

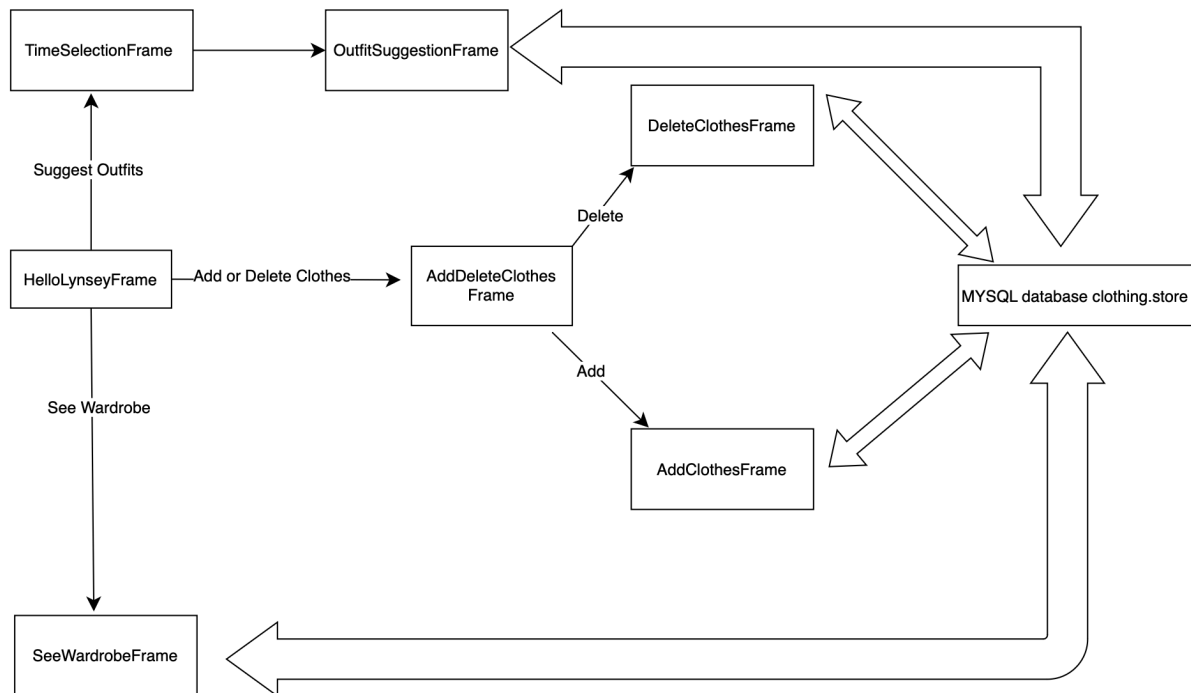
# Your Own Wardrobe - Design

---

## Table of Contents

- 1. Basic Design Overview**
- 2. Input Data types - table**
- 3. Output Data types- table**
- 4. Process Description - Flowcharts**
  - a. Main Flowchart of the Basic program
  - b. In the case where the user chooses to Add or Delete Clothes
  - c. In the case where the user chooses to See Wardrobe
  - d. In the case where the user chooses Suggest Outfit
- 5. Process Description - UML Diagrams**
  - a. UML GUI Classes
- 6. Software Interfaces**
- 7. Database Dictionary and Design**
- 8. Test Plan**

## 1. Basic Design overview



## 2. Input data types - table

| Data                                | Comments  |
|-------------------------------------|---|
| <b>Choices of buttons</b>           | Based on which buttons are pressed on the homepage user interface, the program will be redirecting the user to a different frame or get her to exit the database application as well as confirming a selection/addition/deletion. |
| <b>Jpegs</b>                        | In order for the digital Wardrobe to display the digital wardrobe for the user, the user should be able to input images into the database.  |
| <b>Preference level input (int)</b> | When inputting an item into the digital wardrobe, the user needs to input a number from 1 (lowest level of preference) to 5( highest level of preference)   |
| <b>Worn Button (boolean)</b>        | The user will be inputting a boolean by pressing on the button “worn” so that it declares the item that way.  |

