GROUP ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT038-3-2-OODJ

Object Oriented Development with Java (OODJ)

APU2F2109CS(DA)

HAND OUT DATE: 20 SEPTEMBER 2021

HAND IN DATE: 06 DECEMBER 2021

WEIGHTAGE: 25%

GROUP MEMBERS:

1 **Kevin Setiawan Miharjo – TP059076**

2 **Marcell Agung Wahyudi – TP058650**

Table of Contents

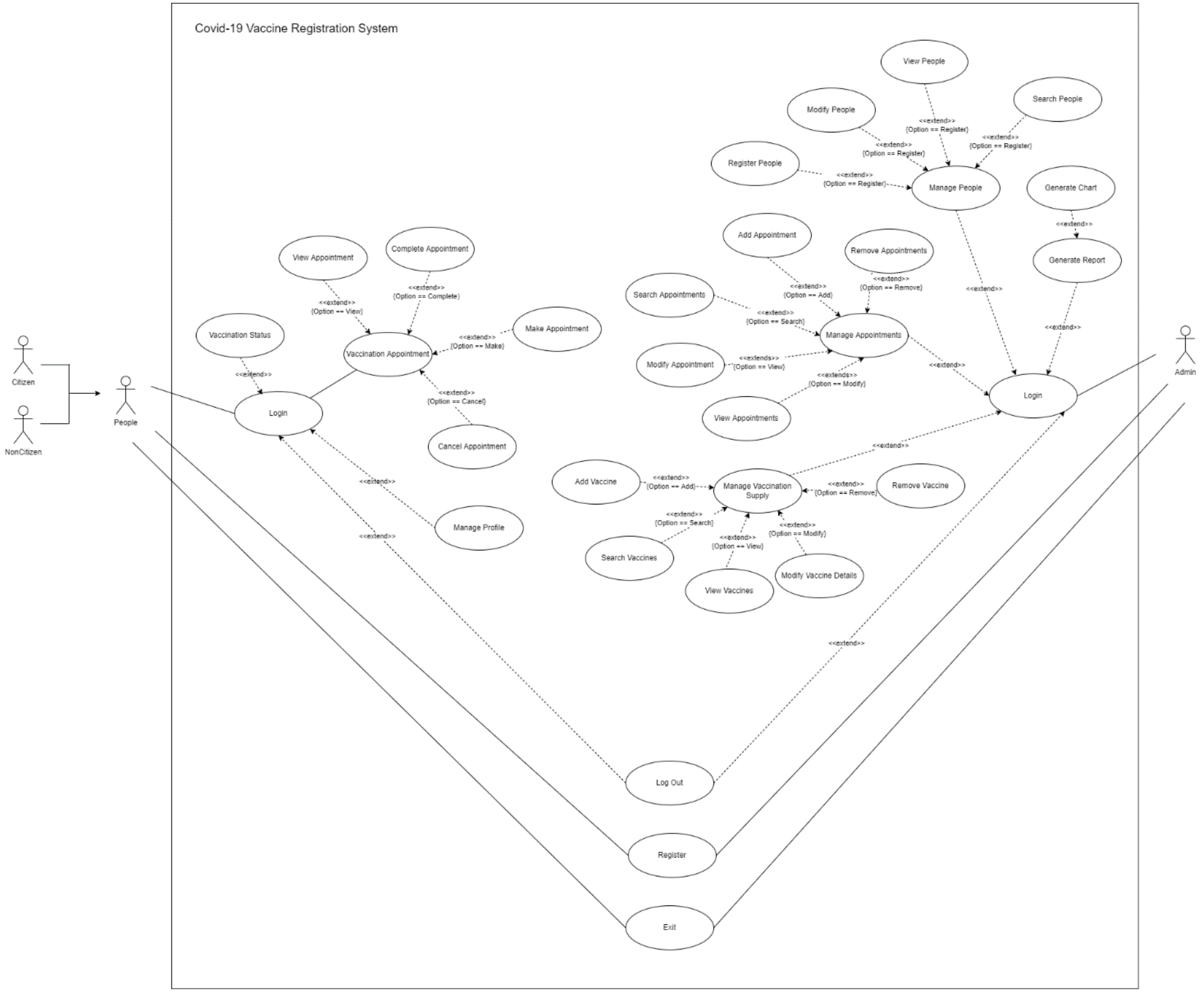
1.0 Assumptions 3
2.0 Diagrams 4
2.1 Use Case Diagram 4
..... 4
2.1.1 Specification Table 5
2.2 Class Diagram 8
3.0 System Documentation 9
3.1 User Manual 9
3.1.1 People 12
3.1.2 Admin 19
3.2 OO Concepts 33
3.2.1 Abstraction 33
3.2.2 Encapsulation 34
3.2.3 Modularity 35
3.2.4 Inheritance 36
3.3 Additional Features 37
3.3.1 JPasswordField 37
3.3.2 JTable 38
3.3.3 TableRowSorter 39
3.3.4 JComboBox 40
3.3.5 BarChart 42
4.0 References 45

1.0 Assumptions

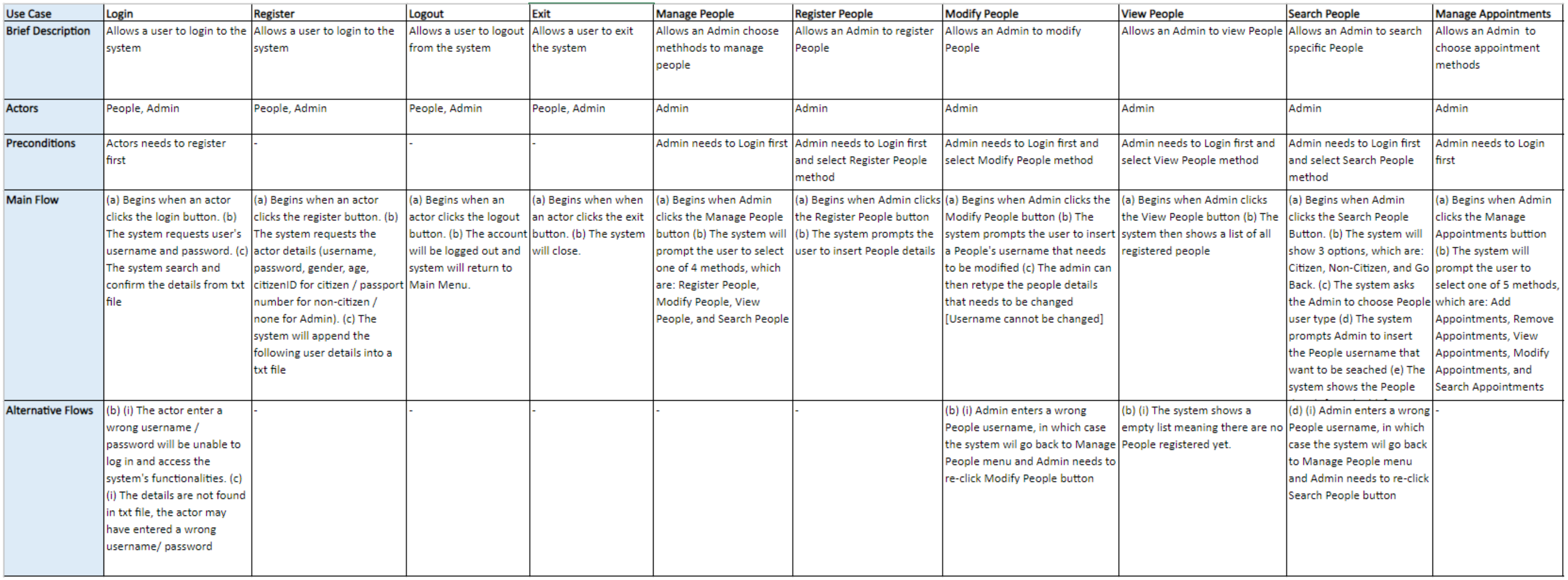
- User can choose their access rights such as Citizen / Non-Citizens
- To register an admin account, special password is required
- People who registered for the system has not been vaccinated.
- People should be able to register and update their profile.
- There should be no duplicate username for each type of access rights (Citizen / Non-Citizens).
- Vaccination Status can only be generated after being vaccinated.
- For each centre, admin can modify the appointment and vaccine supply.
- For each centre, there can be no duplicate appointment (same day & time).
- For each centre, there can be no duplicate vaccine.
- User can only make appointment in centre which had vaccine and appointment timetable available.
- People can complete and cancel their appointment through the system.

2.0 Diagrams

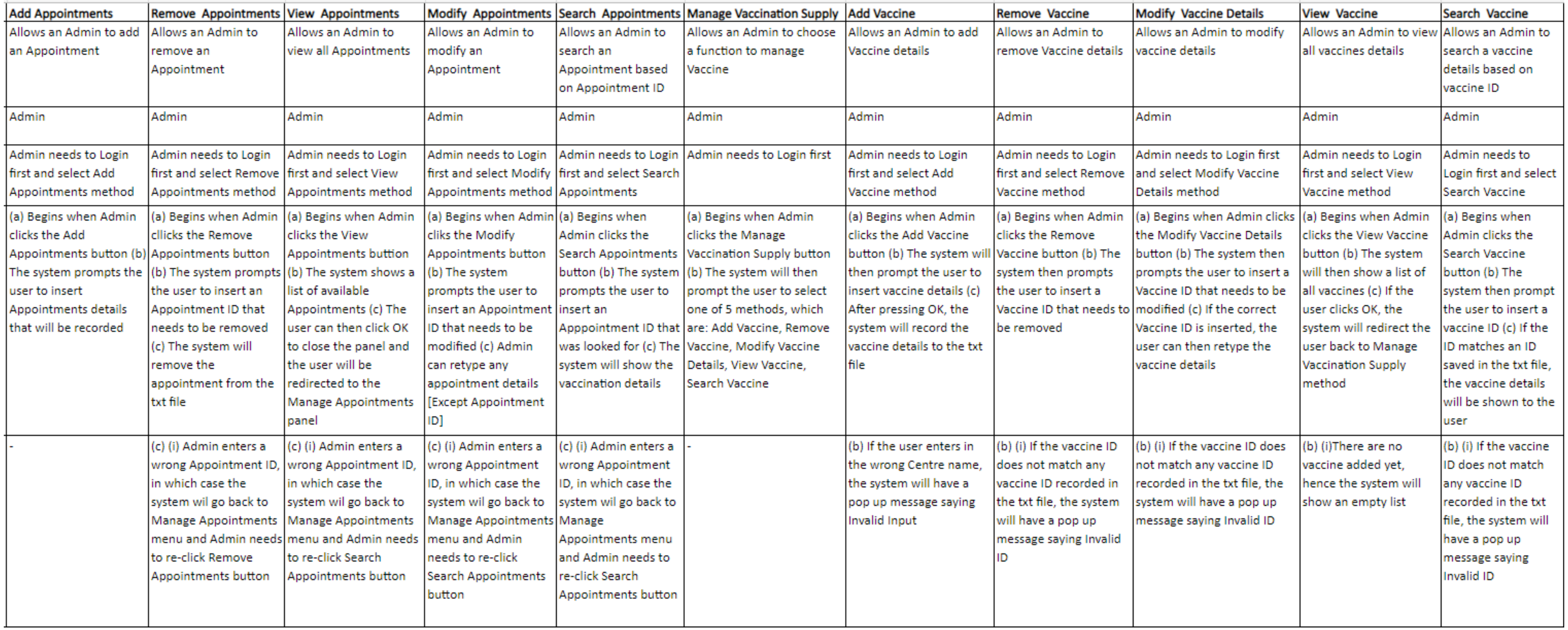
2.1 Use Case Diagram


Covid-19 Vaccine Registration System
Citizen
NonCitizen
People
Admin
View People
Modify People
Search People
Register People
Manage People
Generate Chart
Generate Report
<<extend>> {Option == Register}
<<extend>>
Add Appointment
Remove Appointments
Search Appointments
Manage Appointments
Modify Appointment
View Appointments
<<extend>> {Option == Add}
<<extend>> {Option == Remove}
<<extend>> {Option == Search}
<<extends>> {Option == View}
<<extends>> {Option == Modify}
Login
View Appointment
Complete Appointment
Vaccination Status
Vaccination Appointment
Make Appointment
Cancel Appointment
Manage Profile
<<extend>> {Option == View}
<<extend>> {Option == Complete}
<<extend>> {Option == Make}
<<extend>> {Option == Cancel}
Add Vaccine
Manage Vaccination Supply
Remove Vaccine
Search Vaccines
View Vaccines
Modify Vaccine Details
<<extend>> {Option == Add}
<<extend>> {Option == Remove}
<<extend>> {Option == Search}
<<extend>> {Option == View}
<<extend>> {Option == Modify}
Log Out
Register
Exit

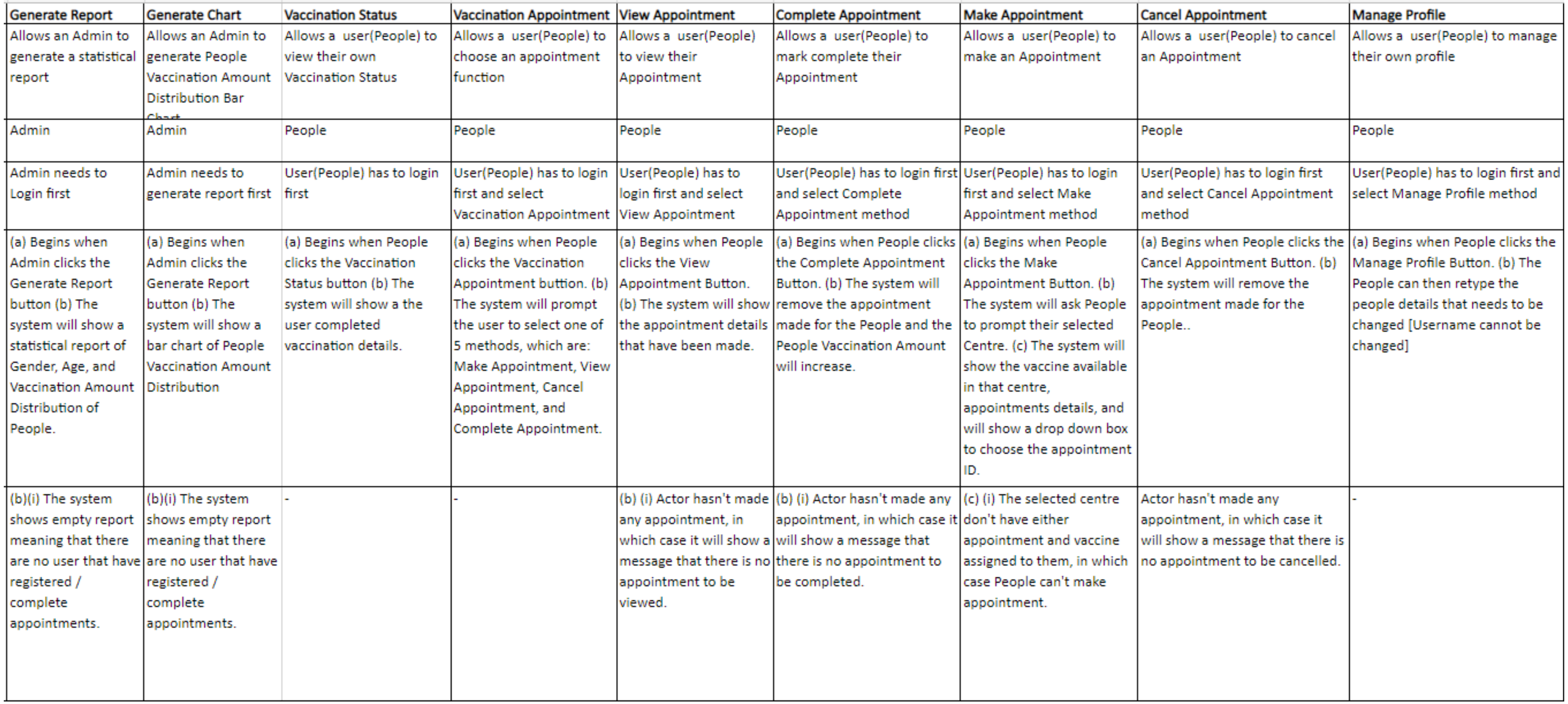
2.1.1 Specification Table



Use Case	Login	Register	Logout	Exit	Manage People	Register People	Modify People	View People	Search People	Manage Appointments
Brief Description	Allows a user to login to the system	Allows a user to login to the system	Allows a user to logout from the system	Allows a user to exit the system	Allows an Admin choose methhods to manage people	Allows an Admin to register People	Allows an Admin to modify People	Allows an Admin to view People	Allows an Admin to search specific People	Allows an Admin to choose appointment methods
Actors	People, Admin	People, Admin	People, Admin	People, Admin	Admin	Admin	Admin	Admin	Admin	Admin
Preconditions	Actors needs to register first	-	-	-	Admin needs to Login first	Admin needs to Login first and select Register People method	Admin needs to Login first and select Modify People method	Admin needs to Login first and select View People method	Admin needs to Login first and select Search People method	Admin needs to Login first
Main Flow	(a) Begins when an actor clicks the login button. (b) The system requests user's username and password. (c) The system search and confirm the details from txt file	(a) Begins when an actor clicks the register button. (b) The system requests the actor details (username, password, gender, age, citizenID for citizen / passport number for non-citizen / none for Admin). (c) The system will append the following user details into a txt file	(a) Begins when an actor clicks the logout button. (b) The account will be logged out and system will return to Main Menu.	(a) Begins when when an actor clicks the exit button. (b) The system will close.	(a) Begins when Admin clicks the Manage People button (b) The system will prompt the user to select one of 4 methods, which are: Register People, Modify People, View People, and Search People	(a) Begins when Admin clicks the Register People button (b) The system prompts the user to insert People details	(a) Begins when Admin clicks the Modify People button (b) The system prompts the user to insert a People's username that needs to be modified (c) The admin can then retype the people details that needs to be changed [Username cannot be changed]	(a) Begins when Admin clicks the View People button (b) The system then shows a list of all registered people	(a) Begins when Admin clicks the Search People Button. (b) The system will show 3 options, which are: Citizen, Non-Citizen, and Go Back. (c) The system asks the Admin to choose People user type (d) The system prompts Admin to insert the People username that want to be seached (e) The system shows the People	(a) Begins when Admin clicks the Manage Appointments buttion (b) The system will prompt the user to select one of 5 methods, which are: Add Appointments, Remove Appointments, View Appointments, Modify Appointments, and Search Appointments
Alternative Flows	(b) (i) The actor enter a wrong username / password will be unable to log in and access the system's functionalities. (c) (i) The details are not found in txt file, the actor may have entered a wrong username/ password	-	-	-	-	-	(b) (i) Admin enters a wrong People username, in which case the system wil go back to Manage People menu and Admin needs to re-click Modify People button	(b) (i) The system shows a empty list meaning there are no People registered yet.	(d) (i) Admin enters a wrong People username, in which case the system wil go back to Manage People menu and Admin needs to re-click Search People button	-

