Criterion B: Design

Table of contents

Table of contents 1

Functionality and Class Diagrams 2

Design Sketch 4

Database 7

Flowcharts 7

Data Structure 10

Searching Algorithms 10

Test Plan 10

Functionality and Class Diagrams

Functionality and Class Diagrams of Each Class:

- Login: To check the username and password and give access to the program

Login
-username: String -password: String -userx: String -passx: String

- Changepass: To change the password

Changepass
-username: String -password: String -userx: String -passx: String -newpass: String -conpass: String (confirm password)

- Stock: The main page where all the stocks are displayed; it is tabbed for the three fields(Hand, Hair and Misc.)

Stock
+face() +hand() +misc()

- Face: To add or subtract stock of face; it displays warning message when stock falls below 10

Face
-alert: String -name: String

-num: int -sat: String (to get inserted number in string form) -stt: int (converts ‘sat’ into int) -final: int
+Fillcombo() +Remarks()

- Hand: To add or subtract stock of hand; it displays warning message when stock falls below 10

-

Hand
-alert: String -name: String -num: int -sat: String (to get inserted number in string form) -stt: int (converts ‘sat’ into int) -final: int
+Fillcombo() +Remarks()

- Misc.: To add or subtract stock of misc.; it displays warning message when stock falls below 10

Misc
-alert: String -name: String -num: int -sat: String (to get inserted number in string form) -stt: int (converts ‘sat’ into int) -final: int
+Fillcombo() +Remarks()

- AddFace: Adds new item in the Face field along with initial stock

AddFace
-itemx: String -stockx: String -stockxx: int (converts ‘stockx’ into int)

- AddHand: Adds new item in the Hand field along with initial stock

AddHand
-itemx: String -stockx: String -stockxx: int (converts ‘stockx’ into int)

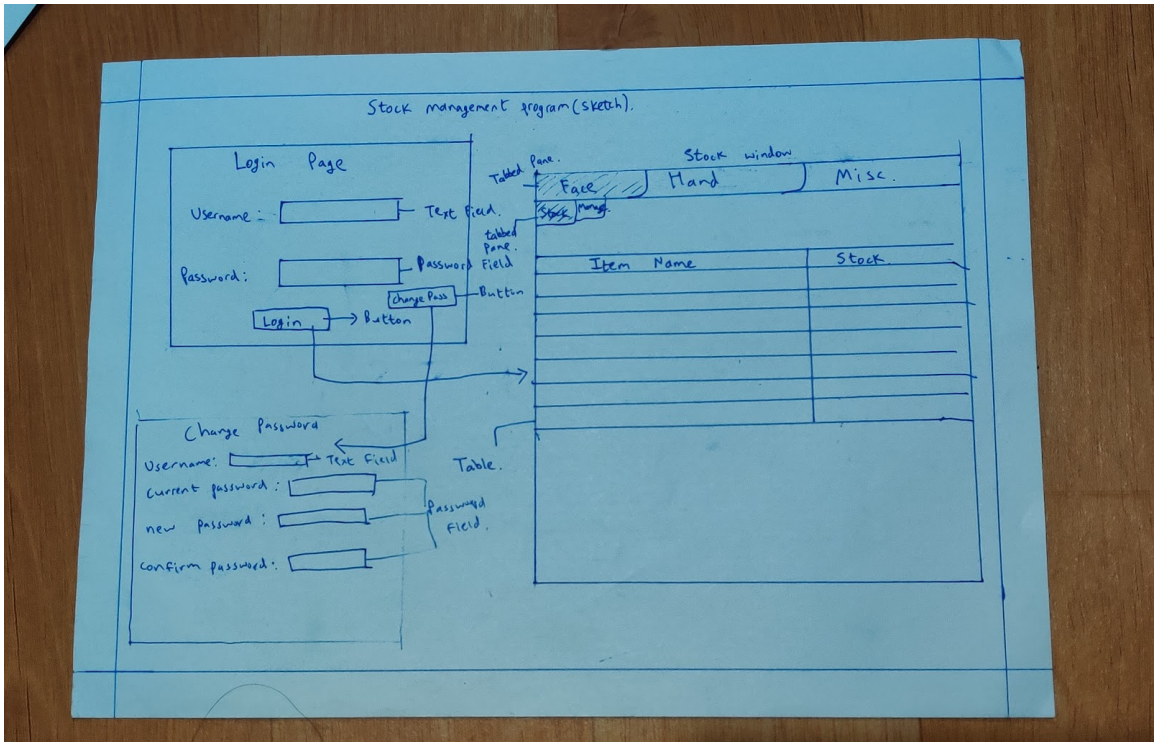
- AddMisc.: Adds new item in the Misc. field along with initial stock

