30th January 2023

UNIVERSITY OF MAURITIUS

BSc Software Engineering

To: Mr. Dassen Sathan

Module: Software Verification and Validation

Students:

Geerna Peeharry - 2116307

Deesha Beerachee – 2117144

Sarah Veerapen Chetty - 2116427

Table of contents

1 Introduction……..4

1.1 Description of project……...4

1.2 System requirements……5

1.2.1 Functional requirements……6

1.2.2 Non-functional requirements……..6

1.3 Modules……8

1.4 Test Cases / Branch coverage……..9

1.4.1 Branch coverage for first name……9

1.4.2 Branch coverage for last name……11

1.4.3 Branch coverage for age……..12

1.4.4 Branch coverage for email address……..13

1.4.5 Branch coverage for user name……..14

1.4.6 Branch coverage for password……15

1.5 Condition coverage……17

1.5.1 Condition coverage for first name……17

1.5.2 Condition coverage for last name……18

1.5.3 Condition coverage for age……19

1.5.4 Condition coverage for email address……20

1.5.5 Condition coverage for user name……..21

1.5.6 Condition coverage for password……22

2.0 Labelling of code……23

2.1 Control flow graph……30

2.1.1 Cyclomatic complexity……..32

2.1.2 Independent path……32

2.1.3 Test data for paths……34

3.0 Sample screenshots……..37

4.0 Appendix……43

1 Introduction

1.1 Description of project

In just a few seconds, this system will enable a customer to book an appointment at Nails&Makeup. The salon offers different services such as manicures, pedicures, and make-up. The services will be allocated on different days, for instance, manicures will be done only on Mondays, Thursdays, and Saturdays, pedicures on Wednesdays and Sundays, and make-up on Tuesdays and Fridays. The list of days and services will be available on a service option. A list of options will be available for each category and the customer will have to choose one or more depending on their liking. After selection, the system will prompt the user to confirm the booking. After confirmation, an email will be sent to the customer marking the confirmation of the booking.

The project has the following functionalities:

- Menu Screen
- Log in, Sign Up or Exit
- Choose Service
- Choose day
- Booking
- Exit system

1.2 System requirements

Performance:

1) The system should provide a response in about 5 to 10 seconds.
2) The system shall be able to load in 1 to 2 seconds of opening or clicking on something.

Usability:

1) The system shall be user friendly so that users feel comfortable using it.
2) The system shall be responsive enough to accommodate 20 users at a time.

Security:

1) The user should get an error message if their username or password is invalid.
2) The system shall be able to check who has access to the system by demanding a password for security purposes as well as a specification of not less than 7 characters.
3) The system shall be able to secure any transaction involved between the customer and the company

Availability:

1) The system shall be available 24 hours a day, 7 days a week.
2) The system shall be able to work in less than 24 hours if the system crashes.

Ethical:

1) The system shall act ethically by helping the user but not make any choice for the user.

1.2.1 Functional requirements

FR1: The system shall prompt the user to enter her first name.

FR2: The system shall prompt the user to enter her last name.

FR3: The system shall prompt the user to enter her age.

FR4: The system shall prompt the user to enter her email address.

FR5: The system shall prompt the user to enter her user name.

FR6: The system shall prompt the user to enter her password.

FR7: The system shall prompt the user to enter choice.

FR8: The system shall prompt the user to enter day.

FR9: The system shall display error messages when appropriate.

1.2.2 Non – Functional requirements

NFR1: The system shall accept first names with length between 3 to 30 characters and contains only [a-z] and [A-Z].

NFR2: The system shall accept last names with length between 3 to 30 characters and contains only [a-z] and [A-Z].

NFR3: The system shall accept ages greater than 18.

NFR4: The system shall accept email address that contain “@” and “.”.

NFR5: The system shall accept user names with length 9 to 20 characters and starts with an uppercase letter.

NFR6: The system shall accept passwords with length 8 – 14 characters and containing one of #, _, &.