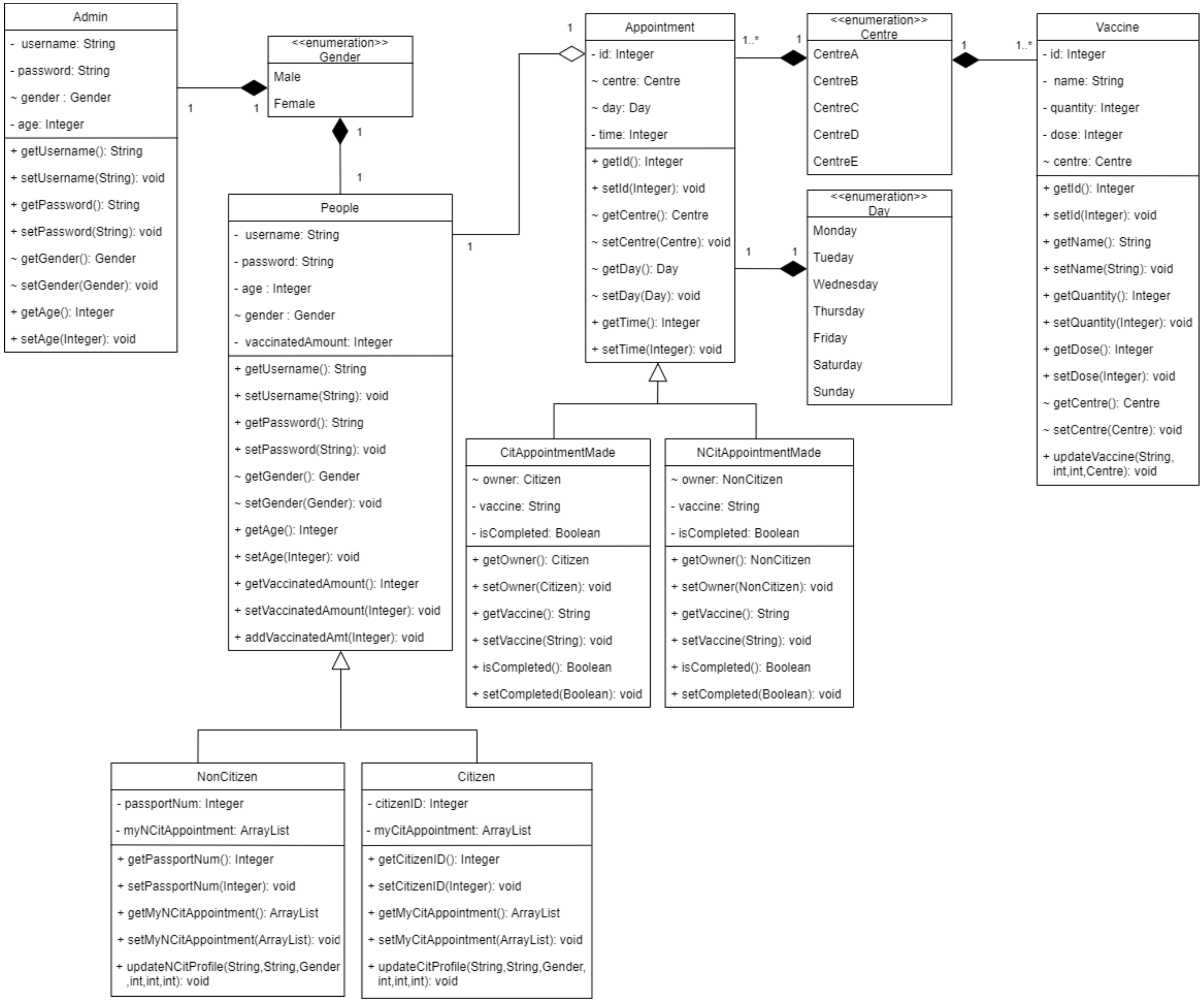


Add Appointments	Remove Appointments	View Appointments	Modify Appointments	Search Appointments	Manage Vaccination Supply	Add Vaccine	Remove Vaccine	Modify Vaccine Details	View Vaccine	Search Vaccine
Allows an Admin to add an Appointment	Allows an Admin to remove an Appointment	Allows an Admin to view all Appointments	Allows an Admin to modify an Appointment	Allows an Admin to search an Appointment based on Appointment ID	Allows an Admin to choose a function to manage Vaccine	Allows an Admin to add Vaccine details	Allows an Admin to remove Vaccine details	Allows an Admin to modify vaccine details	Allows an Admin to view all vaccines details	Allows an Admin to search a vaccine details based on vaccine ID
Admin	Admin	Admin	Admin	Admin	Admin	Admin	Admin	Admin	Admin	Admin
Admin needs to Login first and select Add Appointments method	Admin needs to Login first and select Remove Appointments method	Admin needs to Login first and select View Appointments method	Admin needs to Login first and select Modify Appointments method	Admin needs to Login first and select Search Appointments	Admin needs to Login first	Admin needs to Login first and select Add Vaccine method	Admin needs to Login first and select Remove Vaccine method	Admin needs to Login first and select Modify Vaccine Details method	Admin needs to Login first and select View Vaccine method	Admin needs to Login first and select Search Vaccine
(a) Begins when Admin clicks the Add Appointments button (b) The system prompts the user to insert Appointments details that will be recorded	(a) Begins when Admin cllicks the Remove Appointments button (b) The system prompts the user to insert an Appointment ID that needs to be removed (c) The system will remove the appointment from the txt file	(a) Begins when Admin clicks the View Appointments buttion (b) The system shows a list of available Appointments (c) The user can then click OK to close the panel and the user will be redirected to the Manage Appointments panel	(a) Begins when Admin cliks the Modify Appointments button (b) The system prompts the user to insert an Appointment ID that needs to be modified (c) Admin can retype any appointment details [Except Appointment ID]	(a) Begins when Admin clicks the Search Appointments button (b) The system prompts the user to insert an Apppointment ID that was looked for (c) The system will show the vaccination details	(a) Begins when Admin clicks the Manage Vaccination Supply button (b) The system will then prompt the user to select one of 5 methods, which are: Add Vaccine, Remove Vaccine, Modify Vaccine Details, View Vaccine, Search Vaccine	(a) Begins when Admin clicks the Add Vaccine button (b) The system will then prompt the user to insert vaccine details (c) After pressing OK, the system will record the vaccine details to the txt file	(a) Begins when Admin clicks the Remove Vaccine button (b) The system then prompts the user to insert a Vaccine ID that needs to be removed	(a) Begins when Admin clicks the Modify Vaccine Details button (b) The system then prompts the user to insert a Vaccine ID that needs to be modified (c) If the correct Vaccine ID is inserted, the user can then retype the vaccine details	(a) Begins when Admin clicks the View Vaccine button (b) The system will then show a list of all vaccines (c) If the user clicks OK, the system will redirect the user back to Manage Vaccination Supply method	(a) Begins when Admin clicks the Search Vaccine button (b) The system then prompt the user to insert a vaccine ID (c) If the ID matches an ID saved in the txt file, the vaccine details will be shown to the user
-	(c) (i) Admin enters a wrong Appointment ID, in which case the system wil go back to Manage Appointments menu and Admin needs to re-click Remove Appointments button	(c) (i) Admin enters a wrong Appointment ID, in which case the system wil go back to Manage Appointments menu and Admin needs to re-click Search Appointments button	(c) (i) Admin enters a wrong Appointment ID, in which case the system wil go back to Manage Appointments menu and Admin needs to re-click Search Appointments button	(c) (i) Admin enters a wrong Appointment ID, in which case the system wil go back to Manage Appointments menu and Admin needs to re-click Search Appointments button	-	(b) If the user enters in the wrong Centre name, the system will have a pop up message saying Invalid Input	(b) (i) If the vaccine ID does not match any vaccine ID recorded in the txt file, the system will have a pop up message saying Invalid ID	(b) (i) If the vaccine ID does not match any vaccine ID recorded in the txt file, the system will have a pop up message saying Invalid ID	(b) (i)There are no vaccine added yet, hence the system will show an empty list	(b) (i) If the vaccine ID does not match any vaccine ID recorded in the txt file, the system will have a pop up message saying Invalid ID



Generate Report	Generate Chart	Vaccination Status	Vaccination Appointment	View Appointment	Complete Appointment	Make Appointment	Cancel Appointment	Manage Profile
Allows an Admin to generate a statistical report	Allows an Admin to generate People Vaccination Amount Distribution Bar Chart	Allows a user(People) to view their own Vaccination Status	Allows a user(People) to choose an appointment function	Allows a user(People) to view their Appointment	Allows a user(People) to mark complete their Appointment	Allows a user(People) to make an Appointment	Allows a user(People) to cancel an Appointment	Allows a user(People) to manage their own profile
Admin	Admin	People	People	People	People	People	People	People
Admin needs to Login first	Admin needs to generate report first	User(People) has to login first	User(People) has to login first and select Vaccination Appointment	User(People) has to login first and select View Appointment	User(People) has to login first and select Complete Appointment method	User(People) has to login first and select Make Appointment method	User(People) has to login first and select Cancel Appointment method	User(People) has to login first and select Manage Profile method
(a) Begins when Admin clicks the Generate Report button (b) The system will show a statistical report of Gender, Age, and Vaccination Amount Distribution of People.	(a) Begins when Admin clicks the Generate Report button (b) The system will show a bar chart of People Vaccination Amount Distribution	(a) Begins when People clicks the Vaccination Status button (b) The system will show a the user completed vaccination details.	(a) Begins when People clicks the Vaccination Appointment buttion. (b) The system will prompt the user to select one of 5 methods, which are: Make Appointment, View Appointment, Cancel Appointment, and Complete Appointment.	(a) Begins when People clicks the View Appointment Button. (b) The system will show the appointment details that have been made.	(a) Begins when People clicks the Complete Appointment Button. (b) The system will remove the appointment made for the People and the People Vaccination Amount will increase.	(a) Begins when People clicks the Make Appointment Button. (b) The system will ask People to prompt their selected Centre. (c) The system will show the vaccine available in that centre, appointments details, and will show a drop down box to choose the appointment ID.	(a) Begins when People clicks the Cancel Appointment Button. (b) The system will remove the appointment made for the People..	(a) Begins when People clicks the Manage Profile Button. (b) The People can then retype the people details that needs to be changed [Username cannot be changed]
(b)(i) The system shows empty report meaning that there are no user that have registered / complete appointments.	(b)(i) The system shows empty report meaning that there are no user that have registered / complete appointments.	-	-	(b) (i) Actor hasn't made any appointment, in which case it will show a message that there is no appointment to be viewed.	(b) (i) Actor hasn't made any appointment, in which case it will show a message that there is no appointment to be completed.	(c) (i) The selected centre don't have either appointment and vaccine assigned to them, in which case People can't make appointment.	Actor hasn't made any appointment, in which case it will show a message that there is no appointment to be cancelled.	-

2.2 Class Diagram



3.0 System Documentation

3.1 User Manual

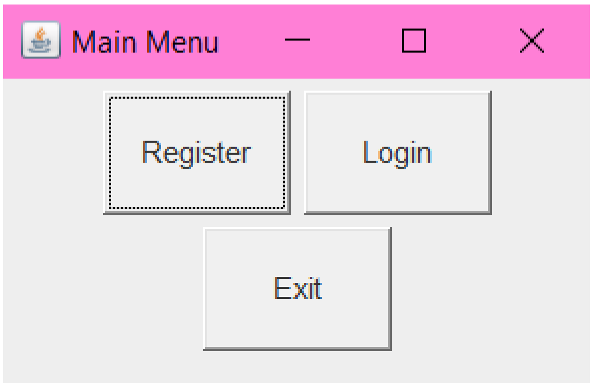


Figure 1 Main Menu

Upon entering the system, 3 button will be available for user to choose. To be able to access the system functionalities, user needs to register an account first.

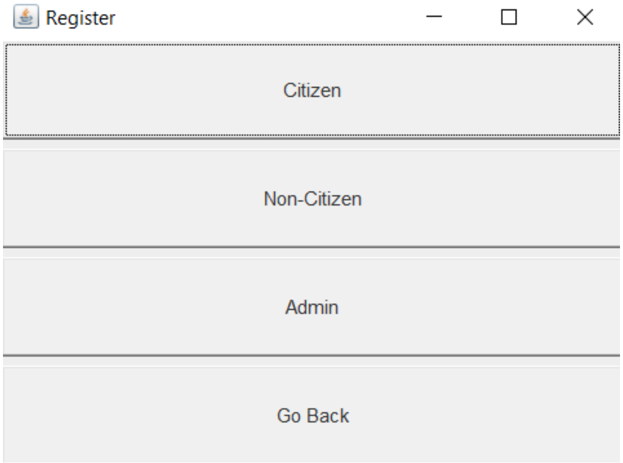
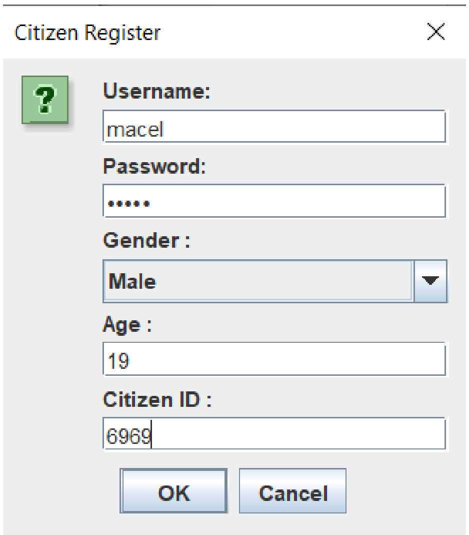


Figure 2 Register

User can register as three accounts, which are Citizen, Non-Citizen and Admin.



Citizen Register

Username:
macel

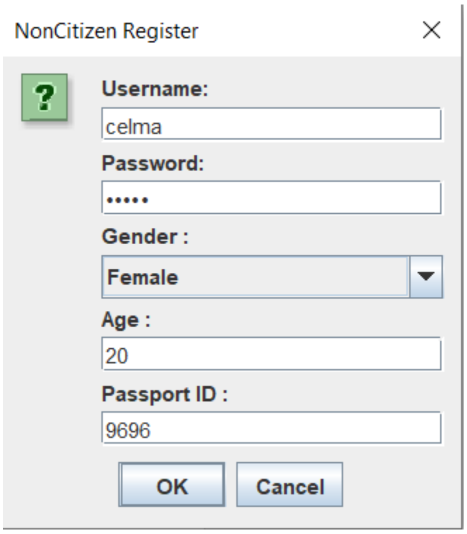
Password:
•••••

Gender :
Male

Age :
19

Citizen ID :
6969

Figure 3 Citizen Register



NonCitizen Register

Username:
celma

Password:
•••••

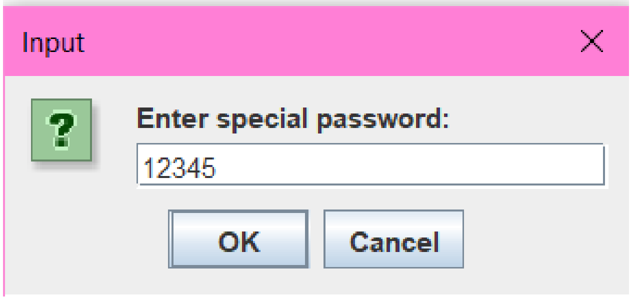
Gender :
Female

Age :
20

Passport ID :
9696

Figure 4 NonCitizen Register

Citizen and Non-Citizen needs to input username, password, gender, age. For citizen they will need their citizen ID and Non-citizen will need Passport Number. For admin, the user needs to enter in a specific special password. This is in order to validate and prevent malicious actions.

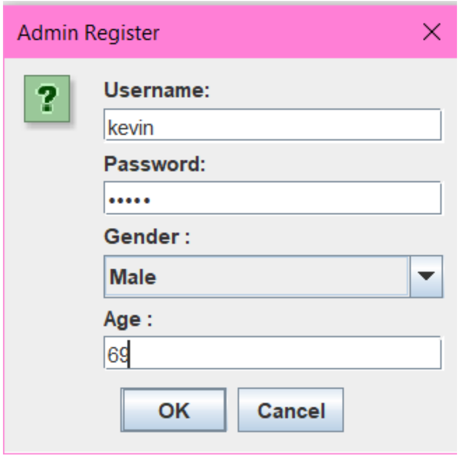


Input

Enter special password:
12345

Figure 5 Admin Special Password

If the inputted special password is correct, then the user can proceed to register as Admin.



Admin Register

Username: kevin

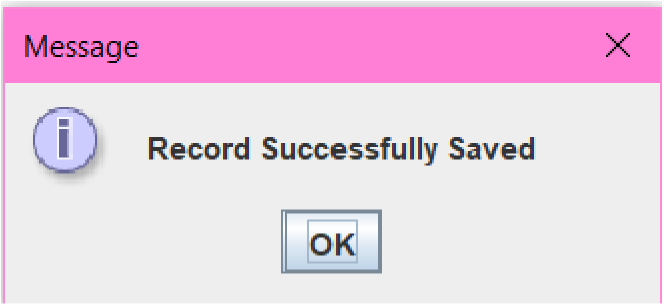
Password: •••••

Gender : Male

Age : 69

Figure 6 Admin Register

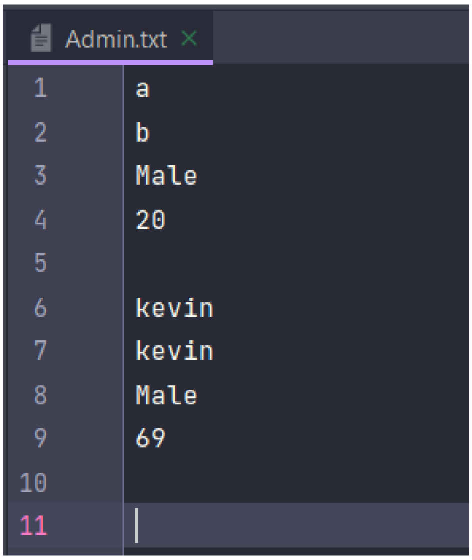
Upon registering as any account, the user will receive a pop-up message and the record will be automatically saved into the respective txt file (Citizen.txt, NonCitizen.txt, Admin.txt).



Message

Record Successfully Saved

Figure 7 Register succeeds pop-up message

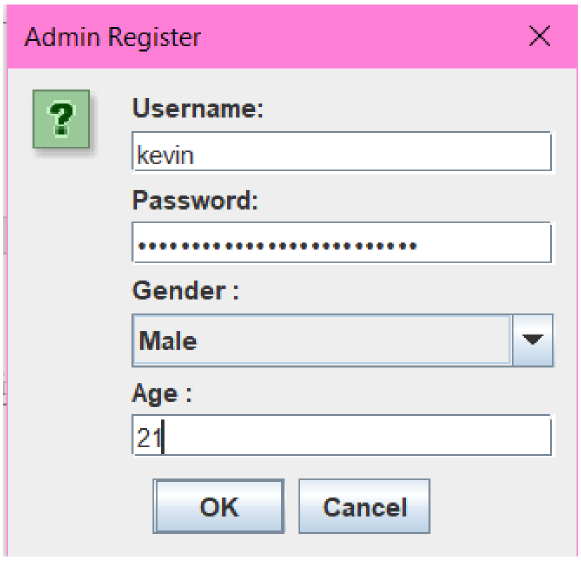


Admin.txt

```
a
b
Male
20

kevin
kevin
Male
69


```

Figure 8 Record saved in txt file

Incidentally, if the username in each user type (Citizen/Non-Citizen/Admin) has already been used by another user, the system will not accept the registration.



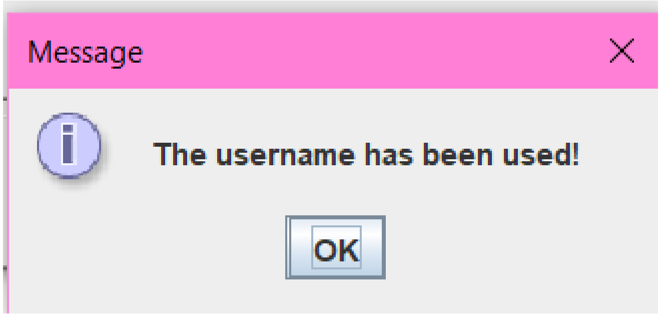
Username: kevin

Password: ••••••••••••••••••••••••

Gender : Male

Age : 21

Figure 9 Register with duplicate username



The username has been used!

Figure 10 Duplicate username found pop-up message

The next main menu option is Login, similar to the Register Panel, the user can login into three different accounts, Citizen, Non-Citizen and Admin.

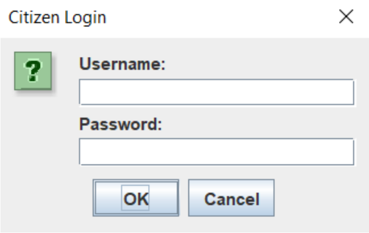


Figure 13 Citizen Login

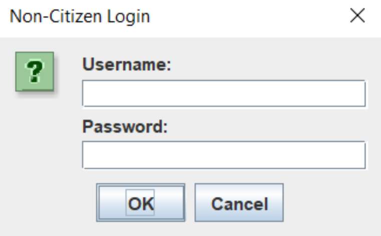


Figure 11 NonCitizen Login

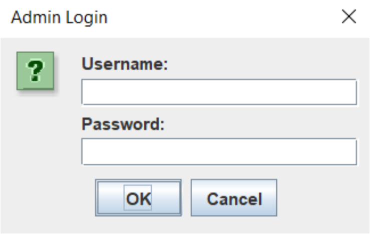


Figure 12 Admin Login

To login, the user is asked to input their registered username and password.

3.1.1 People

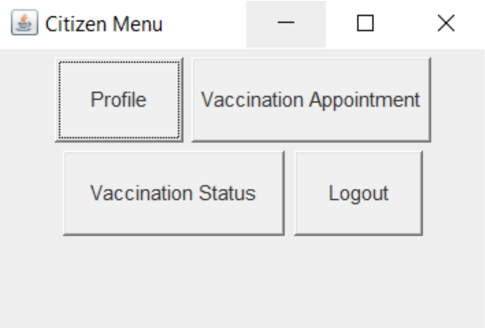
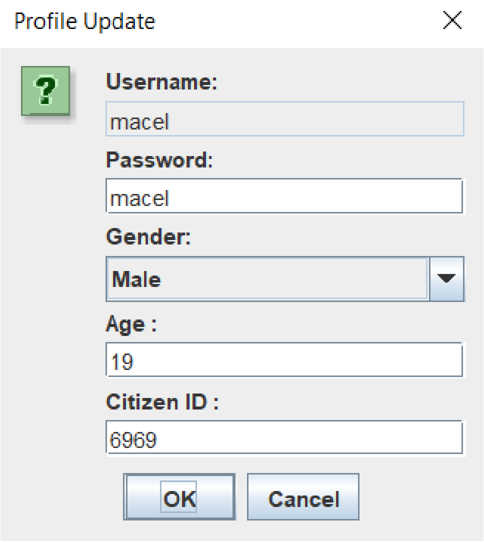


Figure 14 People Menu

In People (Citizen / Non-Citizen) menu there are 3 main functionalities which is Profile, Vaccination Status, Vaccination Appointment, and a logout button.