AddMisc
-itemx: String -stockx: String -stockxx: int (converts ‘stockx’ into int)

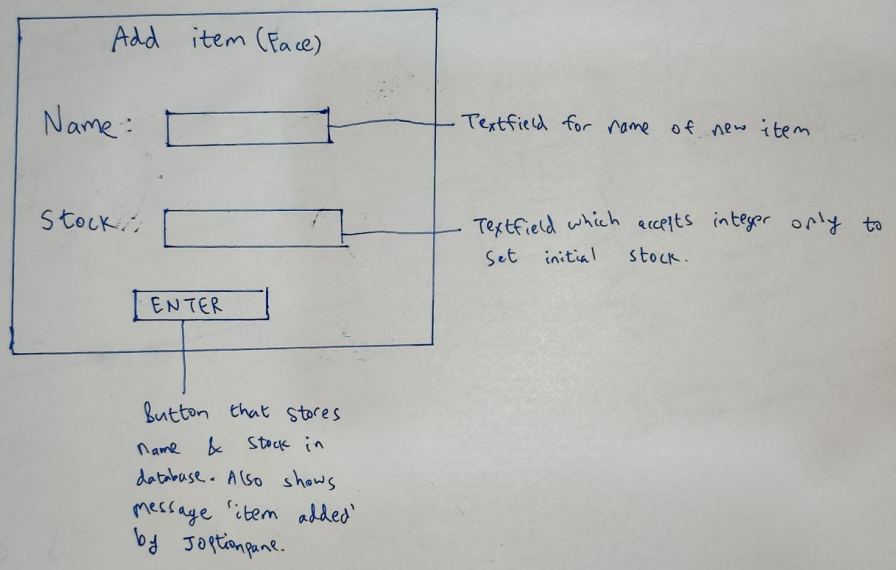
Design Sketch

The images below are a rough sketch of how the UI of my program will look like.