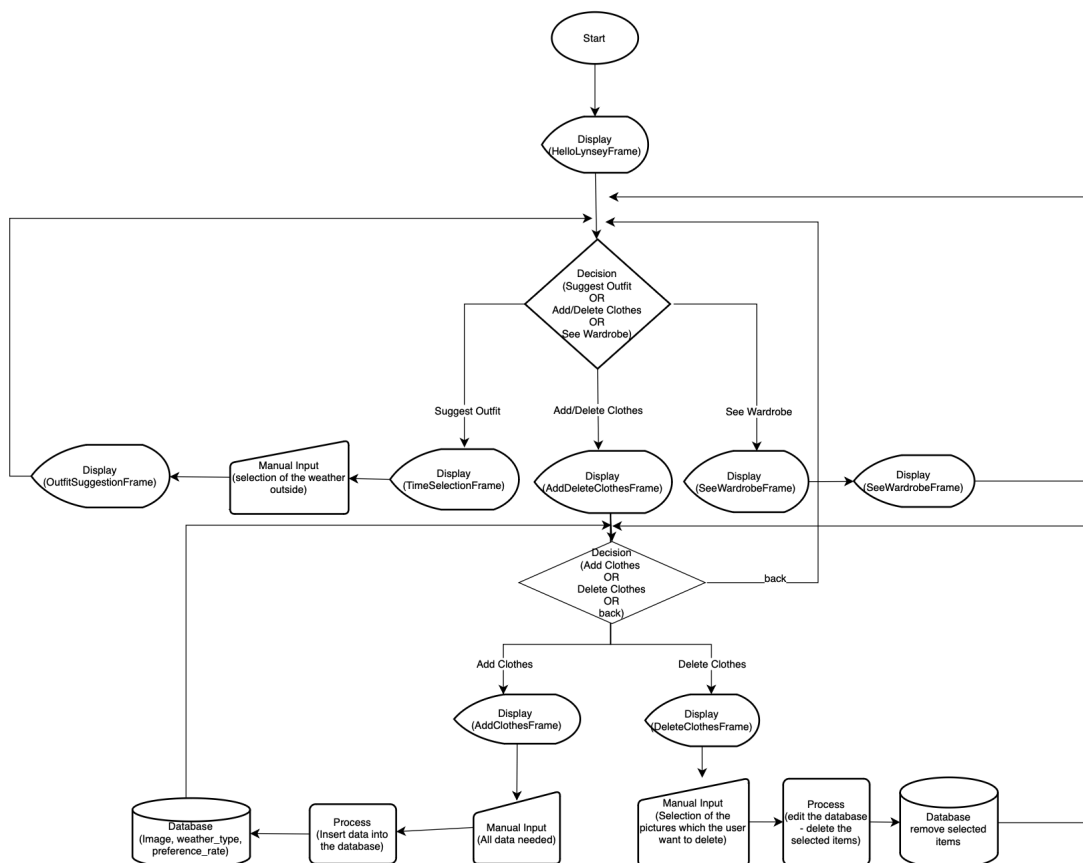
### 3. Output data - table

| Data                                   | Comments  |
|--|---|
| User Interface #1-HelloLynseyFrame     | <ul style="list-style-type: none"> <li>- <u>Edit wardrobe button</u> : redirect her to the AddDeleteFrame(UI#2)</li> <li>- <u>Create outfit button</u>: redirect her to the TimeSelectionFrame(UI#5)</li> </ul>   |
| User Interface #2 -AddDeleteFrame      | <ul style="list-style-type: none"> <li>- <u>Add Clothes button</u>: redirect her to ImageSelectionFrame(UI#3)</li> <li>- <u>Delete Clothes button</u>: redirect her to DeleteClothesFrame(UI#4)</li> </ul>  |
| User Interface #3 -ImageSelectionFrame | <ul style="list-style-type: none"> <li>- <u>Select Image browseButton</u>: lets the user upload a jpg or a jpeg of the clothing item</li> <li>- <u>Weather Type JComboBox</u>: gives the user a choice between the types of weather that the item would be a best fit for (freezing, cold, okay-ish, warm and hot)</li> <li>- <u>Level of preference JTextField</u>: accepts an integer input from 1 to 5 representing her level of preference of each piece (1 = least preferred, 5=most preferred)</li> <li>- <u>Upload button</u>: adds all the data inputted within the level of preference, weather type and select image to the database</li> <li>- <u>Cancel button</u>: redirects the user to the AddDeleteFrame(UI#2)</li> </ul> |
| User Interface #4 - DeleteClothesframe | <ul style="list-style-type: none"> <li>- <u>Display</u>: displays the jpegs and jpegs as buttons and by pressing on the picture it removes it from the database.</li> </ul>   |
| User Interface #5 - SeeWardrobeFrame   | <ul style="list-style-type: none"> <li>- <u>Display</u>: displays all of the jpegs within the database in their order of preference(5 to 1)</li> <li>- <u>Worn button</u>: acts as a toggle, if the button for one item is pressed, its boolean value assigned within the database will be changed to 1 until the button is deselected</li> </ul>   |