Profile Update

Username: macel

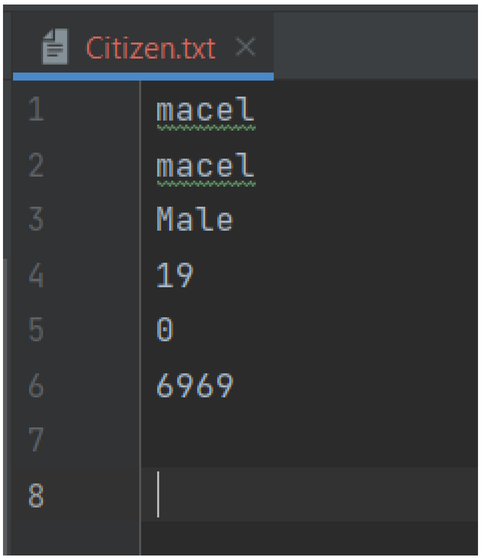
Password: macel

Gender: Male

Age : 19

Citizen ID : 6969

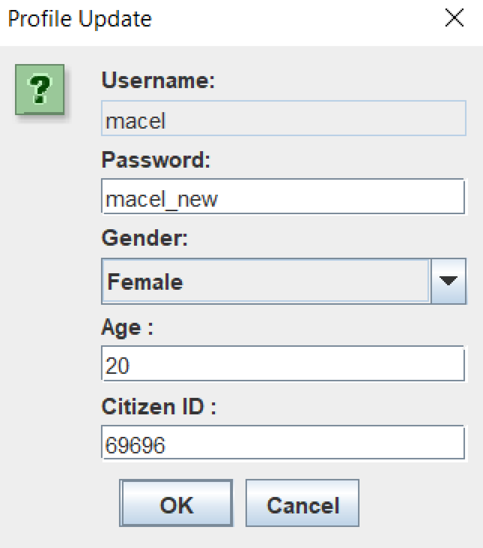
Figure 15 People Profile Update Initial



```
macel
macel
Male
19
0
6969

```

Figure 16 Record Before Update

By clicking on Profile button, the system will show the user their personal information. Here, the user can also change their password, gender, age, citizen ID for Citizen / passport number for NonCitizen. However, the username is not available for change.



Profile Update

Username: macel

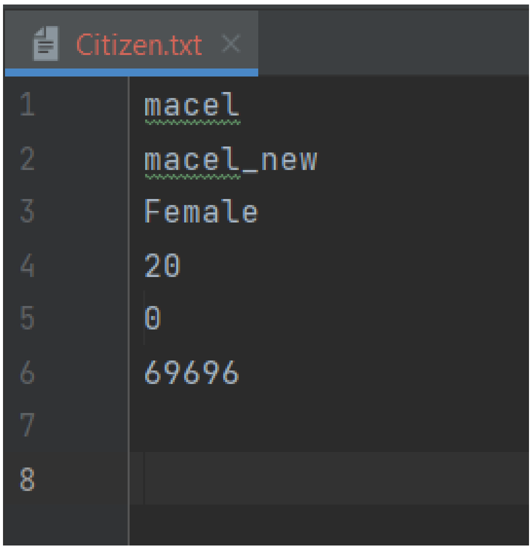
Password: macel_new

Gender: Female

Age : 20

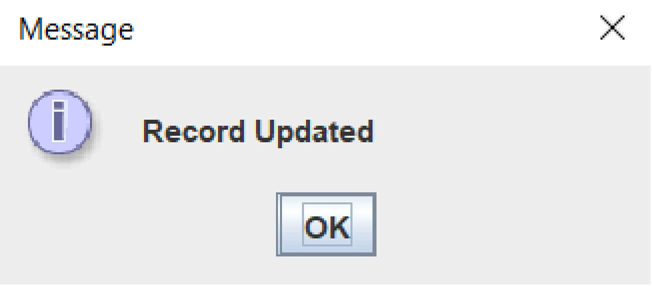
Citizen ID : 69696

Figure 17 People Profile Update Updated



```
macel
macel_new
Female
20
0
69696

```

Figure 18 Record After Update



Record Updated

Figure 19 Record updated pop-up message

By clicking OK, the user personal detail changes will be updated and immediately reflected in the txt file.

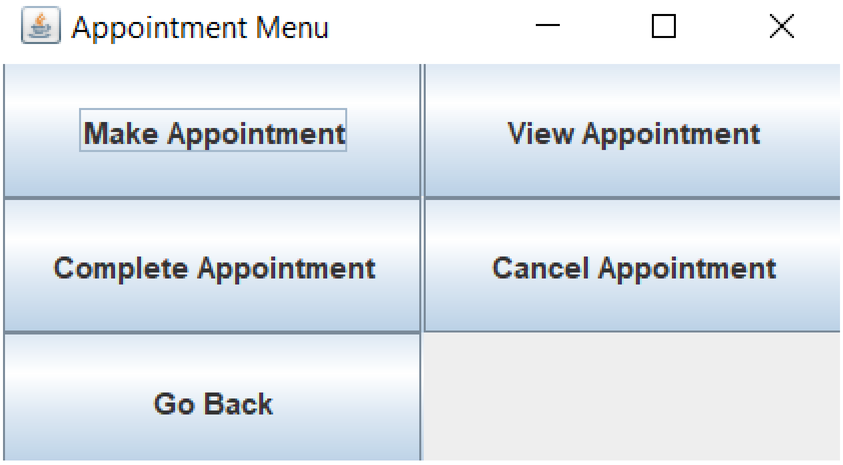
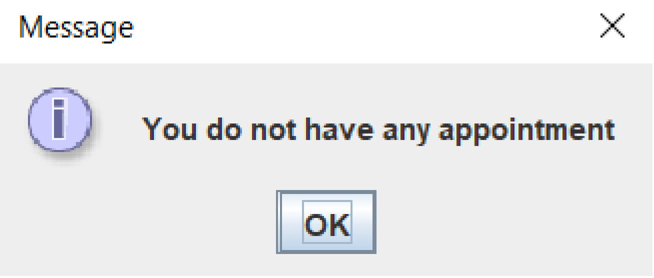


Figure 20 People Appointment Menu

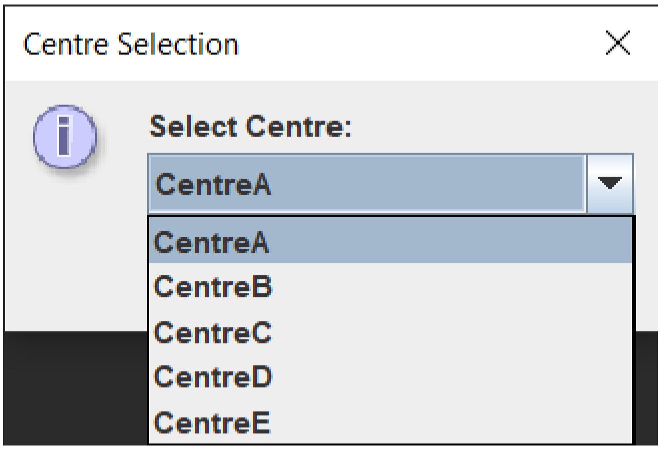
Next, appointment menu had 4 main functionalities which is Make Appointment, View Appointment, Complete Appointment, and Cancel Appointment.



You do not have any appointment

Figure 21 No Appointment pop-up message

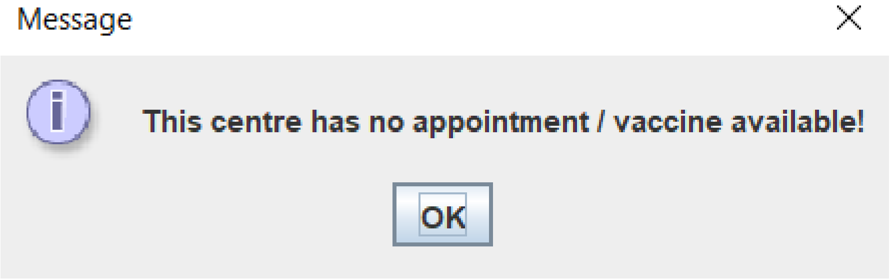
If the user hadn’t made any appointment yet, View Appointment, Complete Appointment, and Cancel Appointment will not be accessible and a pop-up message will show that the user does not have any appointment.



Centre Selection

Select Centre: CentreA

Figure 22 Make Appointment Centre Selection

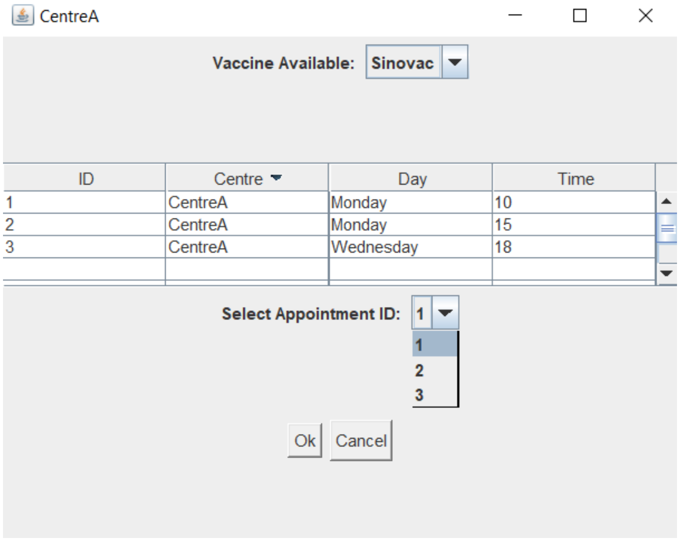


Message

This centre has no appointment / vaccine available!

Figure 23 Centre has no appointment schedule / vaccine assigned pop up message

By clicking make appointment button, user will be given a drop-down list of centres. If the chosen centre doesn't have any appointment schedule and/or vaccine assigned, the user can't make any appointment in that specific centre.



CentreA

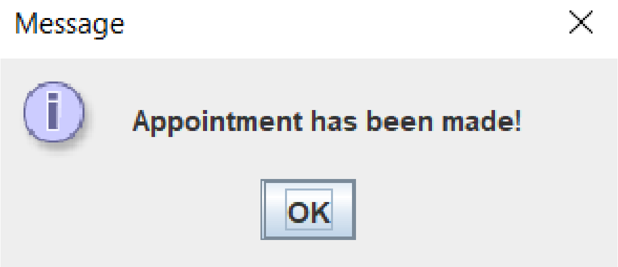
Vaccine Available: Sinovac

ID	Centre	Day	Time
1	CentreA	Monday	10
2	CentreA	Monday	15
3	CentreA	Wednesday	18

Select Appointment ID: 1

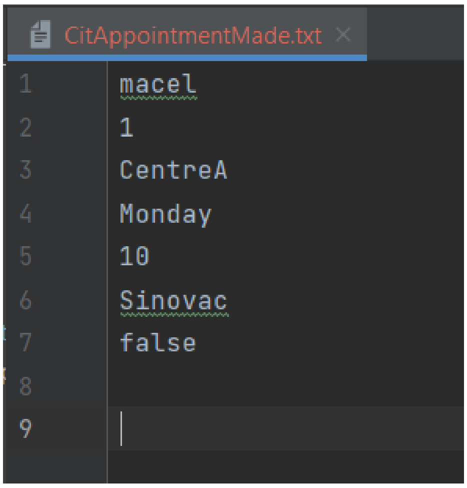
Figure 24 Appointment Schedule Selection

In contrast if the centre had appointment scheduled and vaccine assigned, it will show the vaccine available, a table showing the appointment schedule, and a list where user can pick their preferred appointment ID.



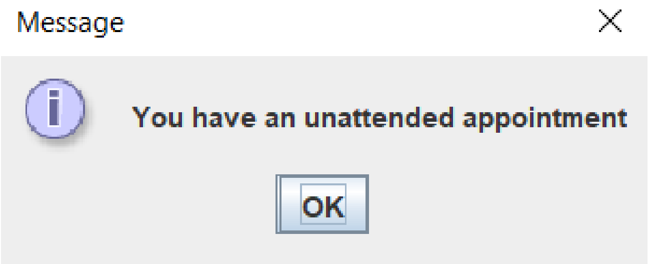
Appointment has been made!

Figure 25 Appointment made pop-up message



```
macel
1
CentreA
Monday
10
Sinovac
false


```

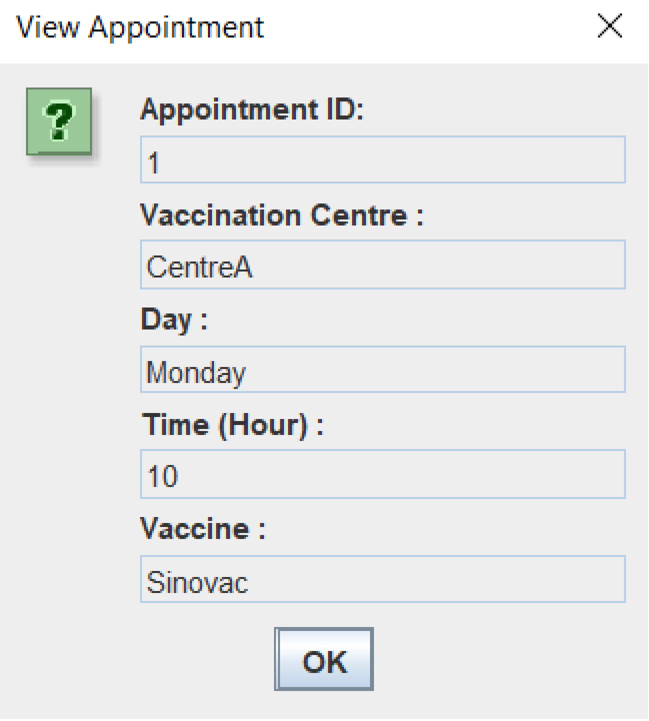
Figure 26 Appointment details recorded in txt file



You have an unattended appointment

Figure 27 Unattended Apppontment pop-up message

After clicking Ok button, a pop-up message will show that the appointment has been made. In the txt file, status of the appointment will be initially set to false, indicating that the appointment has not been completed. Afterwards, the user will not be able to make another appointment while they had an unattended / uncompleted appointment.



View Appointment

Appointment ID: 1

Vaccination Centre : CentreA

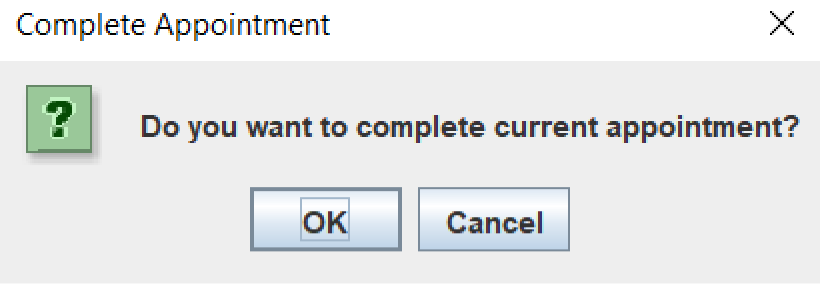
Day : Monday

Time (Hour) : 10

Vaccine : Sinovac

Figure 28 View Appointment

View Appointment function will show the user current appointment details.

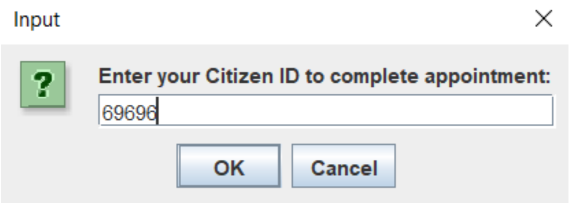


Complete Appointment

Do you want to complete current appointment?

Figure 29 Complete Appointment

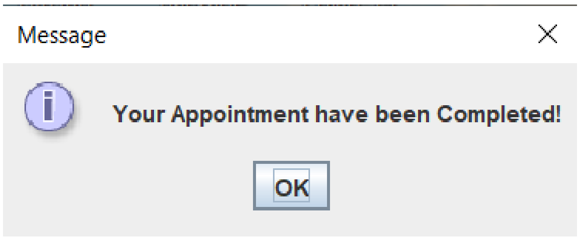
If the user has attended the appointment, they can use Complete Appointment function to finish current appointment.



Input

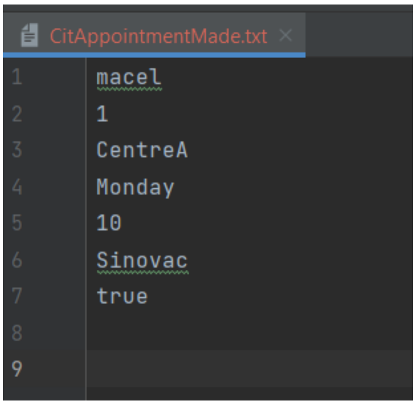
Enter your Citizen ID to complete appointment: 69696

Figure 30 Complete Appointment Validation



Your Appointment have been Completed!

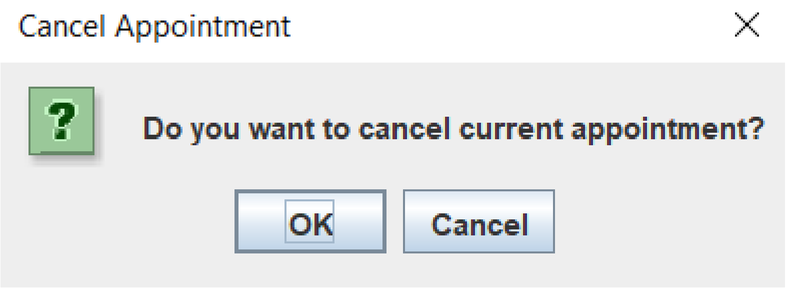
Figure 31 Appointment Completed pop-up message



```
macel
1
CentreA
Monday
10
Sinovac
true
```

Figure 32 Appointment Status changed

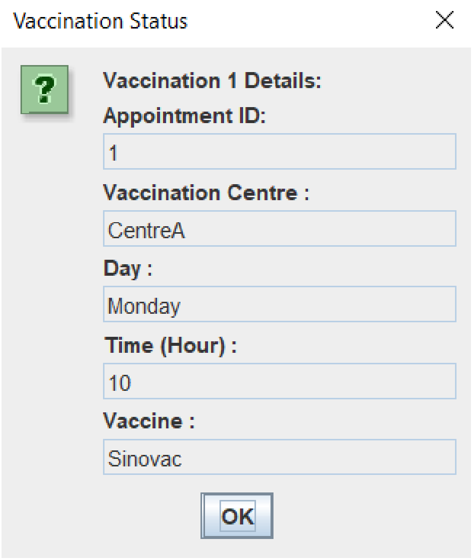
To complete the appointment, the user needed to input their citizen ID / passport number, if it matches, their appointment will be completed. The appointment status in txt file will be changed to true indicating that the user has completed their appointment.



Do you want to cancel current appointment?

Figure 33 Cancel Appointment

Cancel Appointment will let user cancel their current unattended appointment, by clicking OK their appointment will be deleted. The current appointment will then be deleted from the txt file.



Vaccination Status

Vaccination 1 Details:

Appointment ID: 1

Vaccination Centre : CentreA

Day : Monday

Time (Hour) : 10

Vaccine : Sinovac

Figure 34 Vaccination Status

User can see their all their completed vaccination details through Vaccination Status.

3.1.2 Admin

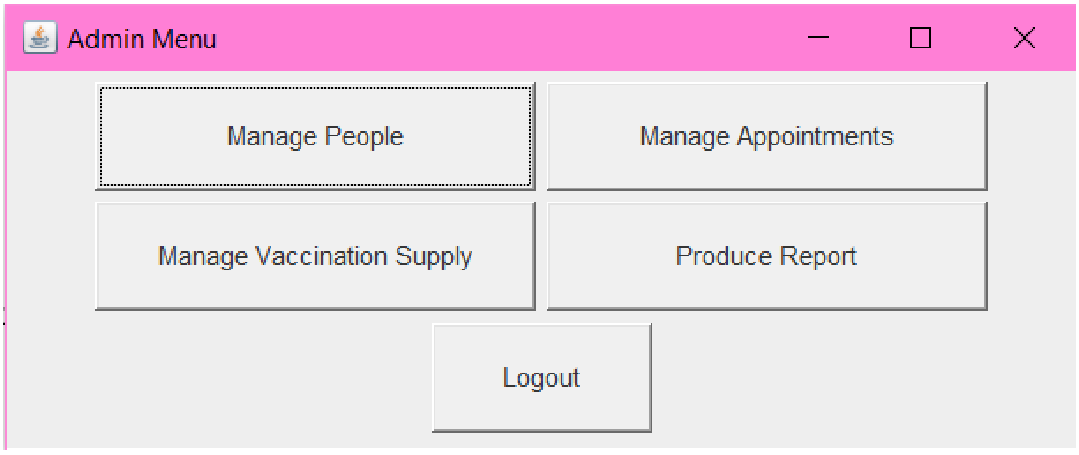


Figure 35 Admin Menu

Upon logging in to Admin account there will be 4 functions prompted by the system, which is Manage People, Manage Appointments, Manage Vaccination Supply, Produce Report.