NFR7: The system shall accept only 1, 2 or 3 for choice.

NFR8: The system shall accept Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday for day.

NFR9: The system shall be written using the Java programming language.

NFR10: The system should be portable, meaning it can be implemented on various operating systems.

NFR11: The system shall allow access to multiple users simultaneously.

NFR12: The system shall has an interface that is user friendly.

NFR13: The system shall respond within 2 to 3 seconds.

NFR14: The system shall be operational 24 hours, 7 days.

1.3 Modules

There are 3 types of modules in this phase namely:

- User Module
- System Module
- Administrator Module

The functionality for each module is provided below:

User Module

The user should be able to login in the application to make her appointment for the service being provided such as pedicure, manicure and facial.

Booking should be confirmed for a particular date according to the user.

System Module

The selection to book for a particular appointment should be done by the user according to their needs. The booking should display immediately after the client has already selected what she wants to do.

Administration Module

The administrator should be able to change the different time and date in case of emergency. The administrator should inform when there is discount on the price of the service provided especially during festive mood.

1.4 Test cases

1. 4 Branch coverage

1.4.1 Branch coverage for First name

```
package nails_and_makeup;
import java.util.Scanner;

public class SignUp {

	Scanner in = new Scanner (System.in);
	public void signup ()
	{
	String fname = new String ();
	String lname = new String ();
	String email = new String ();
	String password = new String();
	int age;
	//Prompt for first name
	System.out.println "Enter your Firstname");
	//Input first name
	fname=in.nextLine();
```

Branch 1

```
	 while ( !(isAlphabet(fname)))// is used to check if the characters
in the first name are valid and are all letters
	{
	 System.out.println "Invalid Characters! Use a-z or A-Z for
firstname");//Display error message
	// Continuous prompt for valid characters
	System.out.println "Enter your Firstname");
	//input first name
	fname=in.nextLine();


	}
	// if the characters are all alphabets then check the first name range
if it is between 3-30
```

Branch 2

```
while (fname.length()<3||fname.length()>30)
      {
      //if it is not in accepted range,
      System.out.println "Firstname should be between 3-30 characters");//
an error message is displayed
      // Continuous prompt using the while loop
      System.out.println "Enter your Firstname");



      //input first name within the range for the loop to break
      fname=in.nextLine();
```

Correct format of First name:

- Should contain only [a-z] and [A-Z]
- Length between 3-30

Test Data for First Name:

Test case ID	Test data	Branch 1	Branch 2	Output
1	Ke	True	False	Invalid
2	Keiva	True	True	Valid

1.4.2 Branch coverage for Last name

```
//Prompt for Last name input
      System.out.println "Enter your Lastname";
      //Input last name
      lname= in.nextLine();
```

Branch 1

```
while (!(isAlphabet(lname)))// is used to check if the characters in the
last name are valid and are all letters
      {
      System.out.println "Invalid Characters! Use a-z or A-Z for
lastname";//Display error message
      // Continuous prompt for valid characters
      System.out.println "Enter your Lastname";
      //input last name
      lname= in.nextLine();
```

```
// if the characters are all alphabets then check the last name range if it
is between 3-30
```

Branch 2

```
      while (lname.length()<3||lname.length()>30)
      //if it is not in accepted range,
      System.out.println "Lastname should be between 3-30 characters";//
an error message is displayed
      // Continuous prompt using the while loop
      System.out.println "Enter your Lastname";
      //input last name within the range for the loop to break
      lname= in.nextLine();
      }
```

Correct format of Last name:

- Should contain only [a-z] and [A-Z]
- Length between 3-30

Test Data for Last name:

Test case ID	Test data	Branch 1	Branch 2	Output
1	Ch5	False	True	Invalid
2	Charlotte	True	True	Valid

1.4.3 Branch coverage for age

```
// Prompt to enter age
      System.out.println "Enter your age";
      //Input age
      age = in.nextInt();
```