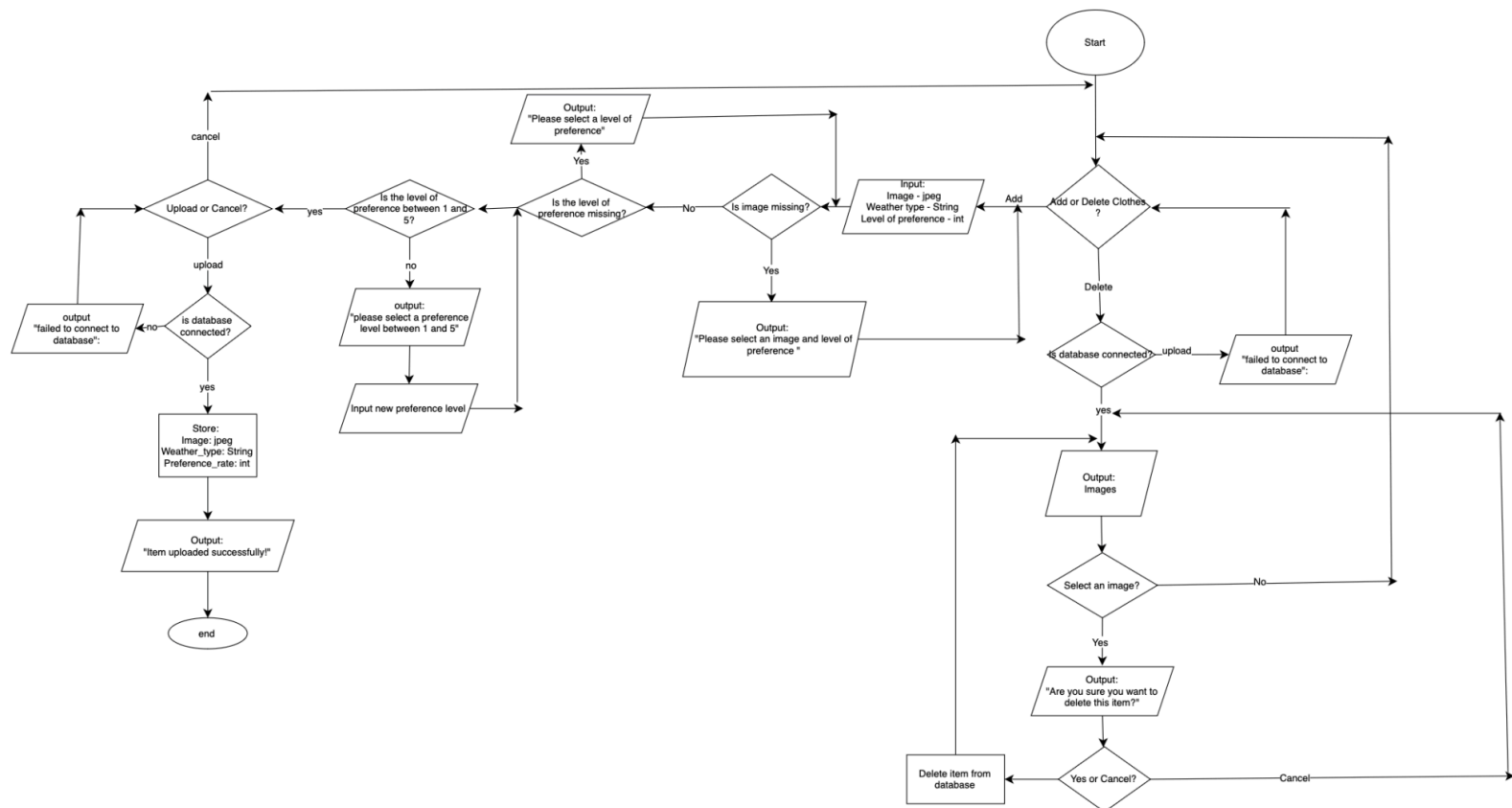
|   |   |
|---|---|
| <b>User Interface #6 - TimeSelectionFrame</b> | <ul style="list-style-type: none"> <li>- <u>Weather buttonGroups</u>: 5 different buttons with a description of the weather (freezing, cold, okay-ish, warm and hot)</li> <li>- <u>Confirm button</u> : redirects the user to the WardrobeFrame(UI#6)</li> <li>- <u>Cancel button</u>: Takes the user to the prior interface</li> </ul> |
| <b>User Interface #7- WardrobeFrame</b>       | <ul style="list-style-type: none"> <li>- <u>Display</u>: this user interface will display the jpegs of the clothes within the specified weather type from the TimeSelectionFrame(UI#5) in the order of their level of preference (5 to 1), placing the worn clothes at the end of the number's category</li> </ul>                      |