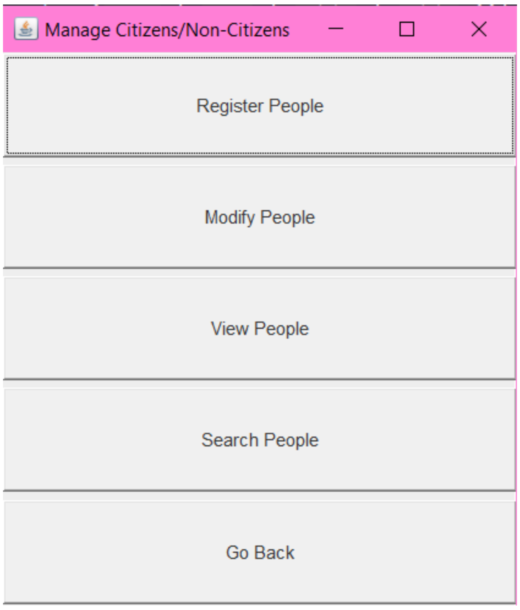


Figure 36 Manage People Menu

In Manage People function, there are 4 branch options, which are Register People, Modify People, View People and Search People.

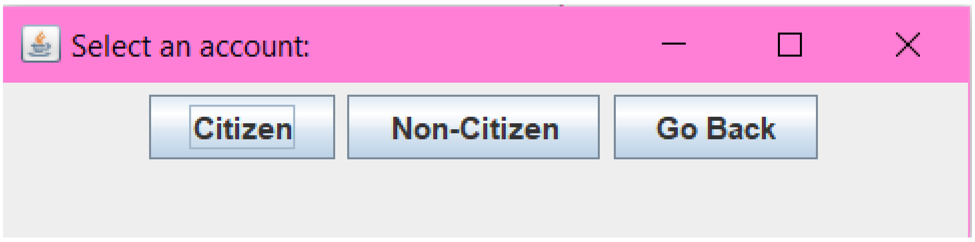
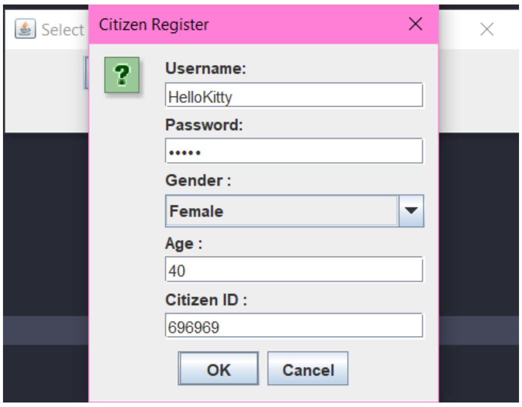
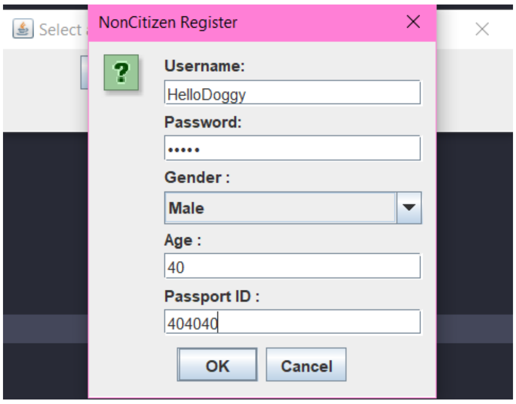


Figure 37 Select Account option



Username: HelloKitty
Password: •••••
Gender : Female
Age : 40
Citizen ID : 696969

Figure 38 Admin Citizen Register



Username: HelloDoggy
Password: •••••
Gender : Male
Age : 40
Passport ID : 404040

Figure 39 Admin NonCitizen Register

If the admin selects the first option Register People. The program will ask which account the admin wants to register. Selecting these options will call the similar function as the general register one.

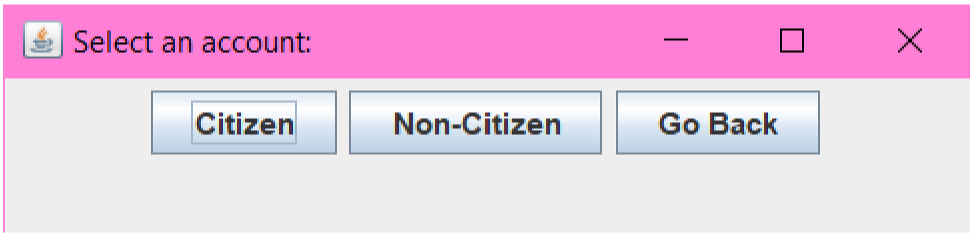
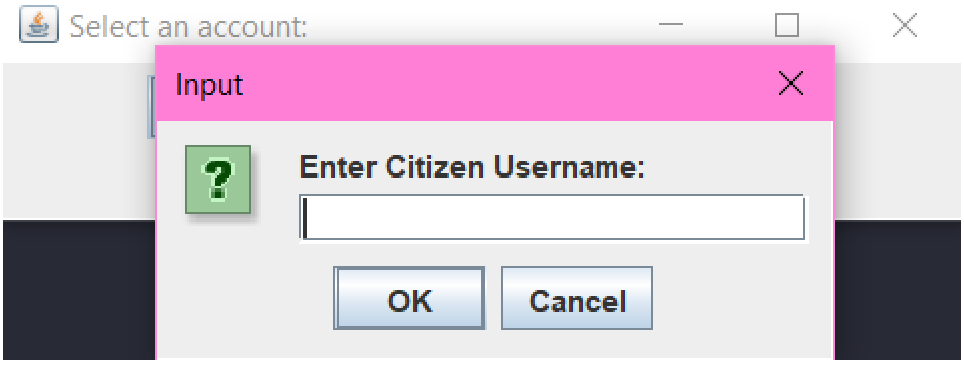


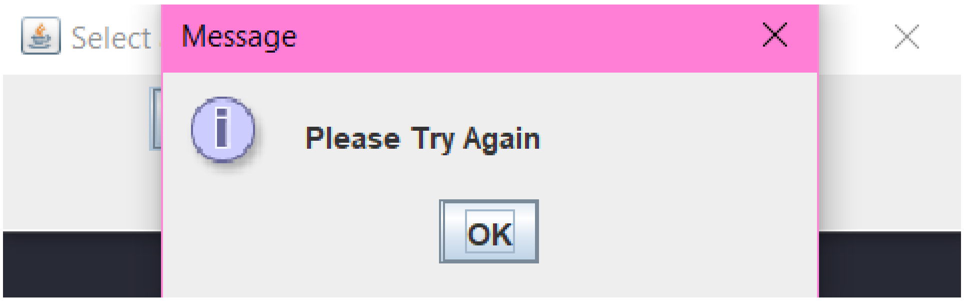
Figure 40 Select Account Option

The second option from the Manage People is the Modify People, upon pressing this button, a similar prompt will pop up asking the user to select an account to modify. Upon selecting one of the accounts, the system will prompt the admin to enter a citizen username.



Enter Citizen Username:

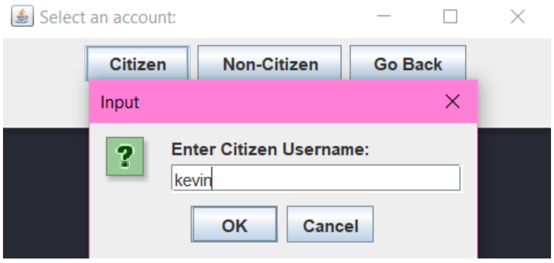
Figure 41 Enter username



Please Try Again

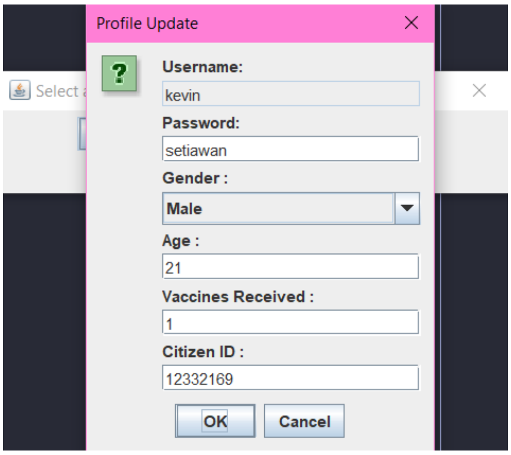
Figure 42 Try Again pop-up message

If the admin enters a username that is not recorded in the txt file, the system will send a pop-up message and tells the user to try again.



Enter Citizen Username:
kevin

Figure 43 Correct Username Input



Username: kevin
Password: setiawan
Gender : Male
Age : 21
Vaccines Received : 1
Citizen ID : 12332169

Figure 44 Profile Update Function Accessed

If the username matches one in the record (txt file), the admin will be able to change all details about the account except for username.

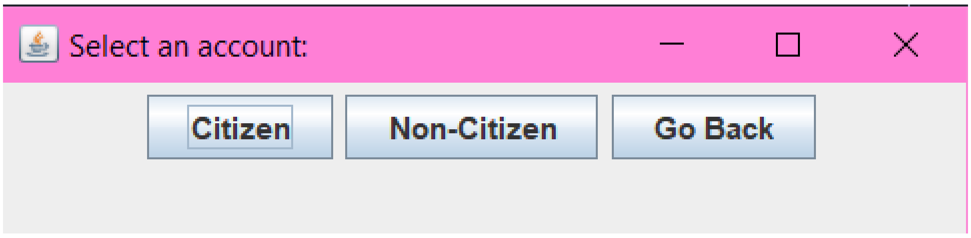
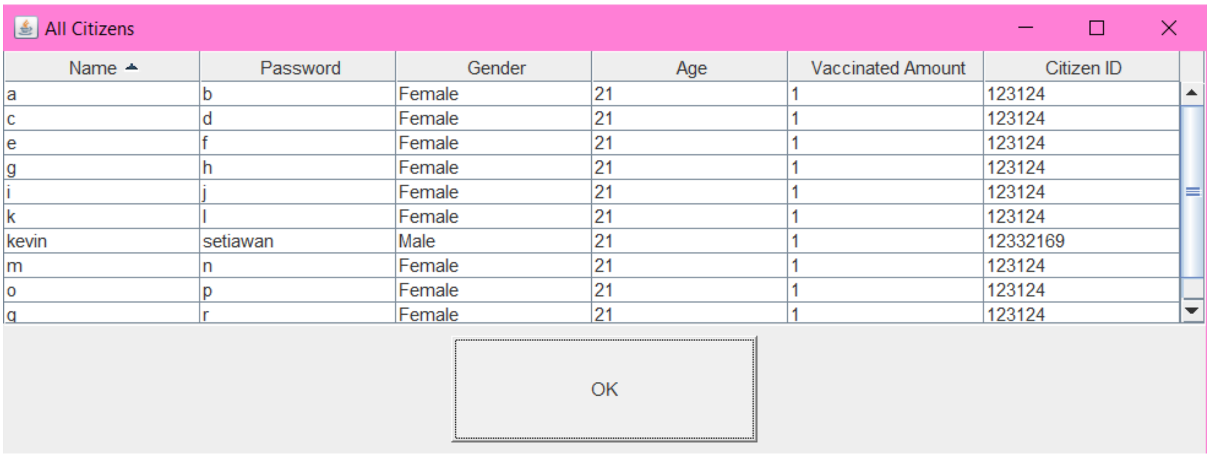


Figure 45 Select Account Option

The third Manage People option is View People, like last time, the system prompts the user to select an account type.



Name ▲	Password	Gender	Age	Vaccinated Amount	Citizen ID
a	b	Female	21	1	123124
c	d	Female	21	1	123124
e	f	Female	21	1	123124
g	h	Female	21	1	123124
i	j	Female	21	1	123124
k	l	Female	21	1	123124
kevin	setiawan	Male	21	1	12332169
m	n	Female	21	1	123124
o	p	Female	21	1	123124
q	r	Female	21	1	123124

Figure 46 View People

After the user selects an account type, the system will show all Citizens/Non-Citizens (Based on the selected account) details, [this data is sorted]

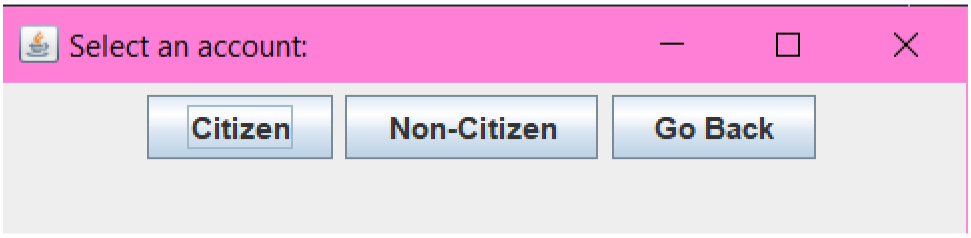
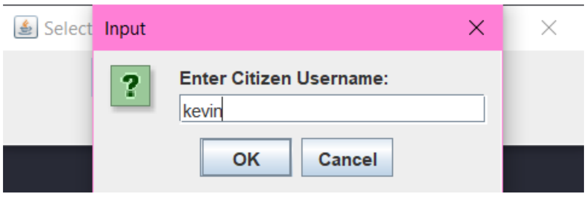


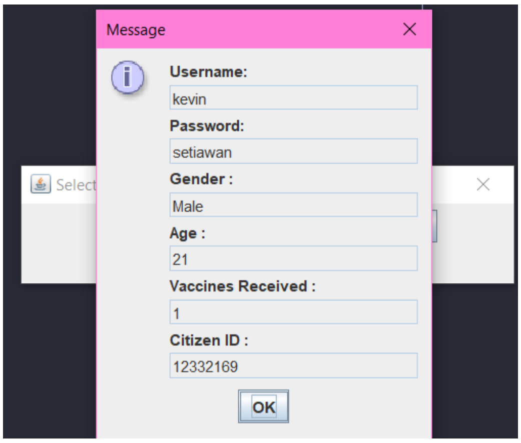
Figure 47 Select Account Option

The last Manage People functions is Search People, and like last time, the system prompts the user to select an account type.



Enter Citizen Username: kevin

Figure 48 Enter username



Username: kevin
Password: setiawan
Gender : Male
Age : 21
Vaccines Received : 1
Citizen ID : 12332169

Figure 49 User Details Displayed

After entering a username, if the name exists in the txt file, the details about that user will be displayed.

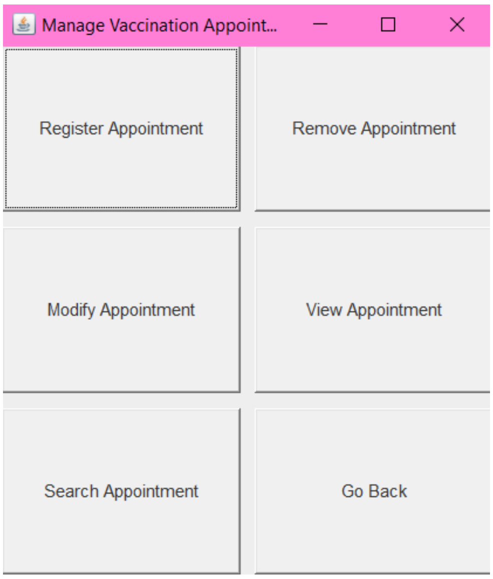
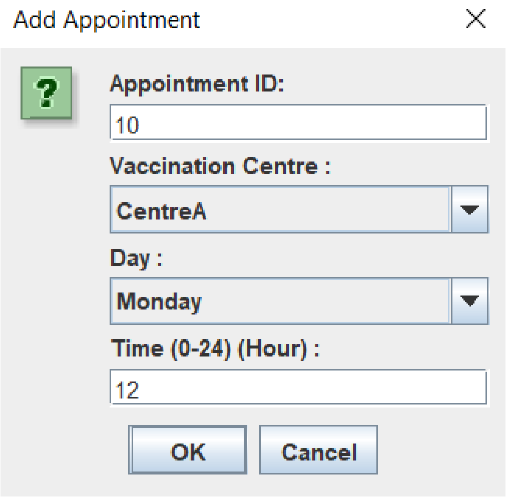


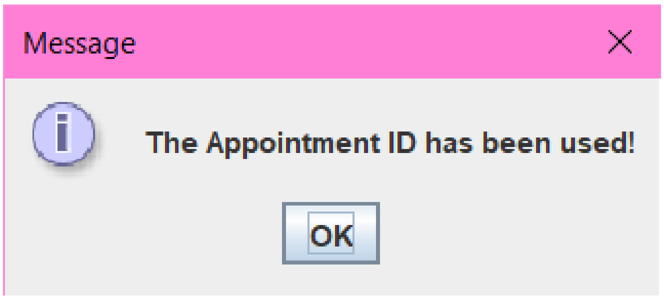
Figure 50 Manage Appointment Menu

The next admin menu option is Manage Appointment. Upon pressing this button, there will be 5 options to choose, which is Register Appointment, Remove Appointment, Modify Appointment, View Appointment, and Search Appointment.



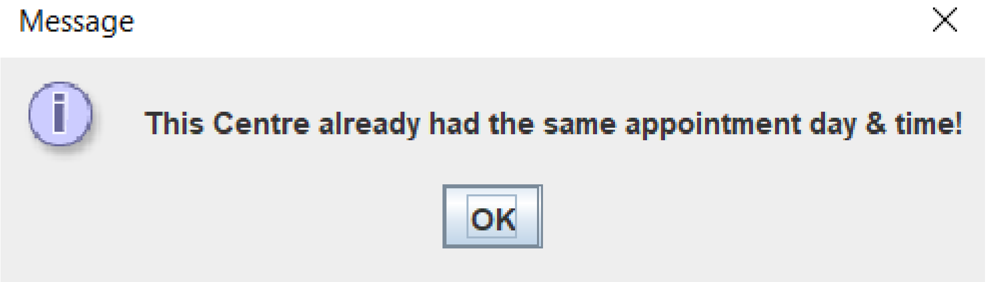
Appointment ID: 10
Vaccination Centre : CentreA
Day : Monday
Time (0-24) (Hour) : 12

Figure 51 Add Appointment



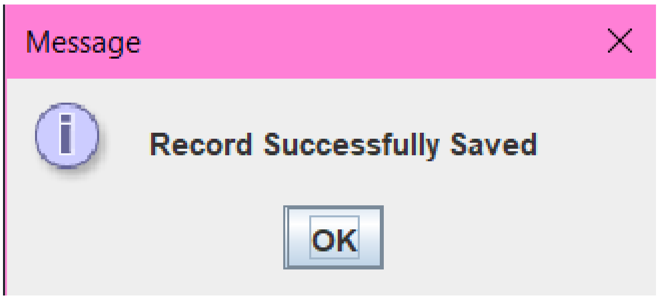
The Appointment ID has been used!

Figure 52 Apppointment ID used pop-up message



This Centre already had the same appointment day & time!

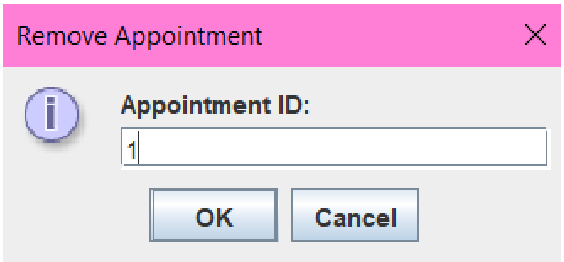
Figure 53 Centre already had the same appointment schedule pop-up message



Record Successfully Saved

Figure 54 Record Successfully Saved pop-up message

The first option is Register Appointment, when the admin presses Register Appointment button, the user needs to fill in the appointment info. Appointment ID should have no duplicates, if there are duplicates, system will decline and say Appointment ID has been used. Also, if admin tries to register a new appointment with same day & time as a registered appointment in the same centre, the new appointment will be rejected. Also, the time should fall between a range of 0-24, else it will reject the new appointment registration. Finally, if all the details are accepted, the details will be saved to the txt file.

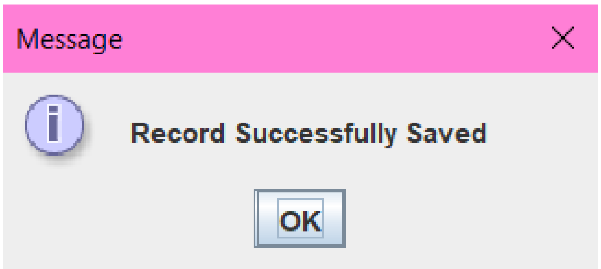


Remove Appointment

Appointment ID:

1

Figure 55 Enter Appointment ID to remove

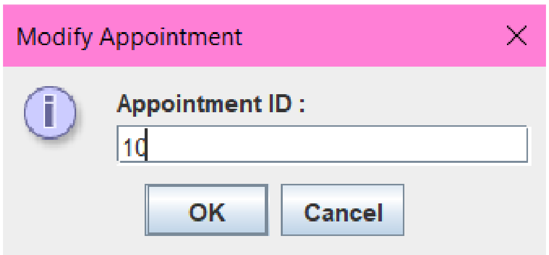


Message

Record Successfully Saved

Figure 56 Record Successfully Saved pop-up message

The second option is Remove Appointment. Upon pressing Remove Appointment button, the admin must enter an ID that's valid to remove the appointment, If successful, a message will pop up.