Branch 1

```
while (age <17|| age > 100) // check if age < 17
      {
      // if yes
      // System.out.println("Age should be greater than 17 years old!
");// Display Error Message
      //Continuous Prompt for an age > 17
      System.out.println "Enter age that is 18 years old or above";
      // Input age
      age = in.nextInt();
```

Correct format of Age

- Age should be 18 or greater
- Numeric

Test Data for Age:

Test case ID	Test data	Branch 1	Output
1	10	False	Invalid
2	25	True	Valid

1.4.4 Branch coverage for Email address

```
System.out.println "Enter your Email Address");
      email=in.next();
```

Branch 1

```
while (!email.contains(".") || !email.contains("@"))//loop to check if the
mail contains '.' and '@'
      // if not then it goes into the loop
      {
      System.out.println "Invalid Email");//Display error message
      //Continuous Prompt for valid mail
      System.out.println "Enter a Valid Email Address");
      //input valid mail
      email=in.next();
```

Correct format of Email address

- Alphanumeric followed by a character “@” follow by alphabets followed by a dot “.”
- Followed by a valid domain

Test Data for Email address:

Test case ID	Test data	Branch 1	Output
1	keivagmailcom	False	Invalid
2	keiva@gmailcom	False	Invalid
3	Keivagmail.com	False	Invalid
4	keiva@gmail.com	True	Valid

1.4.5 Branch coverage for User name

```
System.out.println "Enter Username");
      username = in.nextLine();
      //System.out.println("Successful");
      // check the user name length and continue to prompt if it is outside
the accepted range
```

Branch 1

```
while (!(isUpper(username)))// is used to check if the characters in the
username are valid and are all letters
{
System.out.println "First Character should be of UPPERCASE");//Display
error message
// Continuous prompt for valid characters
System.out.println "Enter a valid username");
//input username
username= in.next();
}
```

Branch 2

```
while (username.length() <9 || username.length()>20)
      {
      System.out.println "Username should be between 9 to 20 characters");
      System.out.println "Enter your username" ;
      username = in.nextLine();
```

Correct format of User name:

- First alphabet in uppercase Branch 1
- Length should be between 9 to 20 branch 2

Test Data for User name:

Test case ID	Test data	Branch 1	Branch 2	Output
1	keiva	False	False	Invalid
2	Keivaaaaaaaa	True	True	Valid
3	Keivaaaaaaaaaaaaaaaa	True	False	Invalid
4	keivaaaaaaaa	False	True	Invalid

1.4.6 Branch coverage for password

```
System.out.println "Enter Password");
      password = in.nextLine();
```

Branch 1

```
while (password.length() <8 || password.length()>14)
      {
      System.out.println "Password should be between 8 to 14 characters");
      System.out.println "Enter a valid password");
      password = in.nextLine();
```

Branch 2

```
while (!isUppertotal(password))
      {
      System.out.println "Password should contain at least 1 UPPERCASE");
      System.out.println "Enter a valid password");
      password = in.nextLine();
      }
```

Branch 3

```
while (!isspecial(password))
      {
      System.out.println "Password should contain at least 1 special
character('#','_', '&')");
      System.out.println "Enter a valid password");
      password = in.nextLine();
      }
```

Correct format of password:

- Length should be 8 to 14.
- Must contain at least one uppercase letter
- Alphabetical, must contain a special character (#, _, &)

Test data for password:

Test case ID	Test data	Branch 1	Branch 2	Branch 3	Output
1	charlotte	True	False	False	Invalid
2	Char	False	True	False	Invalid

