customer selects an item and checks the stock availability and also when making the payment.

As for the use cases it has base and included use case. As an example, the login 'base use case' needs 'included use case' to verify and authenticate the user. Furthermore, this system has 'extended use case' which the relationship with the 'base use case' does not happen every time, such as when login to the system the login error message would not occur every time.

Class Diagram of Proposed System

The visibility of the methods and attributes are shown using the + or – symbol public and private access specifiers accordingly as shown in the figure 11 below. User can give feedback as well as can enter the chest size. After entering the user size, user could select a clothing item from the list, then from that would give user the best fitting size recommendation and the stock availability.

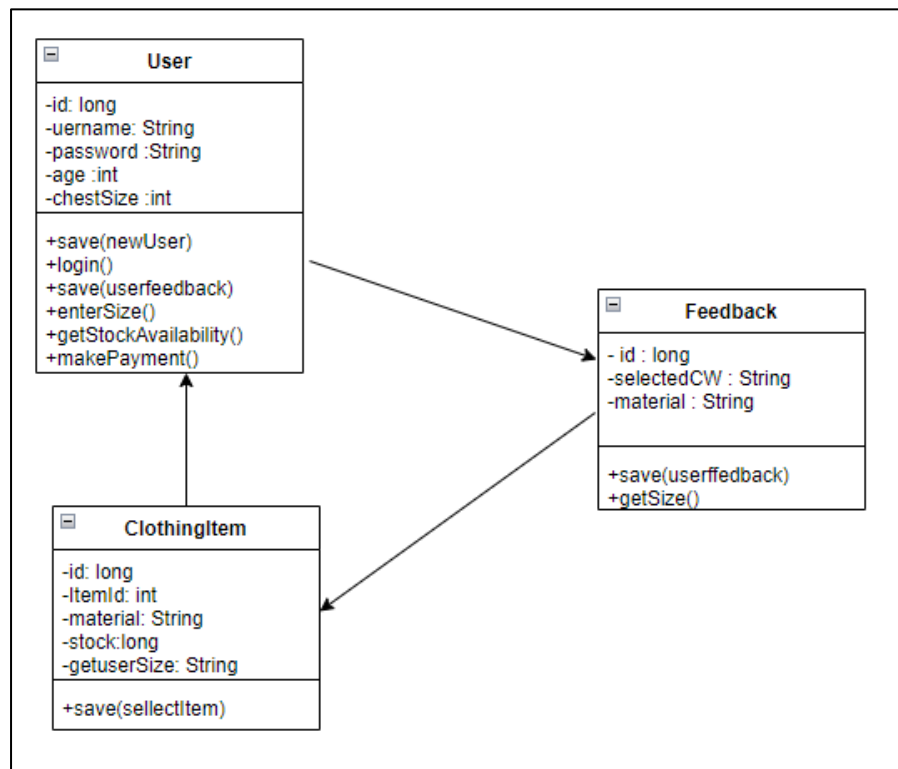


Figure 11: Class Diagram

ER Diagram

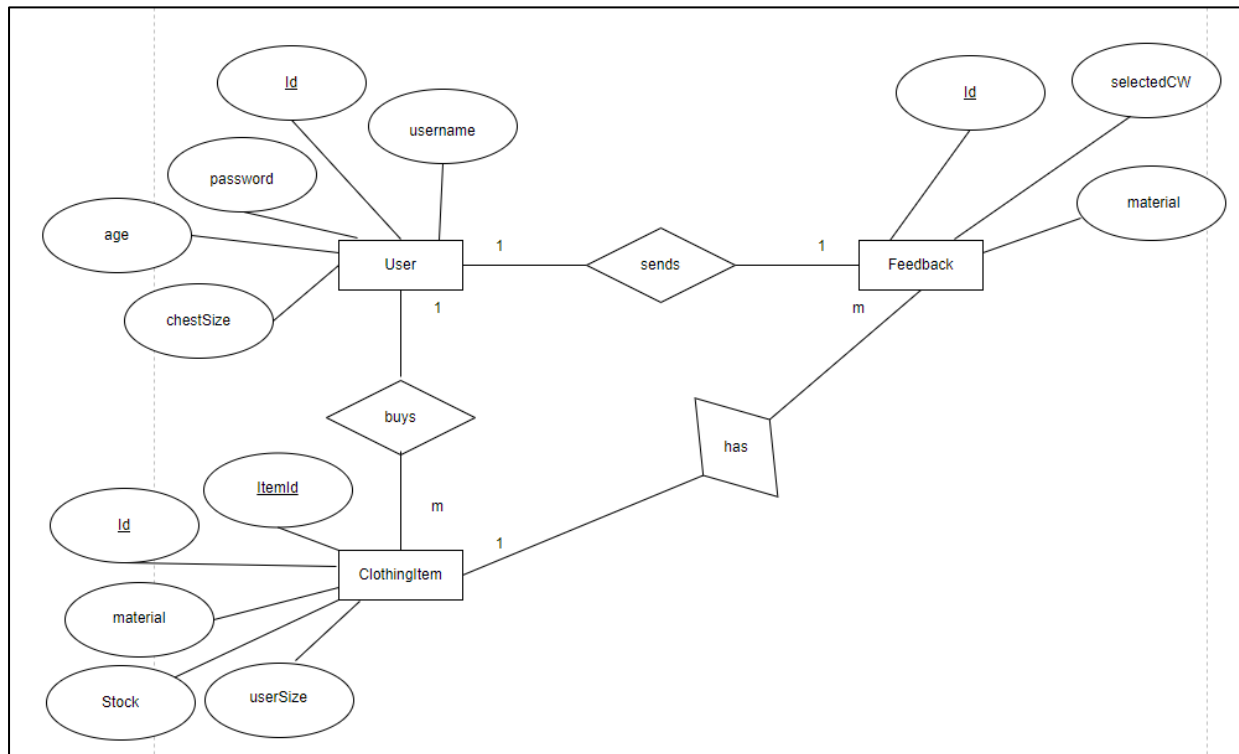


Figure 12: ER Diagram

The entities of this system are user, feedback, and clothing items. As shown in the figure 12, user can send feedback, also assume that one user sends only one feedback while feedback is sent by a user. Furthermore, the clothing items has user feedback and the user can also buy the selected clothing item.

High-level Architectural Diagram

As for the high-level diagram such changes were developed when compared to the previous high-level diagram. As according to the new high-level diagram in the figure 13 below, from react-native application (frontend), would send the http request to the server. Then after processing the requests and it performs the specific tasks while connecting and interacting with MySQL database as necessary.