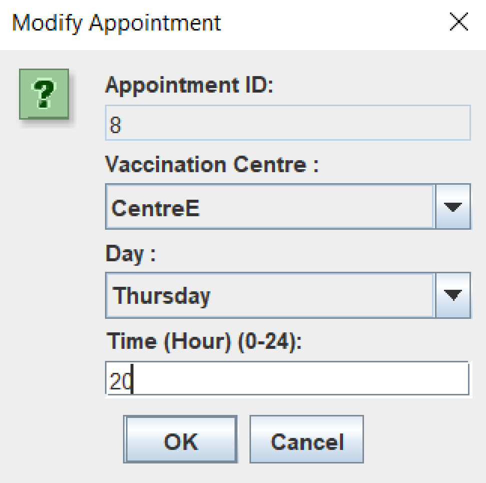


Modify Appointment

Appointment ID :

10

Figure 57 Enter Appointment ID to modify



Modify Appointment

Appointment ID:

8

Vaccination Centre :

CentreE

Day :

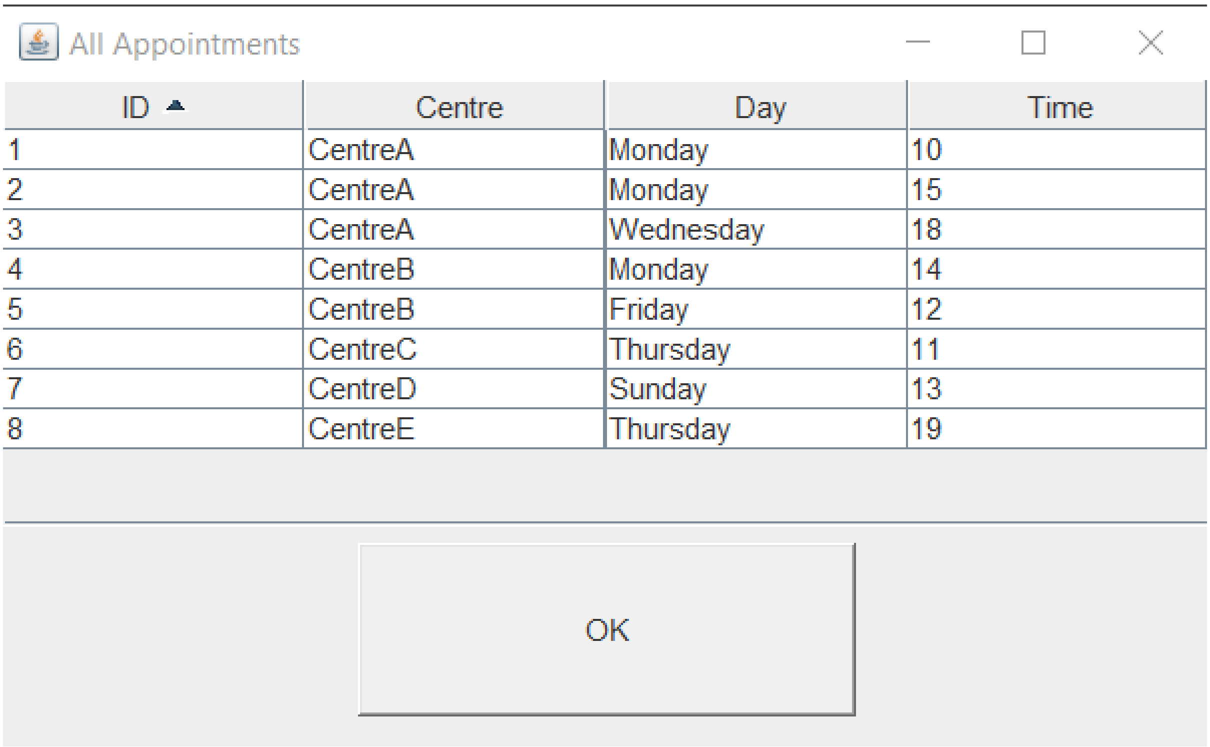
Thursday

Time (Hour) (0-24):

20

Figure 58 Modify Appointment

The third option is Modify Appointment, upon pressing Modify Appointment button, the admin has to enter in the appointment ID that wants to be modified. If the ID is accepted, the admin can then retype the appointment details, except for appointment ID. When modifying, time is also validated to only accept 0-24 range, else it the details will not be updated.

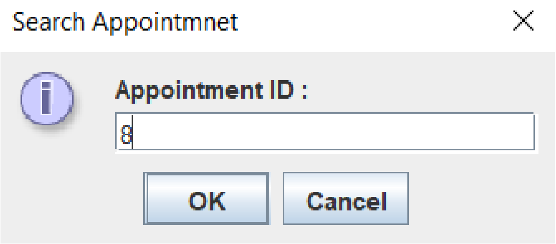


All Appointments

ID ▲	Centre	Day	Time
1	CentreA	Monday	10
2	CentreA	Monday	15
3	CentreA	Wednesday	18
4	CentreB	Monday	14
5	CentreB	Friday	12
6	CentreC	Thursday	11
7	CentreD	Sunday	13
8	CentreE	Thursday	19

Figure 59 View Appointment

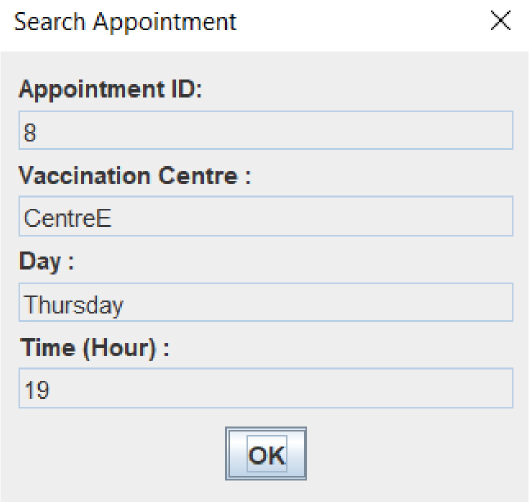
The fourth option is View Appointment, upon pressing this button, the Admin can see all available appointments.



Search Appointmnet

Appointment ID :
8

Figure 60 Enter Appointment ID



Search Appointment

Appointment ID:
8

Vaccination Centre :
CentreE

Day :
Thursday

Time (Hour) :
19

Figure 61 Appointment Details Generated

The last option in the Manage Appointment menu is Search Appointment, upon pressing this button, the Admin must enter in an available Appointment ID, and after it is validated, the full details of the appointment ID will be shown.

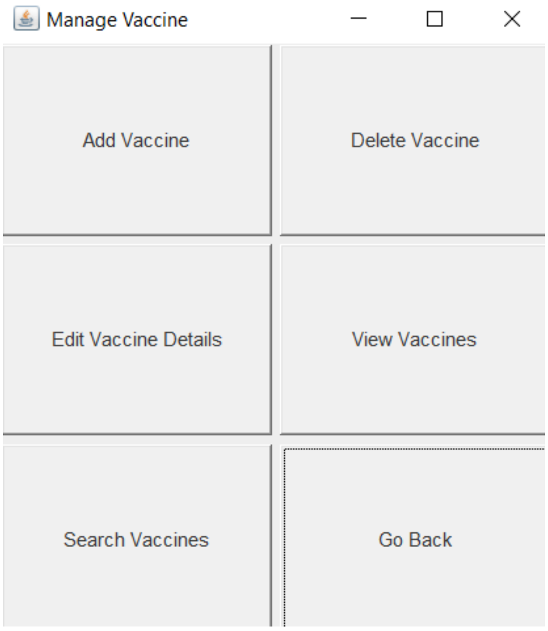
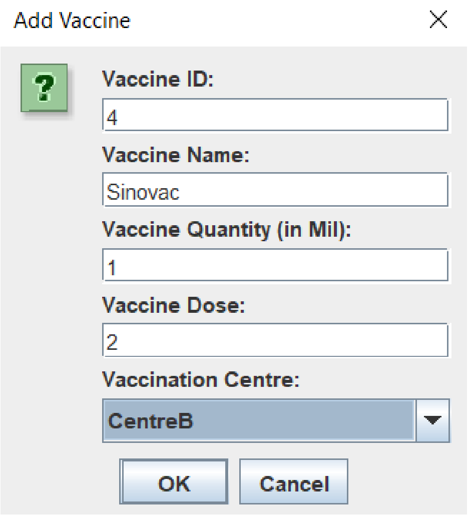


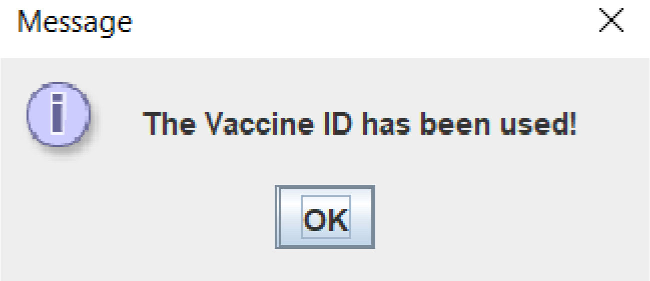
Figure 62 Manage Vaccination Supply Menu

The second to last admin menu function is Manage Vaccination Supply. Upon pressing this button, there will be 5 options to choose, which is Add Vaccine, Delete Appointment, Edit Vaccine Details, View Vaccines, and Search Vaccines.



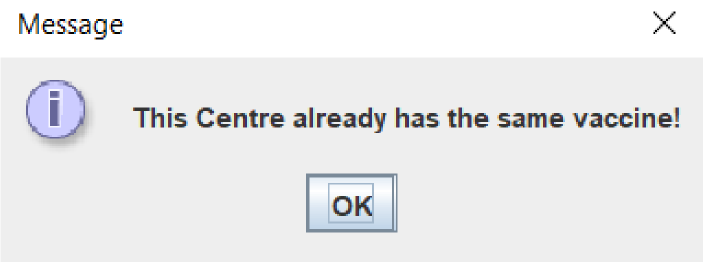
Vaccine ID: 4
Vaccine Name: Sinovac
Vaccine Quantity (in Mil): 1
Vaccine Dose: 2
Vaccination Centre: CentreB

Figure 63 Add Vaccine



The Vaccine ID has been used!

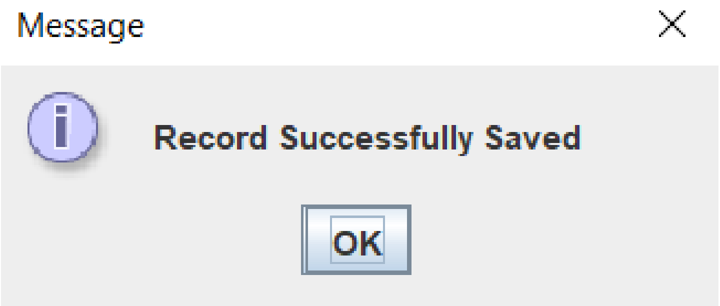
Figure 64 Vaccine ID has been used pop-up message



Message

This Centre already has the same vaccine!

Figure 65 Centre already has the same vaccine pop-up message

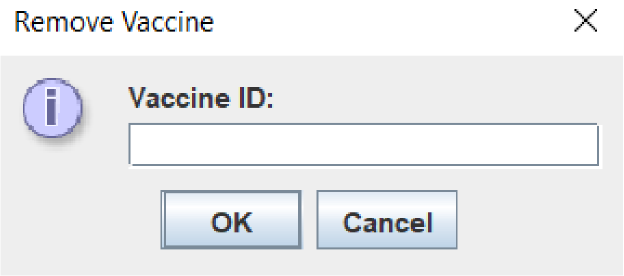


Message

Record Successfully Saved

Figure 66 Record successfully saved pop-up message

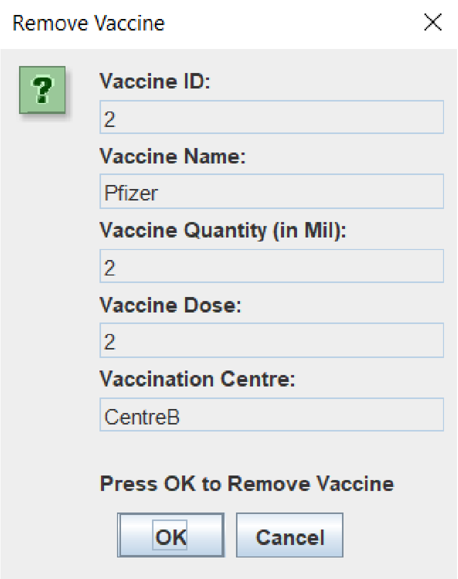
The first option is Add Vaccine. This allows admin to add new vaccine by entering their details such as ID, Vaccine Name, Quantity, Dose, and designated Vaccination Centre. There should be no duplicate Vaccine ID, if admin add a new vaccine with existing ID, the new vaccine will not be added. Also in one centre there should be no duplicate vaccine name, if the same vaccine name is inputted, a pop-up message indicating the centre already has the same vaccine will appear. Lastly after all details required are correct and accepted, record will be successfully saved.



Remove Vaccine

Vaccine ID:

Figure 67 Enter Vaccine ID to remove



Vaccine ID: 2

Vaccine Name: Pfizer

Vaccine Quantity (in Mil): 2

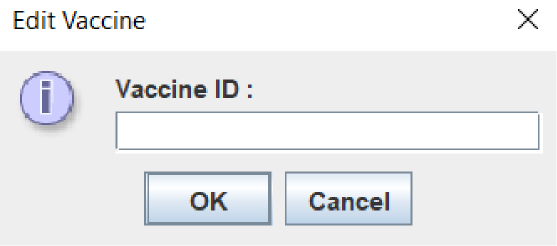
Vaccine Dose: 2

Vaccination Centre: CentreB

Press OK to Remove Vaccine

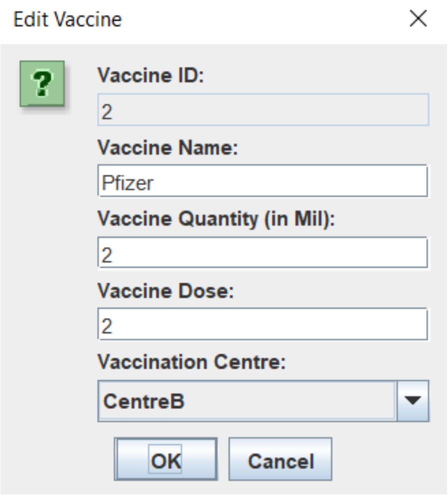
Figure 68 Vaccine Removal Confirmation

The second option is Delete Vaccine, this lets the admin delete a vaccine from the record. The admin needs to enter the ID of vaccine they wanted to delete. If the ID is found, then it will show the vaccine details. Admin can delete the vaccine by clicking on the OK button.



Vaccine ID :

Figure 69 Enter Vaccine ID to edit



Vaccine ID: 2

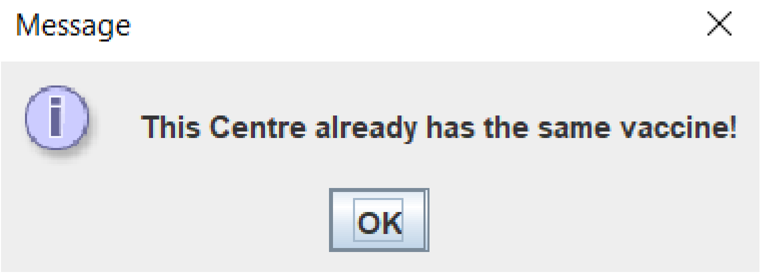
Vaccine Name: Pfizer

Vaccine Quantity (in Mil): 2

Vaccine Dose: 2

Vaccination Centre: CentreB

Figure 70 Edit Vaccine Details

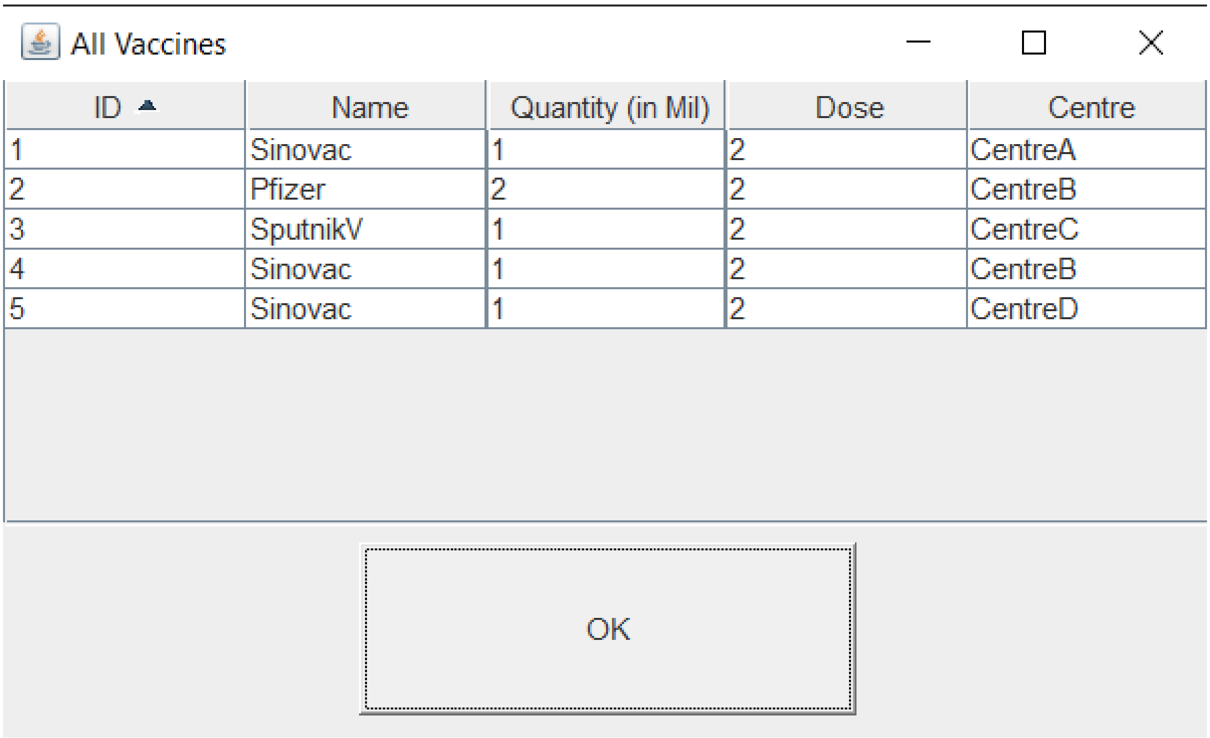


Message

This Centre already has the same vaccine!

Figure 71 Centre already has the same vaccine pop-up message

The third option is Edit Vaccine Details, this allows admin to edit vaccine details. To access editing function, admin needs to enter the ID of vaccine they wanted to edit. If the ID is found, admin can change the details, except Vaccine ID. However, if admin changed the vaccine into a centre with a same vaccine in it, the edit will be cancelled.



All Vaccines

ID ▲	Name	Quantity (in Mil)	Dose	Centre
1	Sinovac	1	2	CentreA
2	Pfizer	2	2	CentreB
3	SputnikV	1	2	CentreC
4	Sinovac	1	2	CentreB
5	Sinovac	1	2	CentreD

Figure 72 View Vaccines

The fourth option is View Vaccine, this allows admin to view all vaccine details that has been registered and saved through the system in a table form.