3	Char#	False	True	True	Invalid
4	chArloooooooootte_	False	True	True	Invalid
5	KeivaCharlot&	True	True	True	Valid

1.5 Condition coverage

1.5.1 Condition coverage for first name

```
//Prompt for first name
      System.out.println "Enter your Firstname" ;
      //Input first name
      fname=in.nextLine();


           Condition 1
while ( !(isAlphabet(fname))// is used to check if the characters in the
first name are valid and are all letters
      {
      System.out.println "Invalid Characters! Use a-z or A-Z for
firstname");//Display error message
      // Continuous prompt for valid characters
      System.out.println "Enter your Firstname" ;
      //input first name
      fname=in.nextLine();


      // if the characters are all alphabets then check the first name range
if it is between 3-30
                Condition 2        Condition 3
      while (fname.length()<3||fname.length()>30)
      {
      //if it is not in accepted range,
      System.out.println "Firstname should be between 3-30 characters");//
an error message is displayed
      // Continuous prompt using the while loop
      System.out.println "Enter your Firstname" ;
      //input first name within the range for the loop to break
      fname=in.nextLine();
      }     fname=in.nextLine();
      }
```

Test Data for First Name:

Test case ID	Test data	Condition 1	Condition 2	Condition 3
1	Ki	True	True	False
2	Ke1vaaaaaaaaaaaaaaaaaaaaaaaaaa	False	False	True

1.5.2 Condition coverage for last name

```
//Prompt for Last name input
      System.out.println "Enter your Lastname";
      //Input last name
      lname= in.nextLine();



            Condition 1
      while (!(isAlphabet(lname)))// is used to check if the characters in
the last name are valid and are all letters
      {
      System.out.println  "Invalid Characters! Use a-z or A-Z for
lastname");//Display error message
      // Continuous prompt for valid characters
      System.out.println "Enter your Lastname";
      //input last name
      lname= in.nextLine();
      }

// if the characters are all alphabets then check the last name range if it
is between 3-30

        Condition 2          Condition 3
while (lname.length()<3||lname.length()>30)
      {
      //if it is not in accepted range,
      System.out.println "Lastname should be between 3-30 characters";// an
error message is displayed
      // Continuous prompt using the while loop
      System.out.println "Enter your Lastname";
      //input last name within the range for the loop to break
      lname= in.nextLine();
      }
```

Test Data for Last Name:

Test case ID	Test data	Condition 1	Condition 2	Condition 3
1	Ch	True	True	False
2	chAr1ooooooooottteeeeeeeeeeeeee	False	False	True

1.5.3 Condition coverage for age

```
// Prompt to enter age
      System.out.println "Enter your age";
      //Input age
      age = in.nextInt();


          Condition 1   Condition 2
      while (age <17|| age > 100  // check if age < 17
      {
      // if yes
      // System.out.println("Age should be greater than 17 years old! ");//
Display Error Message
      //Continuous Prompt for an age > 17
      System.out.println "Enter age that is 18 years old or above";
      // Input age
      age = in.nextInt();
```

Test Data for Age:

Test case ID	Test data	Condition 1	Condition 2
1	17	False	False
2	110	false	True

1.5.4 Condition coverage of email address

```
      System.out.println("Enter your Email Address");
      email=in.next();

                  Condition 1                Condition 2
      while (!email.contains(".") || !email.contains("@"))//loop to check if
the mail contains '.' and '@'
      // if not then it goes into the loop
      {
      System.out.println("Invalid Email");//Display error message
      //Continuous Prompt for valid mail
      System.out.println("Enter a Valid Email Address");
      //input valid mail
      email=in.next();
      //Break the loop
      //System.out.println("Sign Up Successful");
      }
      String username = new String();//Declare username variable
      System.out.println("\nEnter your username");
      username = in.nextLine();
```

Test Data for Email address:

Test case ID	Test data	Condition 1	Condition 2
1	Keivagmail.com	True	False
2	Keiva@gmailcom	False	True