#### 4. Process description- Flowcharts

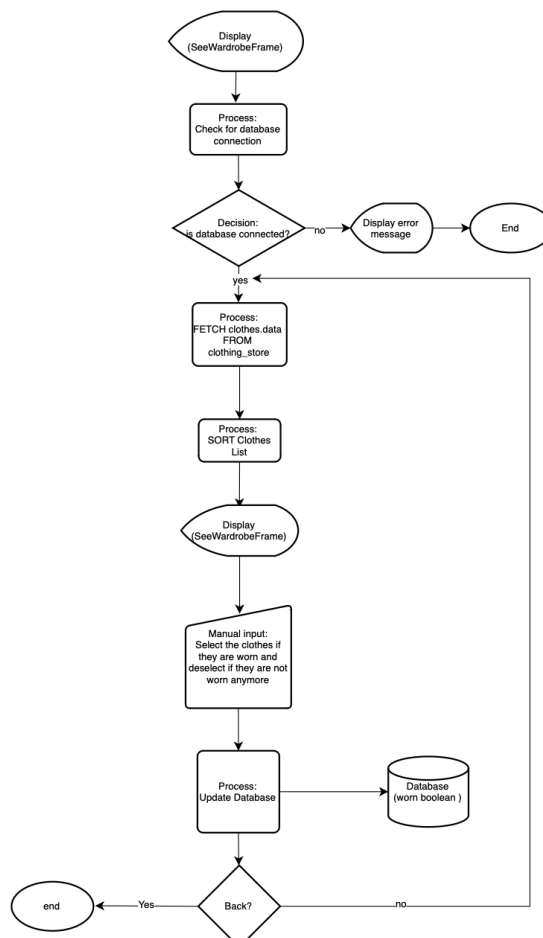
##### a. *Flowchart 1. Main Flowchart of the basic function of the program*