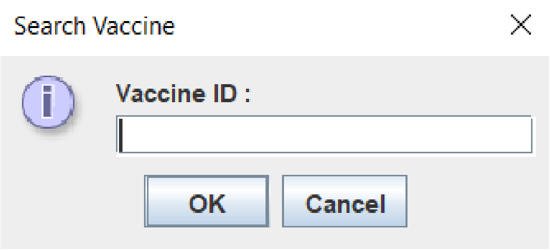
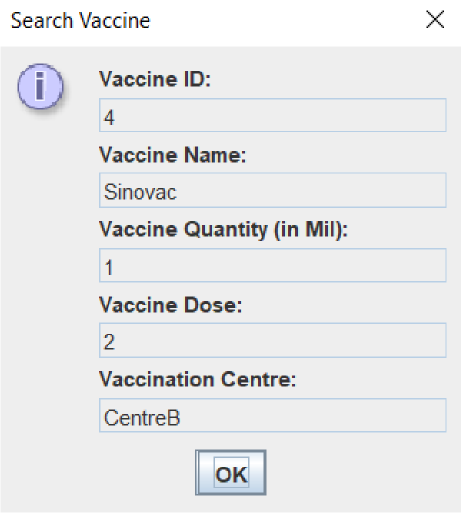


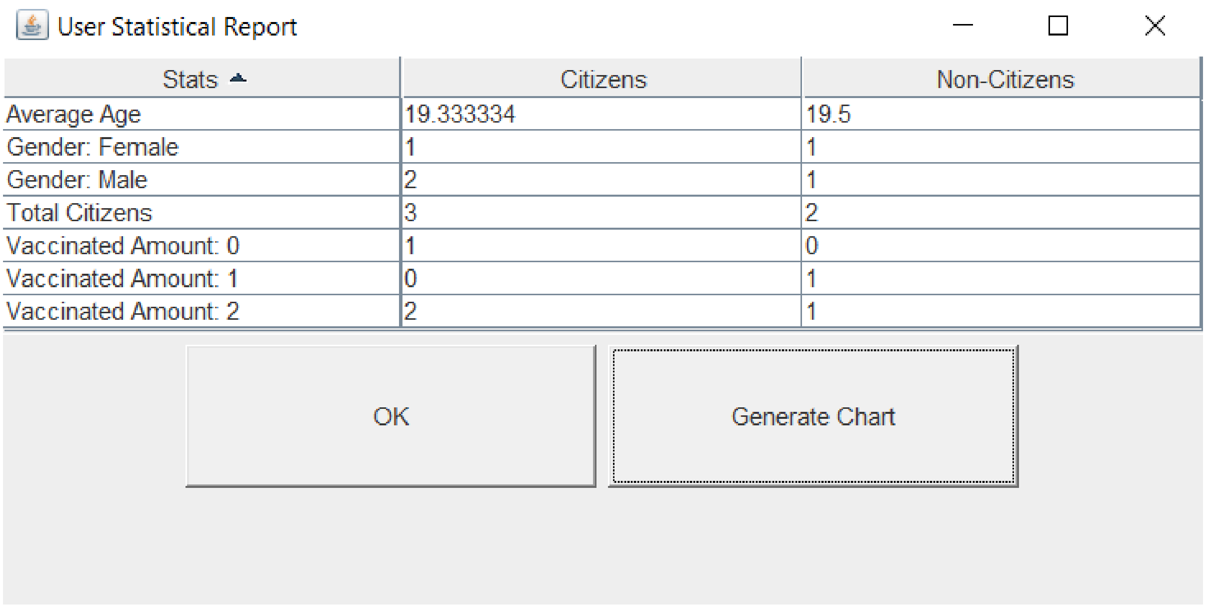
Figure 73 Enter Vaccine ID to search



Vaccine ID:
4
Vaccine Name:
Sinovac
Vaccine Quantity (in Mil):
1
Vaccine Dose:
2
Vaccination Centre:
CentreB

Figure 74 Searched Vaccine Details

The last option in Manage Vaccination Supply Menu is Search Vaccine. This allows admin to view vaccine details by entering their respective ID.



Stats ▲	Citizens	Non-Citizens
Average Age	19.333334	19.5
Gender: Female	1	1
Gender: Male	2	1
Total Citizens	3	2
Vaccinated Amount: 0	1	0
Vaccinated Amount: 1	0	1
Vaccinated Amount: 2	2	1

Figure 75 Produce Report

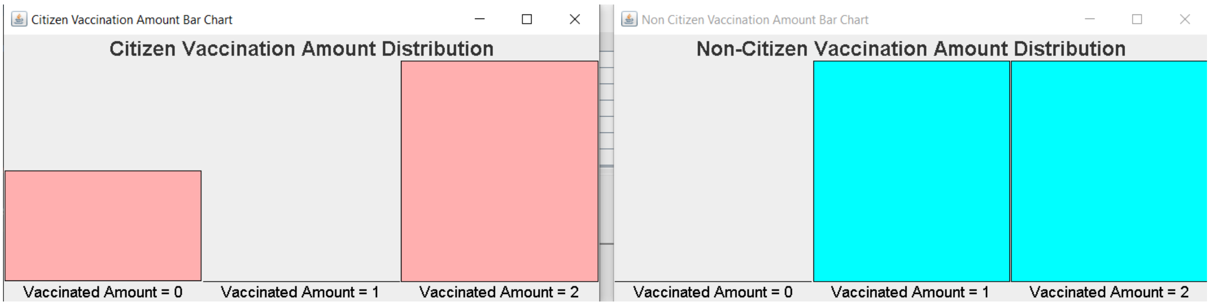
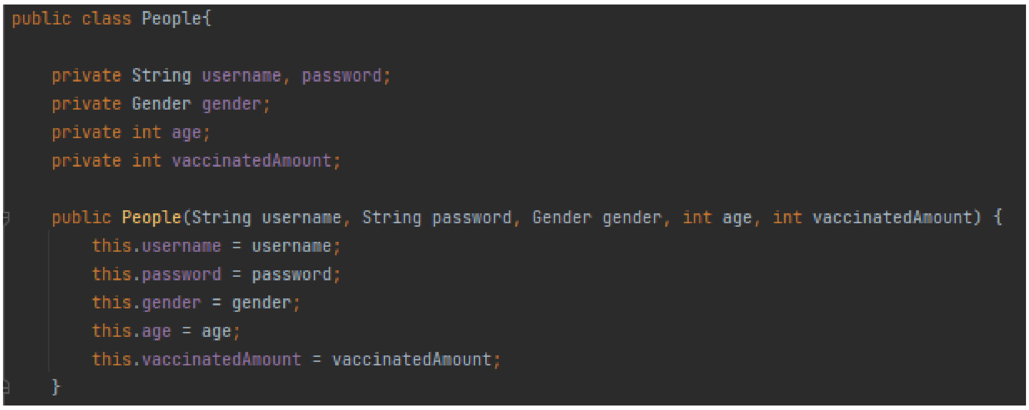

Citizen Vaccination Amount Bar Chart
Citizen Vaccination Amount Distribution
Vaccinated Amount = 0
Vaccinated Amount = 1
Vaccinated Amount = 2
Non Citizen Vaccination Amount Bar Chart
Non-Citizen Vaccination Amount Distribution
Vaccinated Amount = 0
Vaccinated Amount = 1
Vaccinated Amount = 2

Figure 76 Citizen & NonCitizen Vaccination Amount Distribution

The last function in the Admin Menu is Produce Report, this function will show statistical report such as Gender, Age, etc. Bar Chart of Vaccination Amount Distribution of Citizen & NonCitizen can also be generated by clicking ‘Generate Chart’ button.

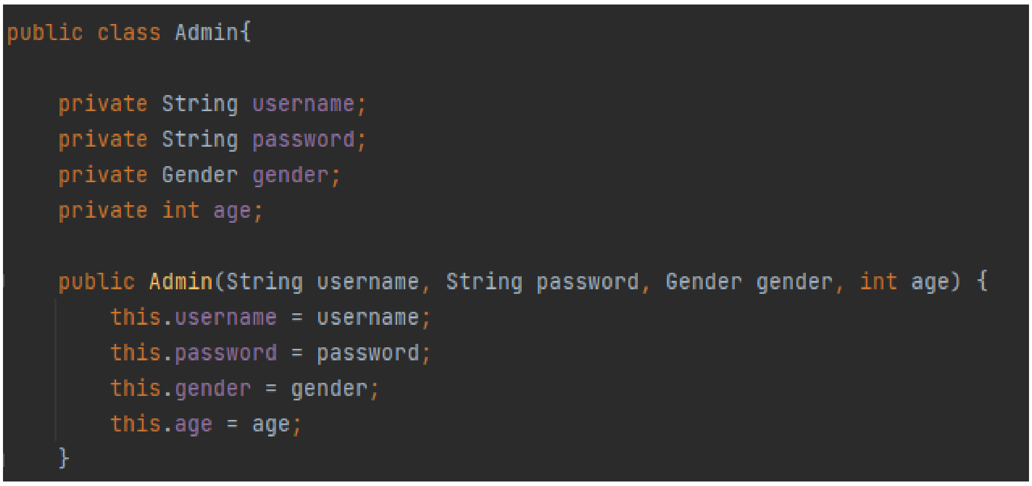
3.2 OO Concepts

3.2.1 Abstraction



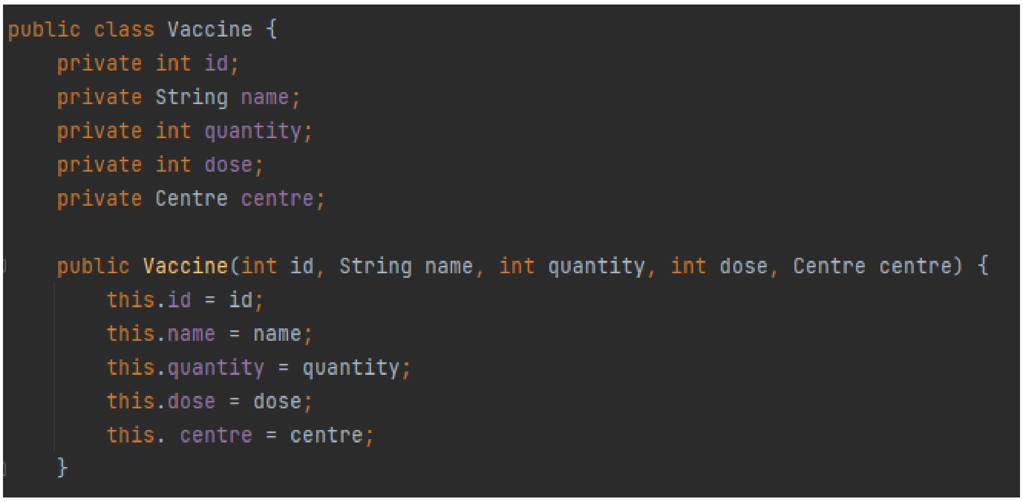
```java
public class People{

    private String username, password;
    private Gender gender;
    private int age;
    private int vaccinatedAmount;

    public People(String username, String password, Gender gender, int age, int vaccinatedAmount) {
        this.username = username;
        this.password = password;
        this.gender = gender;
        this.age = age;
        this.vaccinatedAmount = vaccinatedAmount;
    }
```

Figure 77 People Class



```java
public class Admin{

    private String username;
    private String password;
    private Gender gender;
    private int age;

    public Admin(String username, String password, Gender gender, int age) {
        this.username = username;
        this.password = password;
        this.gender = gender;
        this.age = age;
    }
```

Figure 78 Admin Class



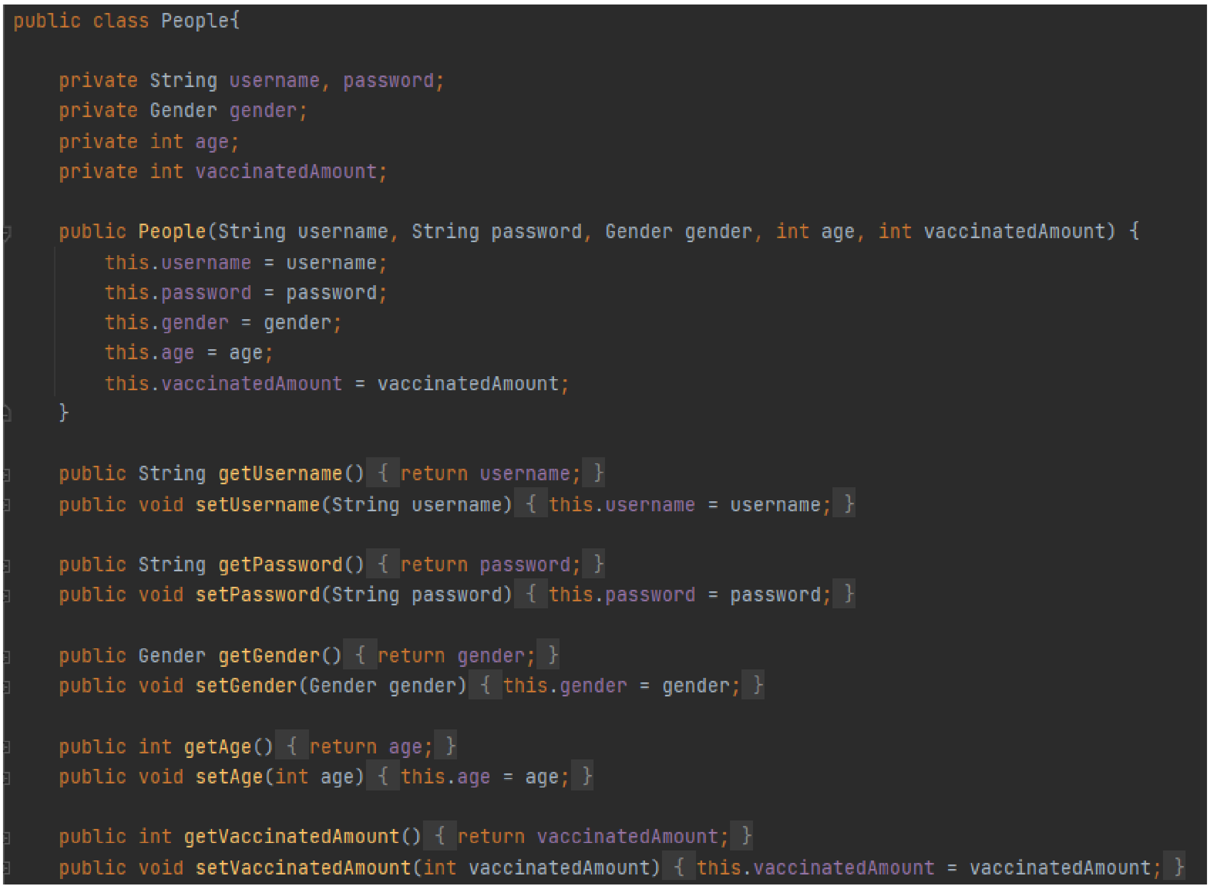
```java
public class Vaccine {
    private int id;
    private String name;
    private int quantity;
    private int dose;
    private Centre centre;

    public Vaccine(int id, String name, int quantity, int dose, Centre centre) {
        this.id = id;
        this.name = name;
        this.quantity = quantity;
        this.dose = dose;
        this. centre = centre;
    }
```

Figure 79 Vaccine Class

Abstraction is defined as the process of reducing the object to its essence so that only the necessary details are exposed to the users (TechVidvan, n.d.).

In the sample program abstraction has been applied. In each Modelling Class the details will be saved in one Object and the data used is relevant. People have Username, Password, Gender, and Vaccinated Amount. Admin have Username, Password, Gender, and Age. Vaccine have Id, Name, Quantity, Dose, and Centre.

3.2.2 Encapsulation



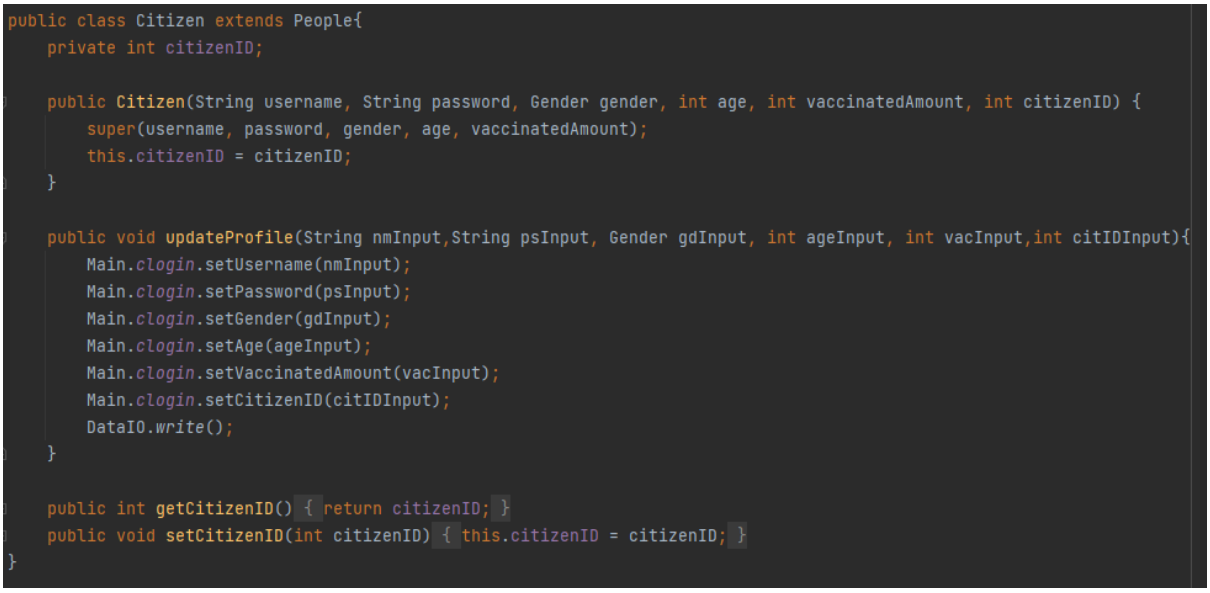
```java
public class People{

    private String username, password;
    private Gender gender;
    private int age;
    private int vaccinatedAmount;

    public People(String username, String password, Gender gender, int age, int vaccinatedAmount) {
        this.username = username;
        this.password = password;
        this.gender = gender;
        this.age = age;
        this.vaccinatedAmount = vaccinatedAmount;
    }

    public String getUsername() { return username; }
    public void setUsername(String username) { this.username = username; }

    public String getPassword() { return password; }
    public void setPassword(String password) { this.password = password; }

    public Gender getGender() { return gender; }
    public void setGender(Gender gender) { this.gender = gender; }

    public int getAge() { return age; }
    public void setAge(int age) { this.age = age; }

    public int getVaccinatedAmount() { return vaccinatedAmount; }
    public void setVaccinatedAmount(int vaccinatedAmount) { this.vaccinatedAmount = vaccinatedAmount; }
```

Figure 80 Encapsulation Sample Code

Encapsulation is a mechanism of wrapping variables and methods together as a single unit. In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class (tutorialspoint, n.d.).

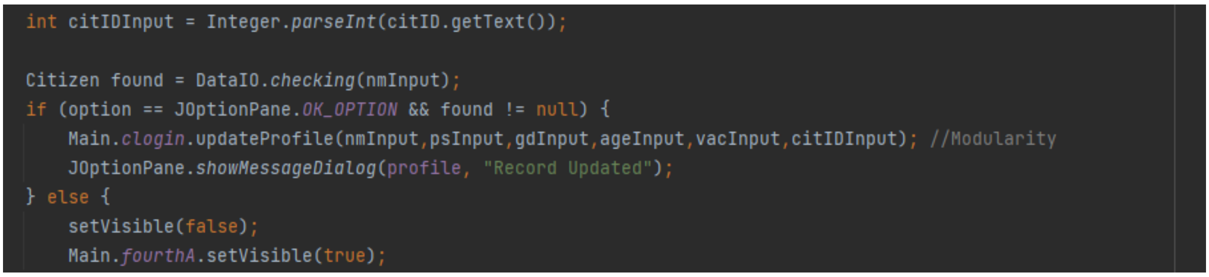
In the sample code, it can be seen that People class, the variables of People [Username, Password, Gender, Age, and Vaccinated Amount] has a ‘private’ modifier, which means that it cannot be accessed by other classes, and thus the only way to access these variables are through ‘Getter’ and ‘Setter’ methods provided.

3.2.3 Modularity



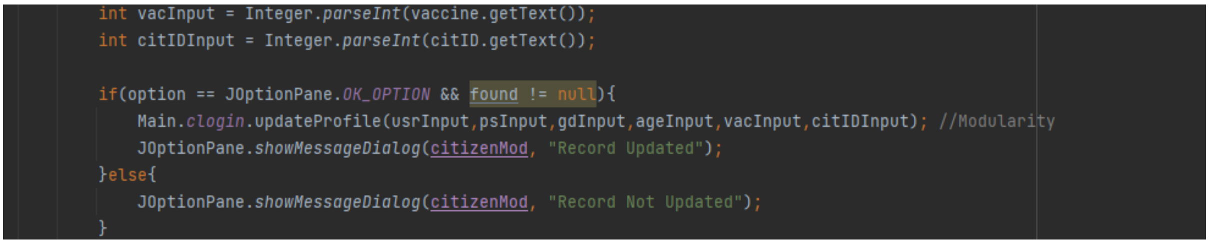
```java
public class Citizen extends People{
    private int citizenID;

    public Citizen(String username, String password, Gender gender, int age, int vaccinatedAmount, int citizenID) {
        super(username, password, gender, age, vaccinatedAmount);
        this.citizenID = citizenID;
    }

    public void updateProfile(String nmInput,String psInput, Gender gdInput, int ageInput, int vacInput,int citIDInput){
        Main.clogin.setUsername(nmInput);
        Main.clogin.setPassword(psInput);
        Main.clogin.setGender(gdInput);
        Main.clogin.setAge(ageInput);
        Main.clogin.setVaccinatedAmount(vacInput);
        Main.clogin.setCitizenID(citIDInput);
        DataIO.write();
    }

    public int getCitizenID() { return citizenID; }
    public void setCitizenID(int citizenID) { this.citizenID = citizenID; }
}
```

Figure 81 updateProfile method in Citizen Class



```java
int citIDInput = Integer.parseInt(citID.getText());

Citizen found = DataIO.checking(nmInput);
if (option == JOptionPane.OK_OPTION && found != null) {
    Main.clogin.updateProfile(nmInput,psInput,gdInput,ageInput,vacInput,citIDInput); //Modularity
    JOptionPane.showMessageDialog(profile, "Record Updated");
} else {
    setVisible(false);
    Main.fourthA.setVisible(true);
```

Figure 82 updateProfile method called in Citizen Update Profile Menu



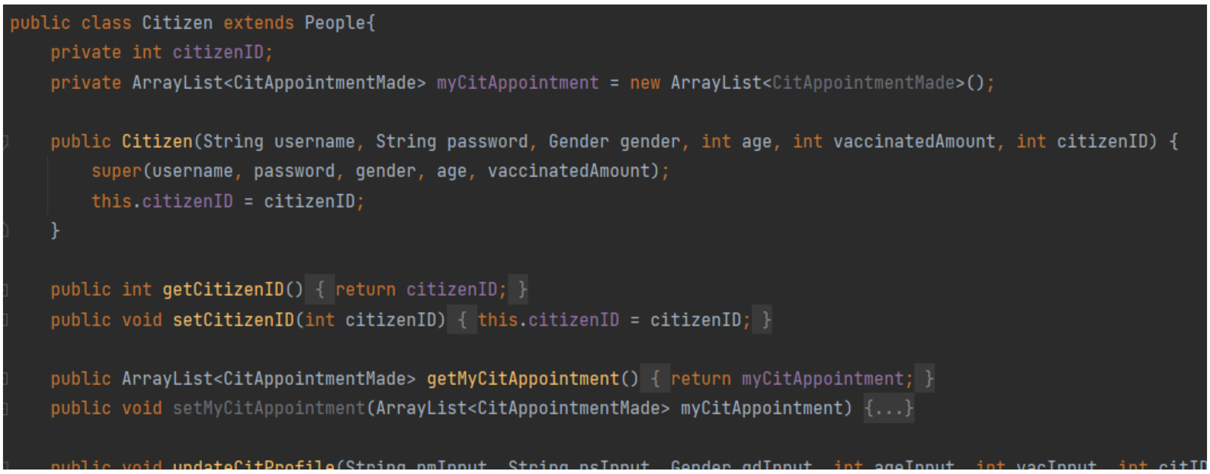
```java
int vacInput = Integer.parseInt(vaccine.getText());
int citIDInput = Integer.parseInt(citID.getText());

if(option == JOptionPane.OK_OPTION && found != null){
    Main.clogin.updateProfile(usrInput,psInput,gdInput,ageInput,vacInput,citIDInput); //Modularity
    JOptionPane.showMessageDialog(citizenMod, "Record Updated");
}else{
    JOptionPane.showMessageDialog(citizenMod, "Record Not Updated");
}
```

Figure 83 updateProfile method called in Admin Modify Citizen Profile

Modularity is the process of decomposing a program into a set of modules so as to reduce the overall complexity of the problem.