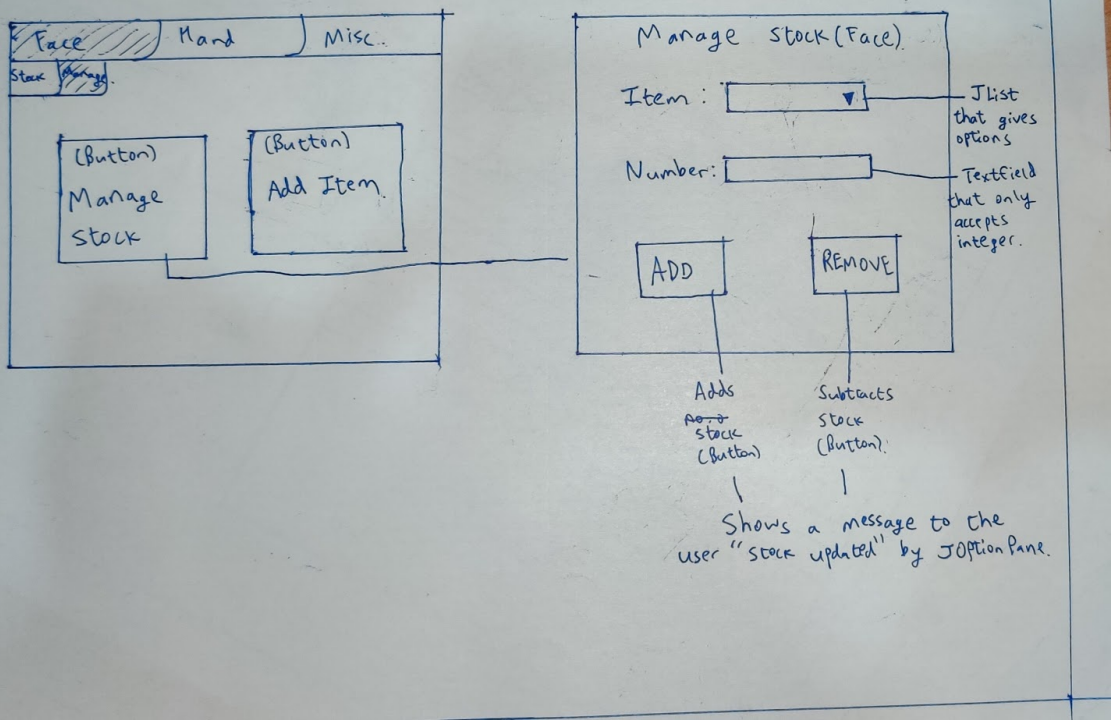
Sketch of the UI



Stock Management Program (Sketch)



Stock Management program (sketch).



Database

Derby database, an in-built database of Netbeans was used in my product. Since I only needed 2 fields in each table, the tables are in 3NF form.