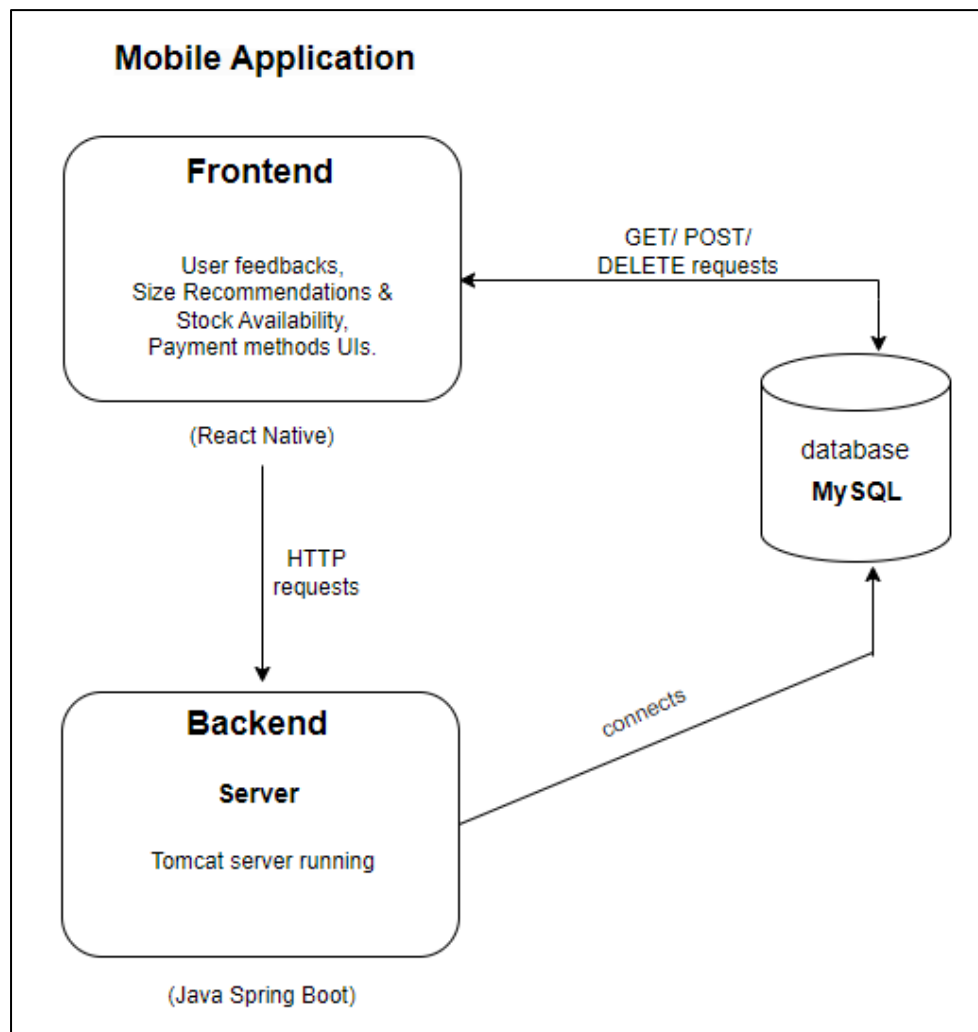


Figure 13: High-level Diagram

Requirements

Functional Requirements

- Displaying the best fitting UK size for the user selected clothing item.
- Could select the clothing item according to the preferred material.
- Ability to give user feedback about previous purchases made.
- Feature to see the stock availability.
- Able to make payments using PayPal.
- Able to login to the system using the registered user details.

Non-Functional Requirements

- Security:

According to the figure 14 below, only if the user exists in the system the user can login. In a register table users can register so the new record with the user details would be added. Then after when the user logs in, data would be retrieved and would be checked.

```
@PostMapping("/userlogin")
public ResponseEntity<String> loginmethod(@RequestBody Login oldUser) {
    //This line retrieves a Register object from the registerRepoObj repository using the username
    // The Register object represents a user that has already registered.
    //call the Register
    Register existingUser = registerRepoObj.findByUsername(oldUser.getLoginusername());

    //if the it is not a existing user, so the password and th username of old user doesnot match,
    if (existingUser != null && existingUser.getPassword().equals(oldUser.getLoginpassword())) {
        //if true- it returns a http rspnse saying login successfull
        return new ResponseEntity<> (body: "Login Successful", HttpStatus.OK);
    } else {
        //if false then return the error msg
        return new ResponseEntity<> (body: "Wrong Password or Username", HttpStatus.UNAUTHORIZED);
    }
}
```

Figure 14: Code set of user authentication

- Accessibility

The application can run on Android smart phones. Though it has only tested on android, this mobile application should be able to run on IOS as well since react-native is cross-platform.

- Maintainability

And have also used getters and setters in java encapsulation as a security measurement. Which helps to improve maintainability of the code structure and the security.

- UI and UX

Provides better user experience and user-friendly UI while securing the data.

Hardware / Software Requirements

Hardware

A Mobile phone (Android preferred) and a computer that has accessed to Wi-Fi and has enough RAM (ex: 8GB and 8GB< would be preferred) and CPU.