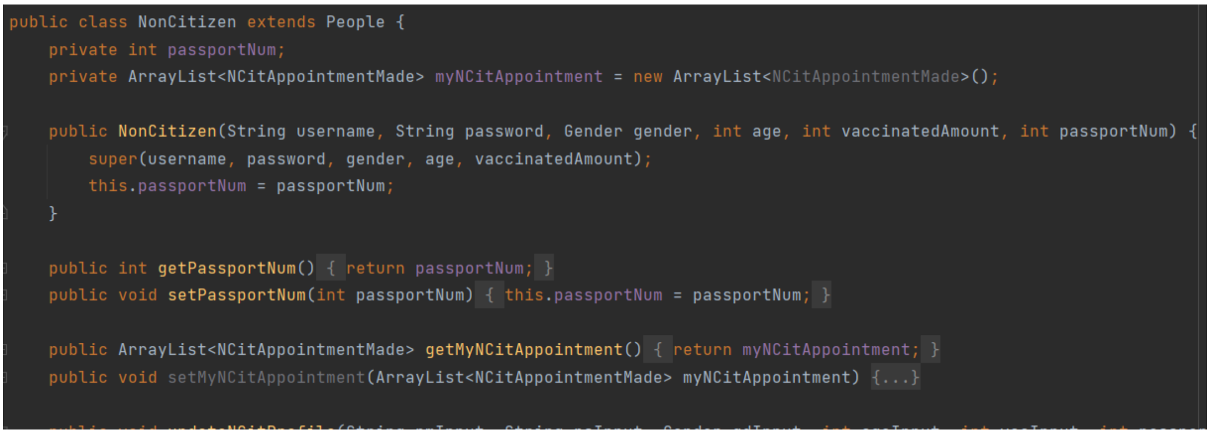
In the sample code, updateProfile method is created in Citizen class. The method will promote re-usability as it can be used in both update profile menu for citizen and admin. This goes as well for non-citizen and admin.

3.2.4 Inheritance



```java
public class Citizen extends People{
    private int citizenID;
    private ArrayList<CitAppointmentMade> myCitAppointment = new ArrayList<CitAppointmentMade>();

    public Citizen(String username, String password, Gender gender, int age, int vaccinatedAmount, int citizenID) {
        super(username, password, gender, age, vaccinatedAmount);
        this.citizenID = citizenID;
    }

    public int getCitizenID() { return citizenID; }
    public void setCitizenID(int citizenID) { this.citizenID = citizenID; }

    public ArrayList<CitAppointmentMade> getMyCitAppointment() { return myCitAppointment; }
    public void setMyCitAppointment(ArrayList<CitAppointmentMade> myCitAppointment) {...}

    public void updateCitProfile(String nmInput, String psInput, Gender gdInput, int ageInput, int vacInput, int citID
```

Figure 84 Citizen class inherited from People Class indicated by extends



```java
public class NonCitizen extends People {
    private int passportNum;
    private ArrayList<NCitAppointmentMade> myNCitAppointment = new ArrayList<NCitAppointmentMade>();

    public NonCitizen(String username, String password, Gender gender, int age, int vaccinatedAmount, int passportNum) {
        super(username, password, gender, age, vaccinatedAmount);
        this.passportNum = passportNum;
    }

    public int getPassportNum() { return passportNum; }
    public void setPassportNum(int passportNum) { this.passportNum = passportNum; }

    public ArrayList<NCitAppointmentMade> getMyNCitAppointment() { return myNCitAppointment; }
    public void setMyNCitAppointment(ArrayList<NCitAppointmentMade> myNCitAppointment) {...}

    public void updateNCitProfile(String nmInput, String psInput, Gender gdInput, int ageInput, int vacInput, int passpor
```

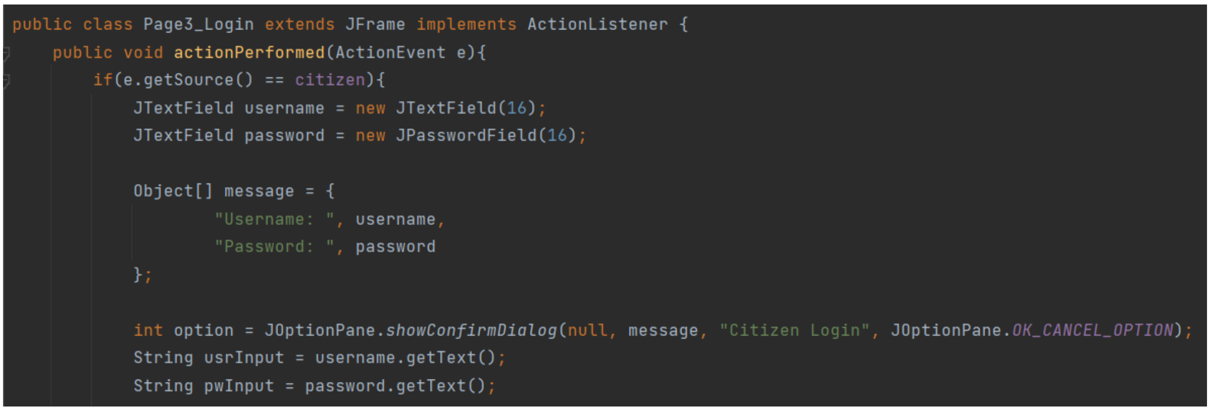
Figure 85 NonCitizen Class inherited from People Class indicated by extends

Inheritance is a relationship among classes where a class can derive from another class forming a hierarchy of classes that share a set of attributes and methods (Janssen, 2017).

In the above sample code, Citizen and NonCitizen are an inheritance from People, indicated by the keyword *extends*. People is the superclass while Citizen and NonCitizen are subclasses, meaning all the variables and methods from the People class can be used by Citizen / NonCitizen class, these include ‘Getter’ and ‘Setter’ of all the People variables.

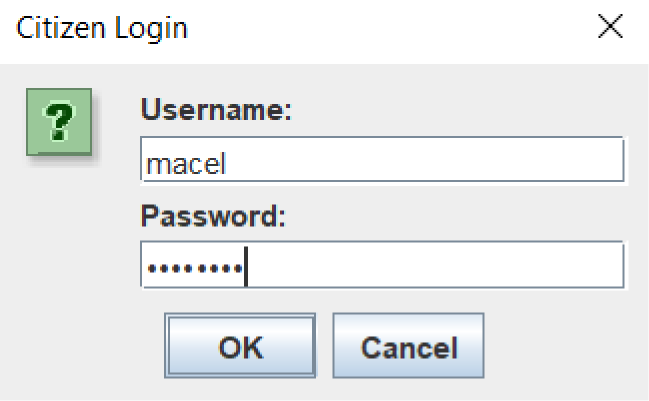
3.3 Additional Features

3.3.1 JPasswordField



```
public class Page3_Login extends JFrame implements ActionListener {
    public void actionPerformed(ActionEvent e){
        if(e.getSource() == citizen){
            JTextField username = new JTextField(16);
            JTextField password = new JPasswordField(16);

            Object[] message = {
                    "Username: ", username,
                    "Password: ", password
            };

            int option = JOptionPane.showConfirmDialog(null, message, "Citizen Login", JOptionPane.OK_CANCEL_OPTION);
            String usrInput = username.getText();
            String pwInput = password.getText();
```

Figure 86 JPasswordField sample code



Citizen Login

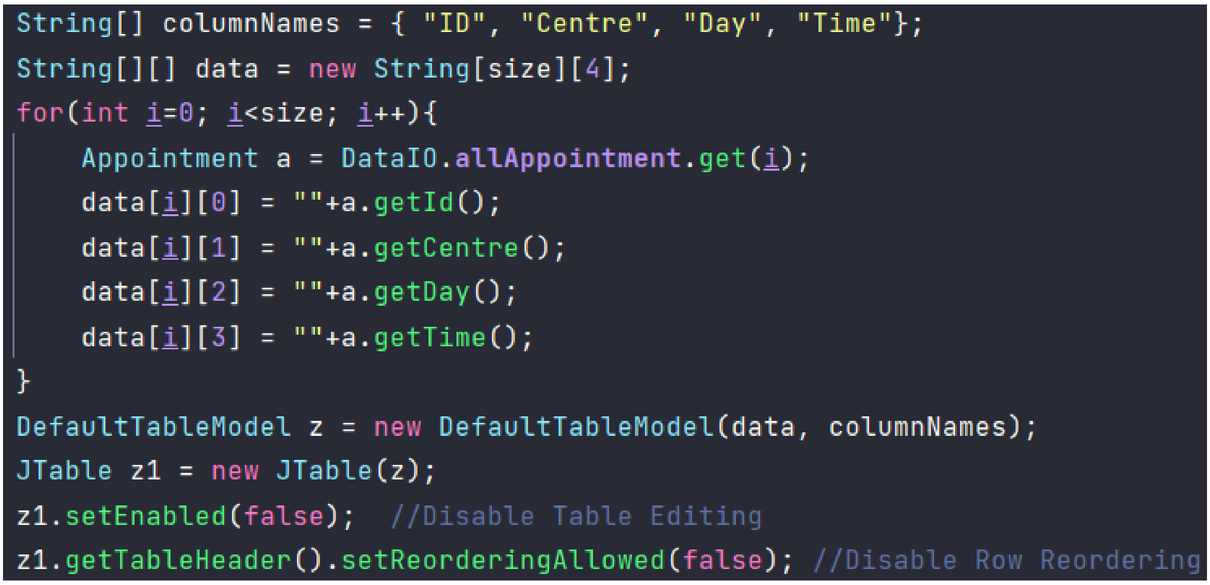
Username: macel

Password: ••••••••

Figure 87 JPasswordField output

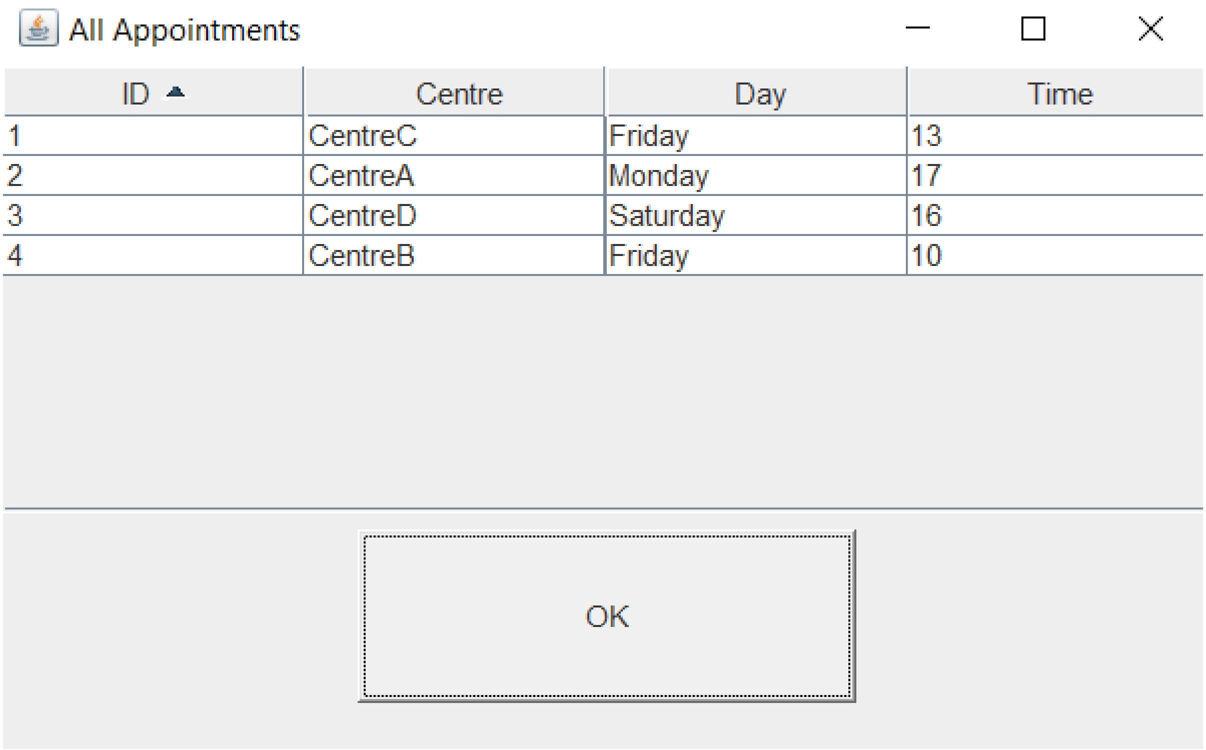
JPasswordField is a lightweight component that allows the editing of a single line of text where the view indicates something was typed, but does not show the original characters (Oracle, n.d.). It is available in java.swing package. To upheld confidentiality when a user login to the system, JPasswordField will be used.

3.3.2 JTable



```
String[] columnNames = { "ID", "Centre", "Day", "Time"};
String[][] data = new String[size][4];
for(int i=0; i<size; i++){
    Appointment a = DataIO.allAppointment.get(i);
    data[i][0] = ""+a.getId();
    data[i][1] = ""+a.getCentre();
    data[i][2] = ""+a.getDay();
    data[i][3] = ""+a.getTime();
}
DefaultTableModel z = new DefaultTableModel(data, columnNames);
JTable z1 = new JTable(z);
z1.setEnabled(false);  //Disable Table Editing
z1.getTableHeader().setReorderingAllowed(false); //Disable Row Reordering
```

Figure 88 JTable sample code



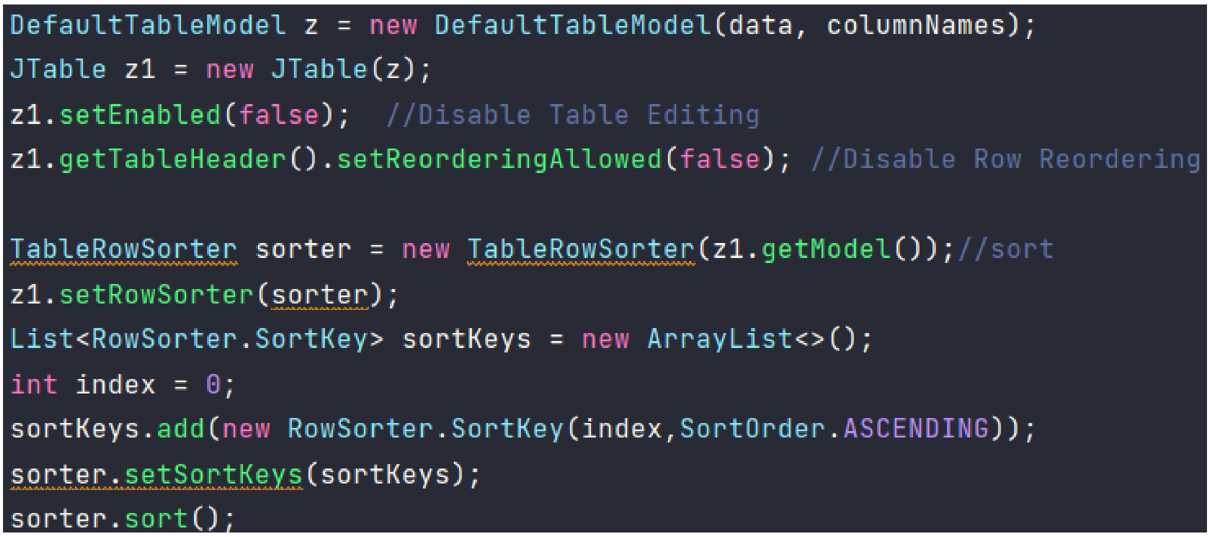
ID ▲	Centre	Day	Time
1	CentreC	Friday	13
2	CentreA	Monday	17
3	CentreD	Saturday	16
4	CentreB	Friday	10

Figure 89 JTable output

JTable is a component from the javax.swing package, and it can be used to make a table with ease, and this extra feature eases the job of managing lists of multiple data attributes, such as in this case: ID, Centre, Day, Time.

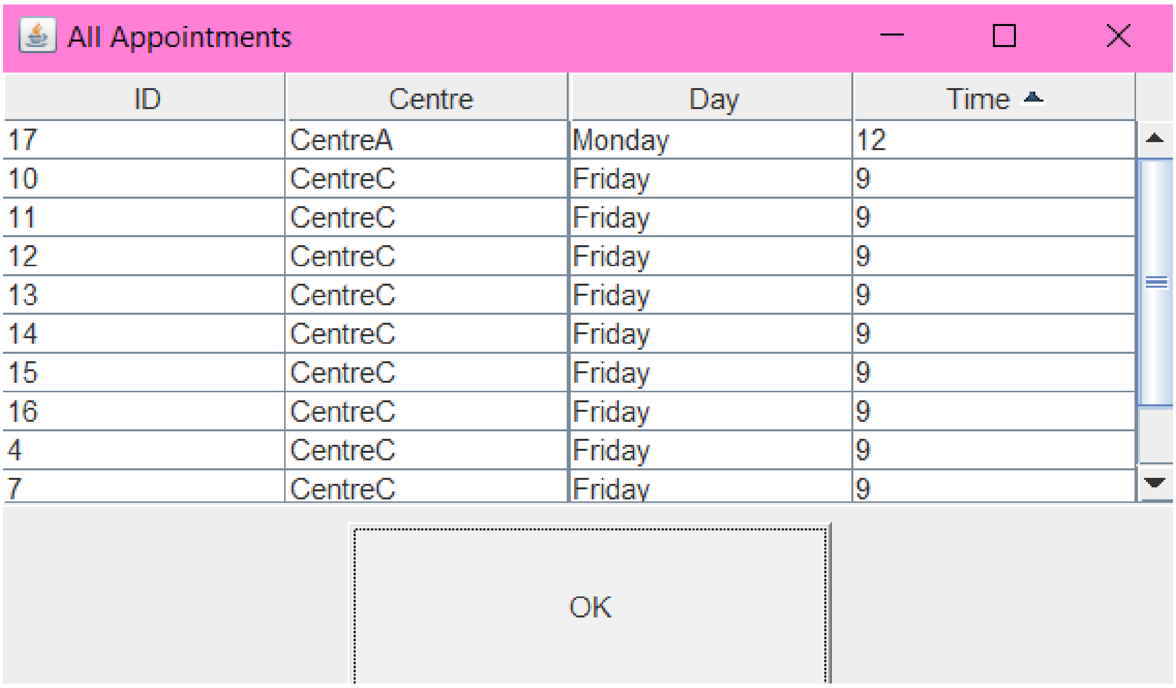
To set the column names an array containing String value of the column name is needed. A double dimension array is needed to initialize and store the value that will be shown in the JTable. To make the JTable noneditable, setEnabled should be false. Also, to disable row reordering, first we need to get the table header and setReorderingAllowed to false.