Database for the login details titled as ‘storelog’ has the table LOGIN with:

Username	Password

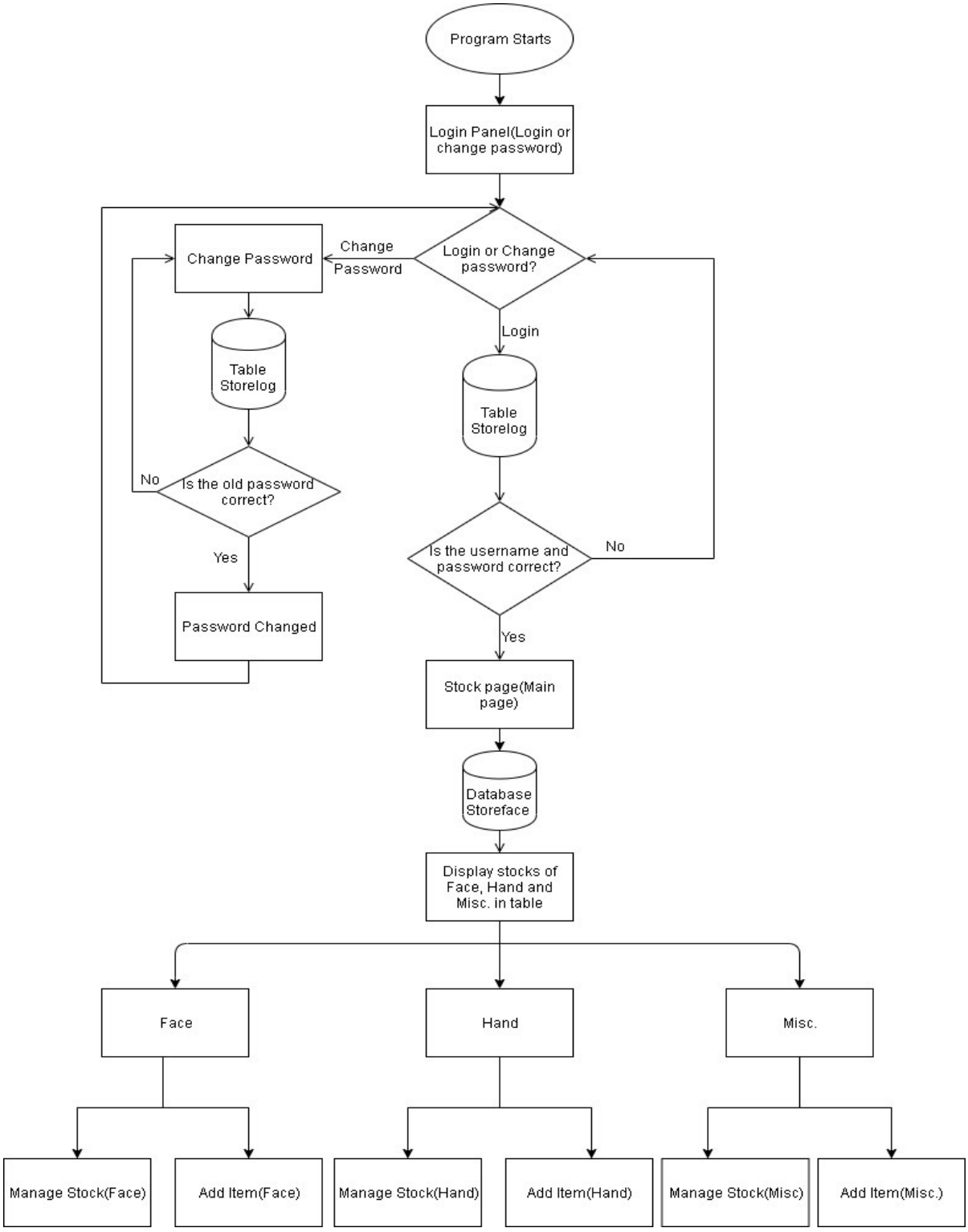
Database for the items and the stock level titled as ‘storeface’ has 3 tables FACE, HAND and MISC with:

Item	Stock

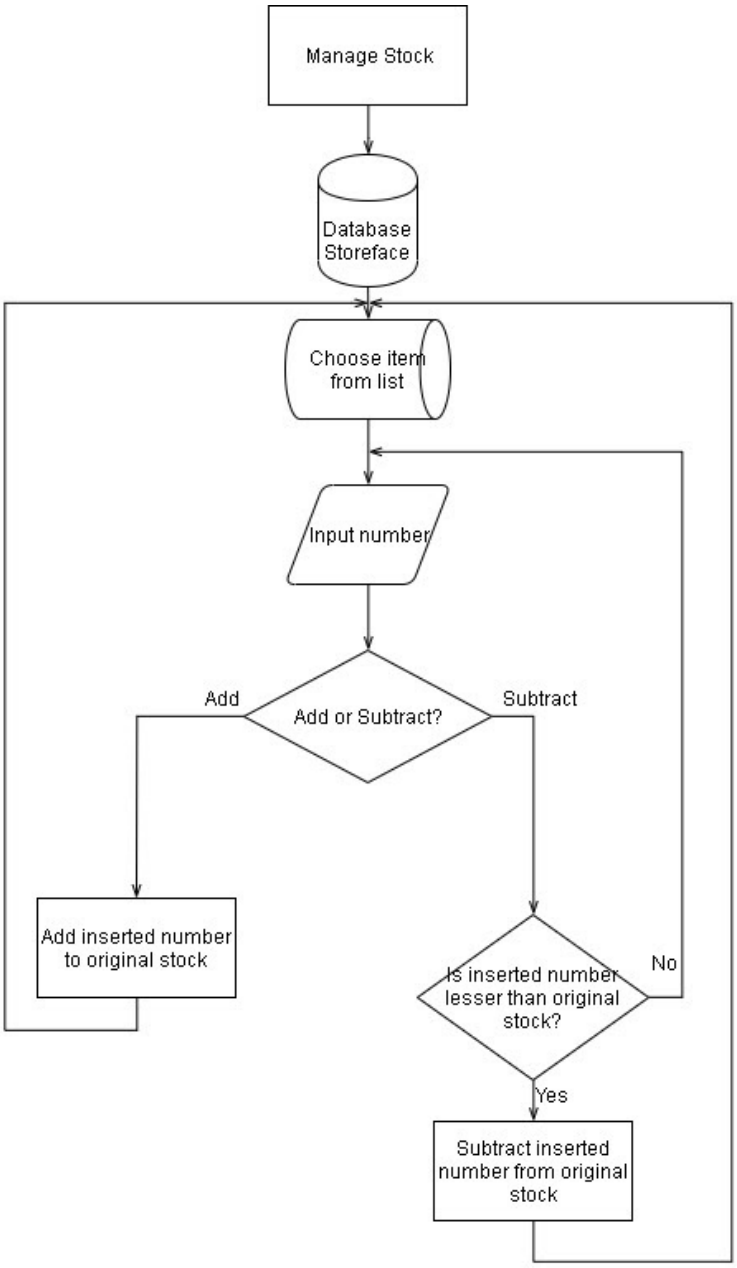
Flowcharts

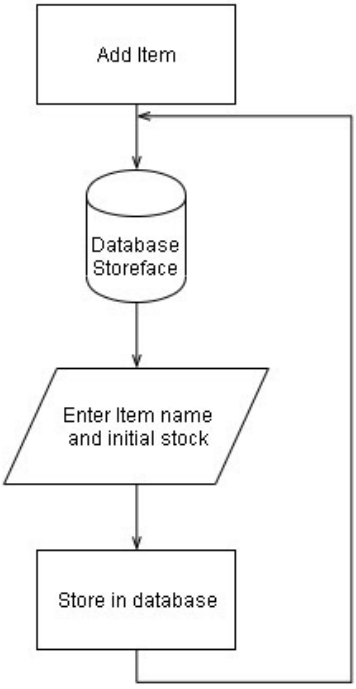
The flowcharts given below show the flow of the program once it is executed.

1.Flowchart of the whole program



All the Manage Stock and Add Item has the following flowcharts:


Manage Stock
Database Storeface
Choose item from list
Input number
Add or Subtract?
Add
Subtract
Add inserted number to original stock
Is inserted number lesser than original stock?
No
Yes
Subtract inserted number from original stock


Add Item
Database Storeface
Enter Item name and initial stock
Store in database

Data Structure

Array: For the storage of login details, Array will be used as there is only 1 user for the program so a static data structure would be appropriate.

Arraylist: For the stock management, Arraylist will be for all the three fields(Face, Hand and Misc.). Arraylist will be used as it is a dynamic data structure which fits to the purpose of the program, because the user should be able to add new items within the execution.

Searching Algorithms

When updating the stock for the selected item, the program will be going through the database with **Linear Search** to detect the item name required for the stock update. Linear Search will be used as there is less number of items and also, the items may not be sorted in order.

Test Plan

Action	Method
Unit Testing	
Different buttons and frames	Different buttons will be pressed to check if they work properly or not, and frames will be ran separately
Database Testing	
To check if the table shows the items in the database	The main ‘stock’ page will be checked if it shows all 3 categories
Checking if database is updated by running the program	By adding or subtracting stocks in the program and checking if the changes have

	been saved in the database.
Functionality Testing	
The login page will be tested	By entering the username and password, it will be checked if the program is only accessible by authenticated users only
UI Components Testing	
Location and accessibility	The program's accessibility and identification of components will be tested to see if the program is easy to use
Integration Testing	
Full testing	The program will be checked as a whole, from the login to the stock update and adding of items to check the transition between frames, proper functioning of the program, etc.
Data Testing	
Normal, abnormal and extreme data	Normal, abnormal and extreme data will be entered to check error handling.