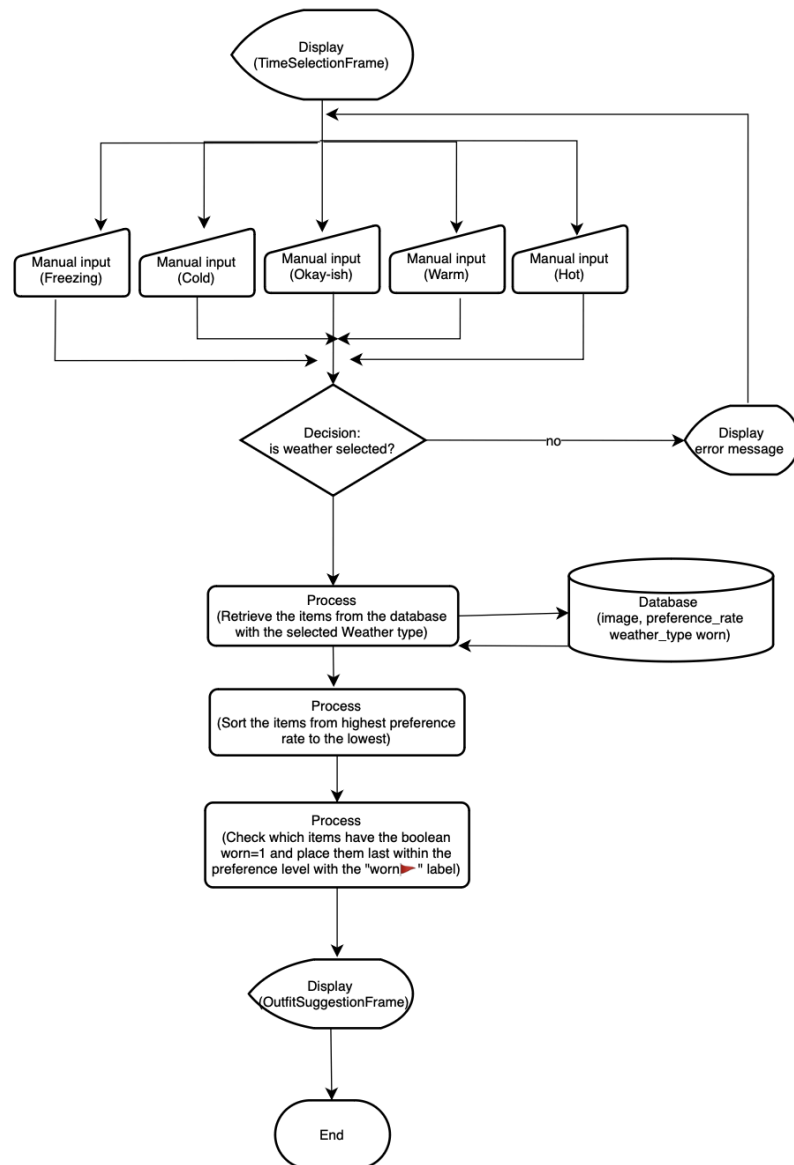
**b. Flowchart 2. Add Delete Clothes functionality for adding and removing items in the**



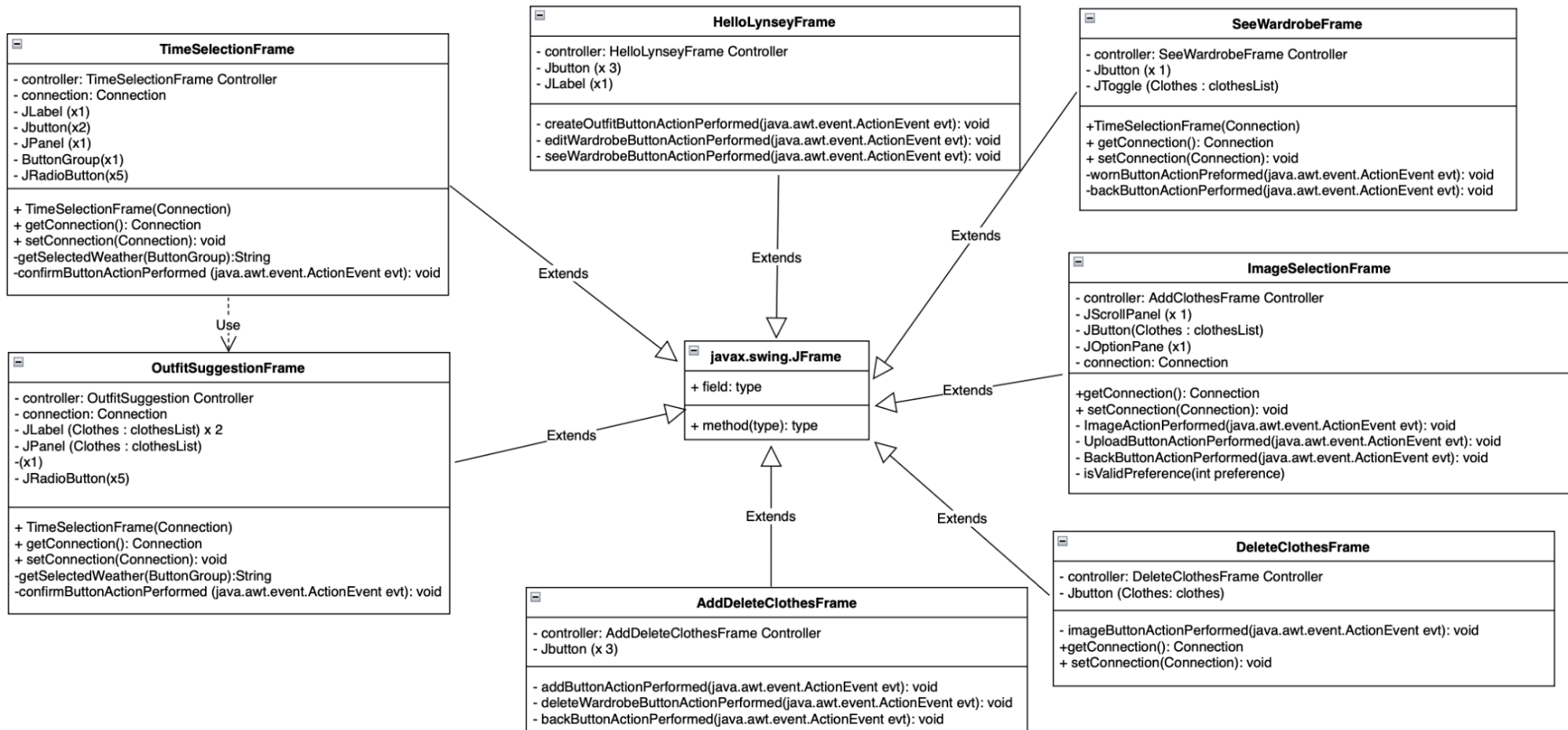
**c. Flowchart 3. See Wardrobe functionality of displaying the items within the database and marking them as being worn or not**