3.3.3 TableRowSorter



```
DefaultTableModel z = new DefaultTableModel(data, columnNames);
JTable z1 = new JTable(z);
z1.setEnabled(false);  //Disable Table Editing
z1.getTableHeader().setReorderingAllowed(false); //Disable Row Reordering

TableRowSorter sorter = new TableRowSorter(z1.getModel());//sort
z1.setRowSorter(sorter);
List<RowSorter.SortKey> sortKeys = new ArrayList<>();
int index = 0;
sortKeys.add(new RowSorter.SortKey(index,SortOrder.ASCENDING));
sorter.setSortKeys(sortKeys);
sorter.sort();
```

Figure 90 TableRowSorter sample code (Reference: (MadProgrammer, 2015))

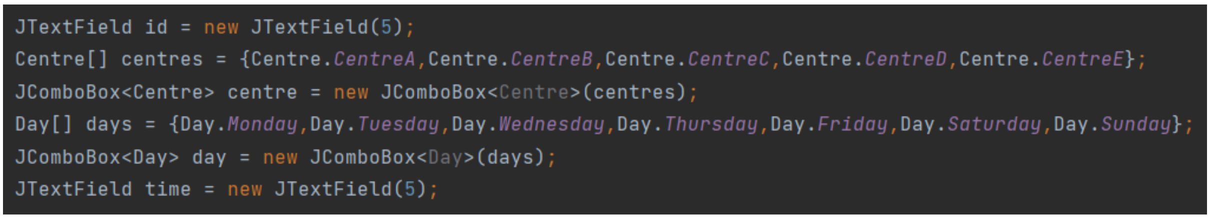


ID	Centre	Day	Time ▲
17	CentreA	Monday	12
10	CentreC	Friday	9
11	CentreC	Friday	9
12	CentreC	Friday	9
13	CentreC	Friday	9
14	CentreC	Friday	9
15	CentreC	Friday	9
16	CentreC	Friday	9
4	CentreC	Friday	9
7	CentreC	Friday	9

Figure 91 TableRowSorter output

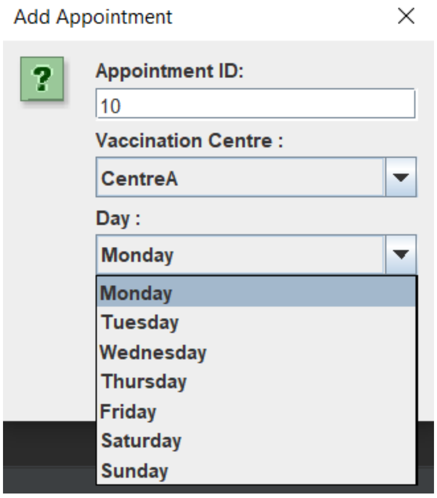
TableRowSorter is a component from the javax.swing.table package and it is useful since it can sort the table row in an ascending order or descending order. In this case, it was used to sort the Time, as can be seen by the Arrow beside the ‘Time’ column.

3.3.4 JComboBox



```
JTextField id = new JTextField(5);
Centre[] centres = {Centre.CentreA,Centre.CentreB,Centre.CentreC,Centre.CentreD,Centre.CentreE};
JComboBox<Centre> centre = new JComboBox<Centre>(centres);
Day[] days = {Day.Monday,Day.Tuesday,Day.Wednesday,Day.Thursday,Day.Friday,Day.Saturday,Day.Sunday};
JComboBox<Day> day = new JComboBox<Day>(days);
JTextField time = new JTextField(5);
```

Figure 92 JComboBox sample code



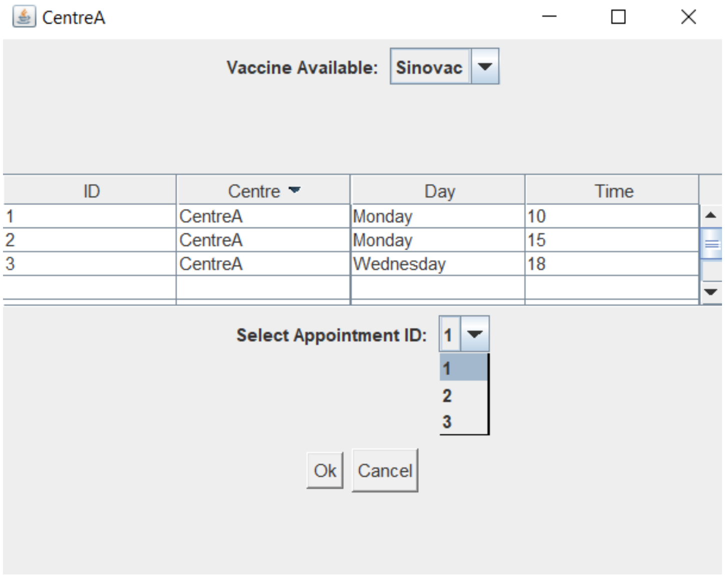
Add Appointment

Appointment ID: 10

Vaccination Centre : CentreA

Day : Monday

Figure 93 JComboBox implemented in Admin Register Appointment output



Vaccine Available: Sinovac

ID	Centre	Day	Time
1	CentreA	Monday	10
2	CentreA	Monday	15
3	CentreA	Wednesday	18

Select Appointment ID: 1

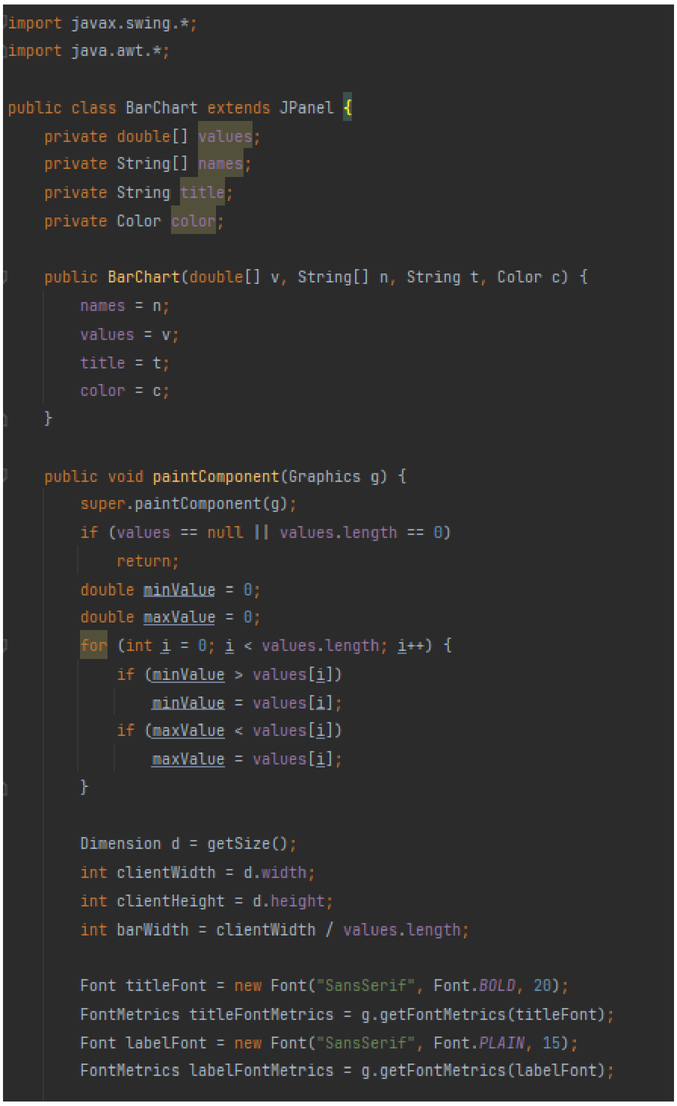
1
2
3

Figure 94 JComboBox implemented in People Make Appointment output

JComboBox is a Swing component that generates a drop-down list where user can select one value from the list. In this program it was used to show most of the enum values which includes Centre and Day. It was also used to show the selection of vaccine and ID user can pick in the Appointment Make method.

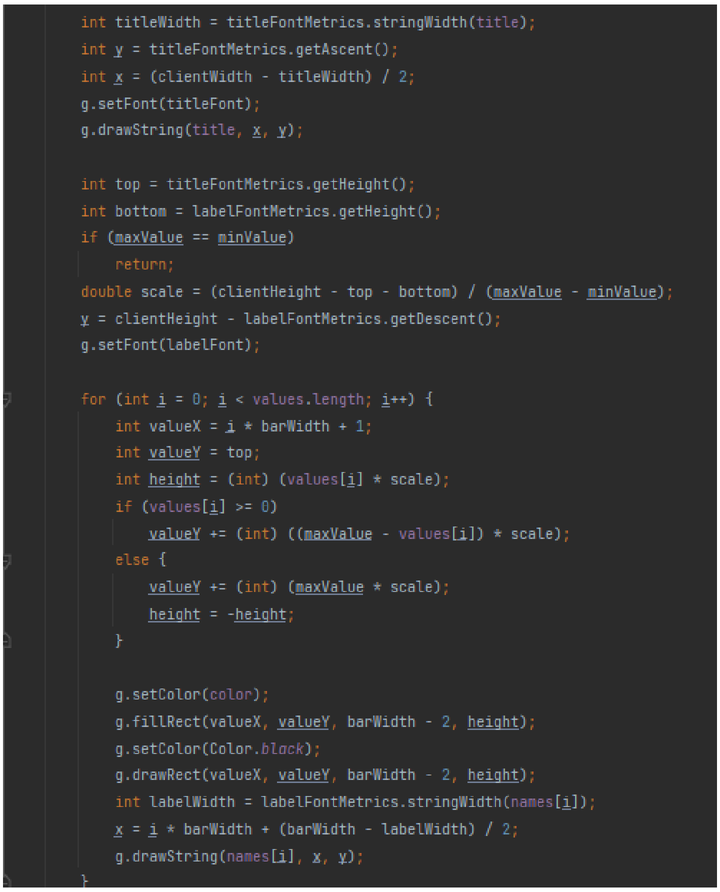
An array list is needed to store all the value that will be shown in the JComboBox. After that JComboBox can be added and the value set will be the name of array list made earlier.

3.3.5 BarChart



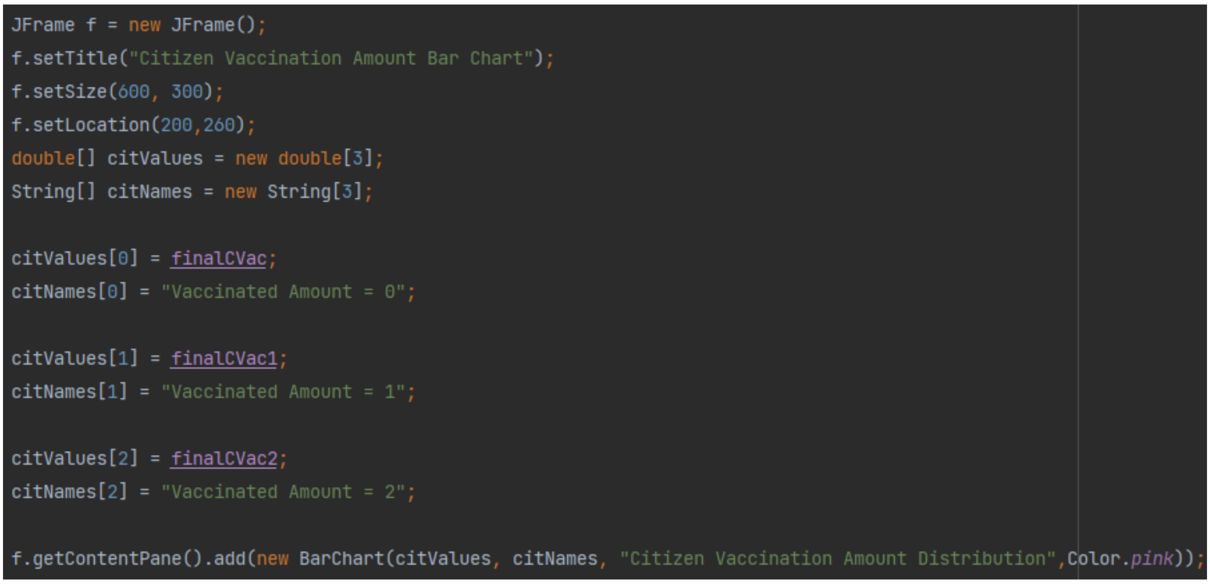
```java
import javax.swing.*;
import java.awt.*;

public class BarChart extends JPanel {
    private double[] values;
    private String[] names;
    private String title;
    private Color color;

    public BarChart(double[] v, String[] n, String t, Color c) {
        names = n;
        values = v;
        title = t;
        color = c;
    }

    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        if (values == null || values.length == 0)
            return;
        double minValue = 0;
        double maxValue = 0;
        for (int i = 0; i < values.length; i++) {
            if (minValue > values[i])
                minValue = values[i];
            if (maxValue < values[i])
                maxValue = values[i];
        }

        Dimension d = getSize();
        int clientWidth = d.width;
        int clientHeight = d.height;
        int barWidth = clientWidth / values.length;

        Font titleFont = new Font("SansSerif", Font.BOLD, 20);
        FontMetrics titleFontMetrics = g.getFontMetrics(titleFont);
        Font labelFont = new Font("SansSerif", Font.PLAIN, 15);
        FontMetrics labelFontMetrics = g.getFontMetrics(labelFont);
```

Figure 95 BarChart Class (1)



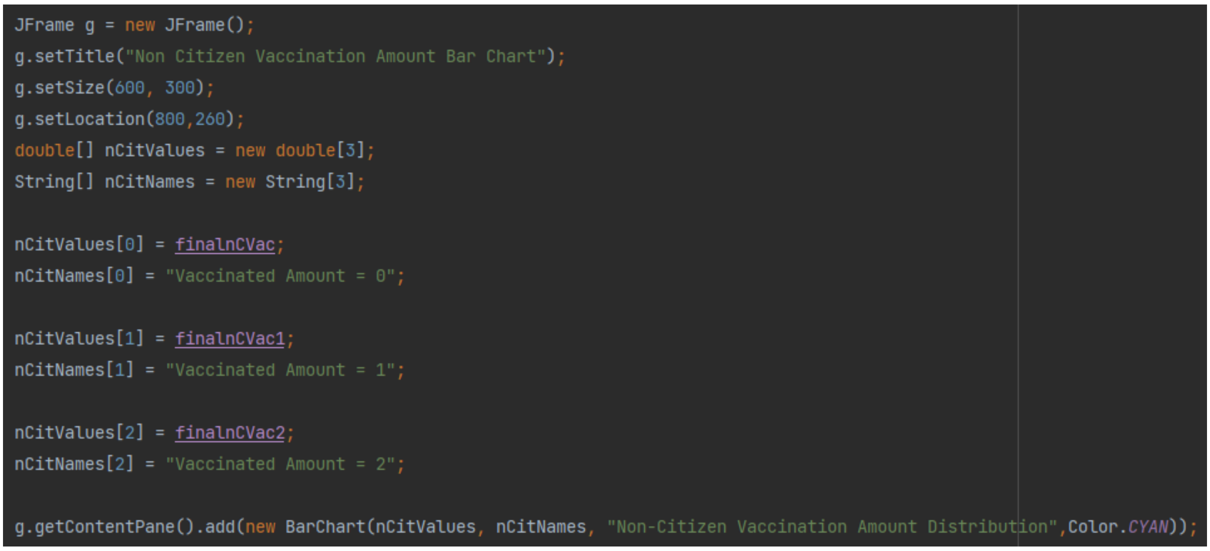
```
int titleWidth = titleFontMetrics.stringWidth(title);
int y = titleFontMetrics.getAscent();
int x = (clientWidth - titleWidth) / 2;
g.setFont(titleFont);
g.drawString(title, x, y);

int top = titleFontMetrics.getHeight();
int bottom = labelFontMetrics.getHeight();
if (maxValue == minValue)
    return;
double scale = (clientHeight - top - bottom) / (maxValue - minValue);
y = clientHeight - labelFontMetrics.getDescent();
g.setFont(labelFont);

for (int i = 0; i < values.length; i++) {
    int valueX = i * barWidth + 1;
    int valueY = top;
    int height = (int) (values[i] * scale);
    if (values[i] >= 0)
        valueY += (int) ((maxValue - values[i]) * scale);
    else {
        valueY += (int) (maxValue * scale);
        height = -height;
    }

    g.setColor(color);
    g.fillRect(valueX, valueY, barWidth - 2, height);
    g.setColor(Color.black);
    g.drawRect(valueX, valueY, barWidth - 2, height);
    int labelWidth = labelFontMetrics.stringWidth(names[i]);
    x = i * barWidth + (barWidth - labelWidth) / 2;
    g.drawString(names[i], x, y);
}
```

Figure 96 BarChart Class (2)



```
JFrame f = new JFrame();
f.setTitle("Citizen Vaccination Amount Bar Chart");
f.setSize(600, 300);
f.setLocation(200,260);
double[] citValues = new double[3];
String[] citNames = new String[3];

citValues[0] = finalCVac;
citNames[0] = "Vaccinated Amount = 0";

citValues[1] = finalCVac1;
citNames[1] = "Vaccinated Amount = 1";

citValues[2] = finalCVac2;
citNames[2] = "Vaccinated Amount = 2";

f.getContentPane().add(new BarChart(citValues, citNames, "Citizen Vaccination Amount Distribution",Color.pink));
```

Figure 97 Creating BarChart for Citizen Vaccine Distribution



```
JFrame g = new JFrame();
g.setTitle("Non Citizen Vaccination Amount Bar Chart");
g.setSize(600, 300);
g.setLocation(800,260);
double[] nCitValues = new double[3];
String[] nCitNames = new String[3];

nCitValues[0] = finalnCVac;
nCitNames[0] = "Vaccinated Amount = 0";

nCitValues[1] = finalnCVac1;
nCitNames[1] = "Vaccinated Amount = 1";

nCitValues[2] = finalnCVac2;
nCitNames[2] = "Vaccinated Amount = 2";

g.getContentPane().add(new BarChart(nCitValues, nCitNames, "Non-Citizen Vaccination Amount Distribution",Color.CYAN));
```

Figure 98 Creating BarChart for NonCitizen Vaccine Distribution

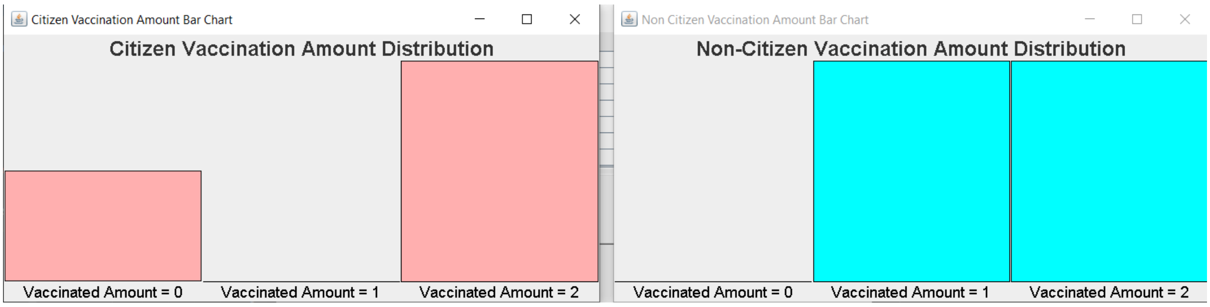

Citizen Vaccination Amount Bar Chart
Citizen Vaccination Amount Distribution
Vaccinated Amount = 0
Vaccinated Amount = 1
Vaccinated Amount = 2
Non Citizen Vaccination Amount Bar Chart
Non-Citizen Vaccination Amount Distribution
Vaccinated Amount = 0
Vaccinated Amount = 1
Vaccinated Amount = 2

Figure 99 Citizen & NonCitizen Vaccination Amount Distribution BarChart output

To generate a bar chart, a class for Bar Chart is created. The constructor consists of Title, Values, Labels, and Bar colour. A for loop is used to decide the Minimum Value and Maximum Value from the values used. Adjustments are made to set the scaling of the bar chart, title, and labels. Afterward, to generate the bar chart, set title, labels, and values, then assign and create a new object in the designated page.

4.0 References

Janssen, T. (2017, December 14). *OOP Concept for Beginners: What is Inheritance?* Retrieved from Stackify: https://stackify.com/oop-concept-inheritance/

java2s. (n.d.). *Simple bar chart*. Retrieved from java2s.com: http://www.java2s.com/Code/Java/2D-Graphics-GUI/Simplebarchart.htm

javaTpoint. (n.d.). *Java JTable*. Retrieved from javaTpoint: https://www.javatpoint.com/java-jtable

MadProgrammer. (2015, March 3). How to sort JTable in shortest way?

Oracle. (n.d.). *Class JPasswordField*. Retrieved from Oracle: https://docs.oracle.com/javase/7/docs/api/javax/swing/JPasswordField.html

Oracle. (n.d.). *The Java™ Tutorials*. Retrieved from A Visual Guide to Layout Managers: https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html

TechVidvan. (n.d.). *Abstraction in Java – Learn with its Types and Real-life Examples*. Retrieved from TechVidvan: https://techvidvan.com/tutorials/abstraction-in-java/

tutorialspoint. (n.d.). *Java - Encapsulation*. Retrieved from tutorialpoints: https://www.tutorialspoint.com/java/java_encapsulation.htm