Software

Node js should be installed prior to installing this application. Thus, Visual studio code IDE is used to run the react-native application. And IntelliJ IDEA is used for the java spring boot backend with MySQL workbench application to run the database.

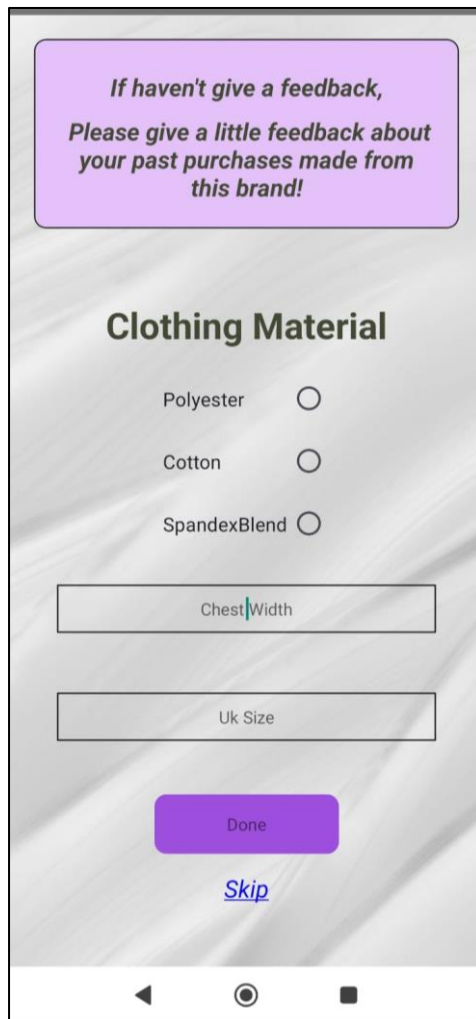
Networking Requirements (Optional)

To run this application the development machine and the user's mobile phone needs to be in the same (Wi-Fi) network, so the development machine's IP address can be accessed by the mobile phone. Since the development machine would be the remote server and have to use its IPv4 address in the react-native's frontend for axios request URL, to interact with the java spring boot with MySQL backend.

End-project report

Algorithms / Models

This developed project uses a mathematical model to give size recommendations to the users. This application provides the user a user feedback page where it gathers data about the material of the clothing she bought, the chest width and the fitted UK size as in the figure 15 UI shows. Thus, using these feedback data can analyze how the sizes are being vary with the material of the clothing.



The image shows a mobile application interface for a user feedback page. At the top, there is a purple rounded rectangle containing the text: "If haven't give a feedback, Please give a little feedback about your past purchases made from this brand!". Below this, the title "Clothing Material" is displayed in bold. Under the title, there are three radio button options: "Polyester", "Cotton", and "SpandexBlend". Below the radio buttons are two text input fields: "Chest Width" and "Uk Size". At the bottom, there is a purple "Done" button and a blue "Skip" link. The interface is set against a background of a light-colored fabric texture. At the very bottom, there is a white bar with three black icons: a back arrow, a home circle, and a recent apps square.