d. *Flowchart 4. Suggest Outfit functionality of selection and sorting of the displayed items from the database*

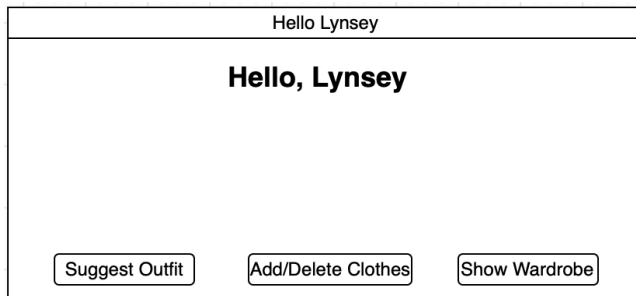


## 5. Process Description - UML Class Diagram



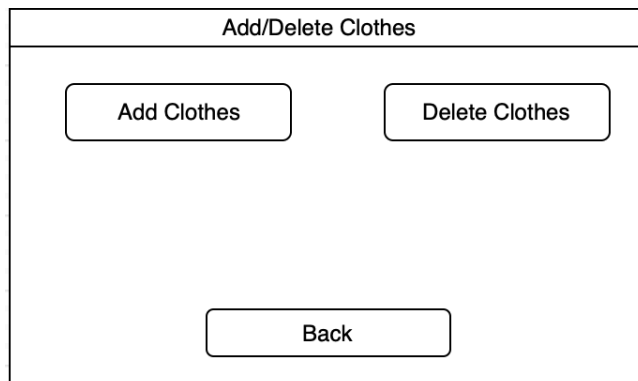
## 6. Software Interface

### UI#1 - HelloLynseyFrame



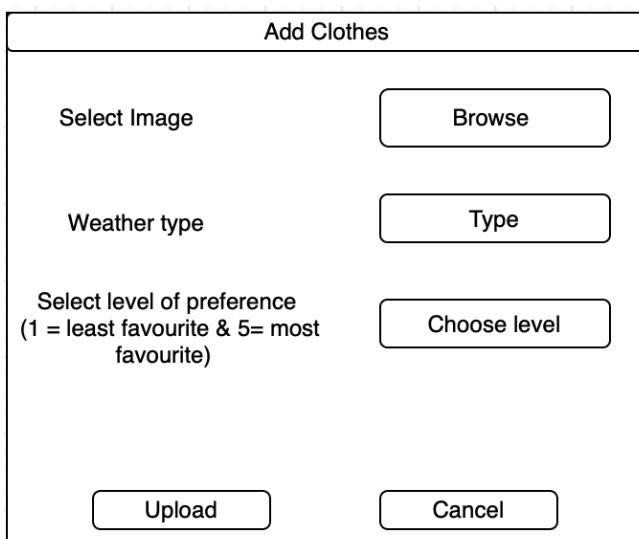
UI#1 - HelloLynseyFrame is a window titled "Hello Lynsey". It displays a large "Hello, Lynsey" message in the center. At the bottom, there are three buttons: "Suggest Outfit", "Add/Delete Clothes", and "Show Wardrobe".

### UI#2 - AddDeleteClothesFrame



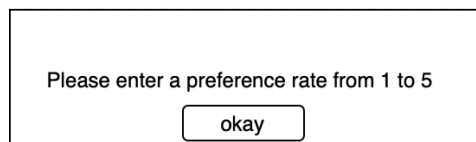
UI#2 - AddDeleteClothesFrame is a window titled "Add/Delete Clothes". It contains two buttons at the top: "Add Clothes" and "Delete Clothes". A "Back" button is located at the bottom center.

### UI#3 - AddClothesFrame



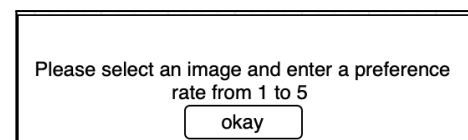
UI#3 - AddClothesFrame is a window titled "Add Clothes". It contains the following elements:

- "Select Image" label with a "Browse" button.
- "Weather type" label with a "Type" button.
- "Select level of preference (1 = least favourite & 5= most favourite)" label with a "Choose level" button.
- "Upload" and "Cancel" buttons at the bottom.



Please enter a preference rate from 1 to 5

okay



Please select an image and enter a preference rate from 1 to 5

okay

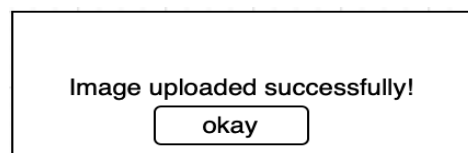


Image uploaded successfully!

okay



#### UI#4 - DeleteClothesFrame

Delete Clothes

jpg

jpg

jpg

jpg

jpg

jpg

Are you sure you want to delete this item?

yes

cancel

#### UI#5 - TimeSelectionFrame

Weather Selection

☐ Hot
☐ Cold

☐ Warm
☐ Freezing

☐ Okay-ish

Confirm

Cancel

Please select a weather option

okay

#### UI#6 - OutfitSuggestionFrame

Outfit Suggestions for [weather]

jpg

jpg

jpg

preference level:

jpg

jpg

jpg

preference level:

preference level:

preference level:

Outfit Suggestions for [weather]

worn▶

jpg

jpg

jpg

preference level:

preference level:

preference level:

jpg

worn▶

jpg

jpg

preference level:

preference level:

preference level:


## UI#7- SeeWardrobeFrame

**See Wardrobe**


☐ worn?

☐ worn?


☐ worn?



jpg



jpg



jpg

preference  
level:

preference  
level:

preference  
level:

## 9. Database Design and Dictionary

### a. Database Design

| id    | image      | weather_type | Preference_rate | Worn<br>(boolean) |
|-------|------------|--------------|-----------------|-------------------|
| (int) | (jpeg/jpg) | (varchar)    | (int)           | (boolean)         |
| (int) | (jpeg/jpg) | (varchar)    | (int)           | (boolean)         |
| (int) | (jpeg/jpg) | (varchar)    | (int)           | (boolean)         |
| (int) | (jpeg/jpg) | (varchar)    | (int)           | (boolean)         |

### b. Database Dictionary

|                 |   |
|-----------------|---|
| id              | Unique identifier of each item which auto-increments                  |
| image           | Binary data storing the image of the clothing item                    |
| weather_type    | Describes the type of the weather suitable for the clothing item      |
| preference_rate | Represents the level of preference between 1 and 5 of each item       |
| worn            | Boolean value representing which item has been worn and which has not |