Figure 15: Frontend UI of the user feedback page

The creation of this model is as follows,

```
no usages  ◀ chamodi2001
@GetMapping("/cotton/recommend/{cw}")
public int getSizeRecommend(@PathVariable double cw) { // the logged-in user's chestwidth is called cw
    // find all the cotton material feedback data from fbCotton model/ table
    List<fbCotton> fbCotDataObj = fbCottonRepoObj.findAll();

    //should group by the UK sizes
    Map<Integer, List<fbCotton>> groupedByUkSize = fbCotDataObj.stream()
        .collect(Collectors.groupingBy(fbCotton::getUkSize)); //this groups/ seperate the data by

    //getting the most appeared chest width size for each uk sizes
    Map<Integer, List<Double>> mostFrequentChestWidthsByUkSize = new HashMap<>();
    for (Map.Entry<Integer, List<fbCotton>> entry : groupedByUkSize.entrySet()) {
        Map<Double, Long> chestWidthCounts = entry.getValue().stream()
            .collect(Collectors.groupingBy(fbCotton::getUserCw, Collectors.counting()));
        //getting specific cw and its count
        List<Double> mostFrequentChestWidths = chestWidthCounts.entrySet().stream()
            .filter(e -> e.getValue().equals(Collections.max(chestWidthCounts.values())))
            .map(Map.Entry::getKey)
            .collect(Collectors.toList());
        //make a list of the mostly appeared cws
        //stored in the bellow variable- the uk size and its most frequently appeared cw
        mostFrequentChestWidthsByUkSize.put(entry.getKey(), mostFrequentChestWidths);
    } //the entry key is where it stores the uk size and its most frequently occurring chest widths
    Map<Integer, DoubleSummaryStatistics> chestWidthStats = mostFrequentChestWidthsByUkSize.entrySet()
        .collect(Collectors.toMap(
            Map.Entry::getKey,
            e -> e.getValue().stream().collect(Collectors.summarizingDouble(Double::doubleValue))
        ));

    //Calculate the min and max cw range for each uk size
    List<Integer> fittingUKSizes = new ArrayList<>(); //list the fitting uk sizes for logged in user
    for (Map.Entry<Integer, DoubleSummaryStatistics> entry : chestWidthStats.entrySet()) { //have the
        if (cw >= entry.getValue().getMin() && cw <= entry.getValue().getMax()) { //getting the min and
            fittingUKSizes.add(entry.getKey());
            //if a fitting uk size is found for the user logged in cw, then add that to the fittingUKS
        }
    }

    //Check if the user's chest width falls within the min and max range of any UK size.
}
```

Figure 16: Code set of creating the mathematical model for Cotton clothing brand.

According to the figure 16 shown above, the feedback data would be saved to the related material tables such as ‘fbSpandex’ table, ‘fbCotton’ table, ‘fbPolyester’ table in the MySQL database. Then that data would be grouped by the UK sizes, such as for UK 6 what are the chest width sizes that has entered by other users.

Then after, from grouped UK size it would find the most appeared chest widths while calculating what are the minimum and maximum chest width range for that grouped UK size. From that it would create a range where it has the specific UK size with its chest width size range.

Furthermore, after this analyze or the calculation, when the system retrieves the logged in user's chest width, it is able to find the best fitting UK size for that person.

```
// await axios.post('http://192.168.239.125:8080/userlogin', Loginde
axios.post(`${BASE_URL}/userlogin`, Logindetails)
.then(response => {
  if (response.status === 200) {
    AsyncStorage.setItem('loginusername', loginusername);
    Alert.alert('Login Successful');
    // navigateTo.navigate('HomeStack');
```

Figure 17: Showing use of AsyncStorage to temporary store logged in user's username.

```
useEffect(() => {
  if (screenName === 'PolyesterScreen') {
    // Fetch data for polyester
    // from AsyncStorage get the username of the ogged in user
    AsyncStorage.getItem('loginusername')
    //then using the 'username' argument, check if it exists.
    .then(username => {
      if (username) {
        //getting that logged in user
        axios.get(`${BASE_URL}/polyester/chestwidth/${username}`)
        .then(output => {
          setChestWidth(output.data); // update the state value of the logged
          //Recomemding a uk size
          axios.get(`${BASE_URL}/polyester/recommend/${output.data}`)
          .then(output => {
            setUkSize(output.data);
          })
          .catch(error => {
            console.error('No enough data to recomend a size', error);
          });
        })
        .catch(error => {
          console.error('There was an error!', error);
        });
      }
    })
  }
});
```

Figure 18: Code snippet of getting the logged in user's username and chest width.

Project post-mortem

As for the self-evaluation, this developed system could be more improved from training the model with top design types such as A-Line, Off-Shoulder, Button-Up, etc., with the material related data from users purchase history.

Moreover, this developed system was planned to be developed using the Android studio, but since using Android studio with java spring boot and MySQL database as the backend is somewhat difficult. This developed system uses MySQL remote server with a localhost so, the database has to be hosted for another user to use. Thus, it would have been better to use a structured relational cloud database for this project.

Conclusion

Finding the best fitting size for a woman can be a bit tricky and difficult in online shopping. This developed system would give the size recommendations of a specific clothing brand based on the material attributes. Thus, in this project has mentioned only one specific clothing brand called 'clothing brandA', but for further development can use different clothing brands with its customer purchase history details to train the model which improves scalability.

For this system, to develop the mathematical model it was planned to use the questionnaire result data. However, since the responses from the questionnaire was not enough, the accuracy of training the model using those data would not be reliable.

So, this developed system used dummy data from the user feedback page about user's previous purchases data such as material, fitted UK size and material details. Though it has used dummy data here, in real-life scenario, this application can be trained using the real customers.

To make the application more reliable in future development, can train the model according to the design of the blouses/ tops as well. Such as A-Line, Off-Shoulder, Button-Up, Peplum, Wrap, etc... blouse design types. The reason of limitation to t-shirts, blouses and tops is that the trousers, frocks having major difference in length, design, and cuts. Though it is possible to reach those types of appear, in this project it only focuses on the t-shirts, blouses and other tops of women. Thus, in further development this application can be used by men, from getting the feedback from men about the previous perchance.

Reference List

1. Coresight & Zheng, S., 2023. The True Cost of Apparel Returns: Alarming Return Rates Require Loss-Minimization Solutions. [Online]
Available at: <https://coresight.com/research/the-true-cost-of-apparel-returns-alarming-return-rates-require-loss-minimization-solutions/>
[Accessed 13 November 2023].
2. Yin, W. & Xu, B., 2021. Effect of online shopping experience on customer loyalty in apparel business-to-consumer ecommerce, s.l.: Sage Journals.
3. Dopson, E., 2023. Ecommerce Returns: Expert Guide to Best Practices (2024). [Online]
Available at: <https://www.shopify.com/enterprise/ecommerce-returns>
[Accessed 12 November 2023].
4. Kumar, D. V. & Singh, R., 2014. Women Online Shopping: A Critical Review of Literature, s.l.: SSRN.
5. Administration, I. T., 2021. eCommerce Sales & Size Forecast. [Online]
Available at: <https://www.trade.gov/ecommerce-sales-size-forecast>
[Accessed 10 November 2023].
6. Vaghasiya, C. & Sitapara, J., 2023. A STUDY ON RETURNS RATES IN THE FASHION INDUSTRY WITH REFERENCE TO AN E-COMMERCE BUSINESS, s.l.: JETIR.
7. Cullinane, S., Karlsson, E., browne, m. & Wang, Y., 2017. Retail clothing returns: A review of key issues, s.l.: ResearchGate.
8. ReactNative, n.d. Introduction. [Online]
Available at: <https://reactnative.dev/docs/getting-started>
[Accessed 28 February 2023].
9. Hashmi, M. F. et al., 2020. FashionFit: Analysis of Mapping 3D Pose and Neural Body Fit for Custom Virtual Try-On, s.l.: IEEE.
10. Abdulla, G. M., Singh, S. & Borar, S., 2019. Shop your Right Size: A System for Recommending Sizes for Fashion products, s.l.: ACM Digital Library.
11. Polke, N. & Kumari, S., 2018. Avatar Manager System for Online Fashion Clothing APP, India: IEEE.

12. Yuan, Y., Park, M.-J. & Huh, J.-H., 2021. A Proposal for Clothing Size Recommendation System Using Chinese Online Shopping Malls: The New Era of Data, s.l.: MDPI.
13. Romain Guigourès, Y. K. H. K.-S. S. B. S., 2018. A hierarchical bayesian model for size recommendation in fashion. s.l., ResearchGate.