### **Schedule for developing the product**

The development of the database will be split into three stages: the database development, the Java interface development and connectivity. The database is going to be managing the clothes of the client while the interface will be getting information from the database and outputting it to the client. 1-2 weeks for each stage should be enough time to develop everything.

| Database   | Java Interface   |
|--|--|
| <ul style="list-style-type: none"><li>- Deciding between mySQL and microsoft Access as my database foundations</li><li>- Downloading the software (CodeJava,2023)</li><li>- Creating the fields : image (Input from the user); preference_rate (Input from the User), weather_type (Input from the User) and worn(boolean)</li><li>- Building up the query commands</li><li>- Arranging the filters of order</li></ul> | <ul style="list-style-type: none"><li>- Downloading Eclipse IDE Java for Web Developers (Joey's Tech, 2023)</li><li>- Creating 8 x Dynamic Java Projects (Rai, 2023)</li><li>- Going to the Design section</li><li>- Adding buttons to the interface</li><li>- Adding Action Listeners for each buttons and setting their functions</li><li>- Adding Action Performers for each class</li><li>- Adding connections between the interfaces</li><li>- Adding connections between the interfaces and the database</li><li>- Adding inheritance within the classes</li></ul> |
| Connectivity   |  |
| <ul style="list-style-type: none"><li>- Connecting the 8 java files between them in Eclipse</li><li>- Downloading MySQL Java Connector (ProgrammingKnowledge, 2024)</li><li>- Using Java Connector to connect the Interfaces of the database (TutorialsField, 2023)</li><li>- Testing the product</li></ul>  |  |

### Testing plan of the database

| Action test  | Method of testing and results   |
|--|---|
| Test if the program runs correctly and main window appears on the screen               | Open the java application containing the user interfaces and see whether the home interface appears directly.   |
| Main window buttons work properly  | Checking whether the add button is redirecting to the next user interference frame where the user should be able to add the clothing to the database.   |
| Exit buttons work correctly  | Checking whether the exit button in the home frame closes the frame and whether the exit frame from the other two frames return the user to the home frame .                                  |
| Add/Delete Clothes button redirects the user to the AddDeleteClothesFrame              | The user should be redirected effectively to the AddDeleteClothesFrame to testt its connectivity  |
| Add Clothes button works   | Adding a piece of clothing and then verifying the database and see whether the information has actually been added or not.  |
| Level of preference is correctly added within the database and it is within the limits | The user will be trying to add a piece of clothing where they should be selecting a level of preference. The user should be able to have access to every choice from 1 to 5 for this section. |

|   |  |
|---|--|
| Weather type selection is correctly working | The user should be able to choose the weather type of the day she wants wear the outfit.   |
| Picture display                             | The user should be able to see on the third user interface a series of pictures with clothing according to the weather and level of preference. All of the pictures should be clearly visible and appropriate for the weather.                     |
| Picture order                               | All the pictures that are displayed should be ordered from the most preferred to the least preferred piece of clothing appropriate for the user so that the user could scroll starting with her favourite pieces down to her least favourite ones. |
| Database accuracy                           | The database should be ordered accordingly with the level of accuracy but also be well connected with the weather app so that for each range of degree from the outside the clothes should be accordingly ordered.                                 |

## **Bibliography**

1. *How to install mysql on mac | install mysql on macos (2024) (2024) YouTube*. Available at: <https://www.youtube.com/watch?v=ODA3rWfmzg8&t=338s> (Accessed: 08 April 2024).
2. *Download and install MySQL Workbench on macos (2022) YouTube*. Available at: [https://www.youtube.com/watch?v=sY\\_QPWileDQ](https://www.youtube.com/watch?v=sY_QPWileDQ) (Accessed: 08 April 2024).
3. *How to connect mysql database in Java using Eclipse IDE | Connect | insert | update | delete (2022) YouTube*. Available at: <https://www.youtube.com/watch?v=kiuEl1hRbNA&t=587s> (Accessed: 08 April 2024).
4. *Create first java GUI using Eclipse IDE [2023] | how to install swing in Eclipse| Window Builder (2023) YouTube*. Available at: <https://www.youtube.com/watch?v=Bi48mFHWmX8&t=66s> (Accessed: 08 April 2024).