1.5.5 Condition coverage of user name

```
String username = new String ();//Declare username variable
      System.out.println "\nEnter your username");
      username = in.nextLine();

                  Condition 1                Condition 2
      while (username.length() <9 || username.length()>20)
      {
      System.out.println "Username should be between 9 to 20 characters");
      System.out.println "Enter your username");
      username = in.nextLine();


      }          Condition 3
      while (!(isUpper(username)))// is used to check if the characters in
the username are valid and are all letters
      {
      System.out.println "First Character should be of UPPERCASE");//Display
error message
      // Continuous prompt for valid characters
      System.out.println "Enter a valid username");
      //input username
      username= in.nextLine();
      }
```

Test Data for User Name:

Test case ID	Test data	Condition 1	Condition 2	Condition 3
1	Keiva	True	False	True
2	keiiiiiiiiiiivaaaaaaaaaaaa	False	True	False
3	Keivachorlotte	False	False	True

1.5.6 Condition coverage for password

```
System.out.println "Enter Password");
      password = in.nextLine();
                  Condition 1                Condition 2
      while (password.length() <8 || password.length()>14)
      {
      System.out.println "Password should be between 8 to 14 characters");
      System.out.println "Enter a valid password");
      password = in.nextLine();

              Condition 3
      while (!isUppertotal(password))
      {
      System.out.println "Password should contain at least 1 UPPERCASE");
      System.out.println "Enter a valid password");
      password = in.nextLine();



      }
        Condition 4
while (!isspecial(password))
      {
      System.out.println "Password should contain at least 1 special
character('#','_', '&')");
      System.out.println "Enter a valid password");
      password = in.nextLine();
      }
```

Test case ID	Test data	Condition 1	Condition 2	Condition 3	Condition 4
1	Keivaa&	True	False	True	True
2	Keiiiivvaaaaaaa	False	True	True	False
3	keivaaaaa&	False	False	False	True
4	keivaaaaa	False	False	False	False

2.0 Labelling of code

Sign up

```
package nails_and_makeup;

import java.util.Scanner;

public class SignUp

{

Scanner in = new Scanner (System.in);

public void signup ()

{

1

 String fname = new String();

 String lname = new String();

 String email = new String();

 String password = new String();

 int age;



//Prompt for first name

System.out.println("Enter your Firstname");

//Input first name

fname=in.nextLine();

2

 while ( !(isAlphabet(fname)))
```

```
// is used to check if the characters in the first name are valid and are all letters

3

{

System.out.println("Invalid Characters! Use a-z or A-Z for firstname");//Display error message

// Continuous prompt for valid characters

System.out.println("Enter your Firstname");

//input first name



4

}



// if the characters are all alphabets then check the first name range if it is between 3-30

5

while (fname.length()<3||fname.length()>30)



{

6

//if it is not in accepted range,

System.out.println("Firstname should be between 3-30 characters");

// an error message is displayed

// Continuous prompt using the while loop

System.out.println("Enter your Firstname");

//input first name within the range for the loop to break

fname=in.nextLine();
```

```
7
}

//Prompt for Last name input

System.out.println("Enter your Lastname");

//Input last name

lname= in.nextLine();

// is used to check if the characters in the last name are valid and are all
letters

8

while (!(isAlphabet(lname)))


{

9

System.out.println("Invalid Characters! Use a-z or A-Z for
lastname");//Display error message

// Continuous prompt for valid characters

System.out.println("Enter your Lastname");

//input last name

lname= in.nextLine();


10
}

// if the characters are all alphabets then check the last name range if it
is between 3-30  11
while (lname.length()<3||lname.length()>30)
```

```
{

//if it is not in accepted range,

12

 System.out.println("Lastname should be between 3-30
 characters");// an error message is displayed

 // Continuous prompt using the while loop

 System.out.println("Enter your Lastname");

 //input last name within the range for the loop to break

 lname= in.nextLine();

13

 }



// Prompt to enter age

System.out.println("Enter your age");

//Input age

age = in.nextInt();

14

     while (age <17|| age > 100)        // check if age < 17

{

// if yes

// System.out.println("Age should be greater than 17 years old! ");// Display
Error Message

//Continuous Prompt for an age > 17
```

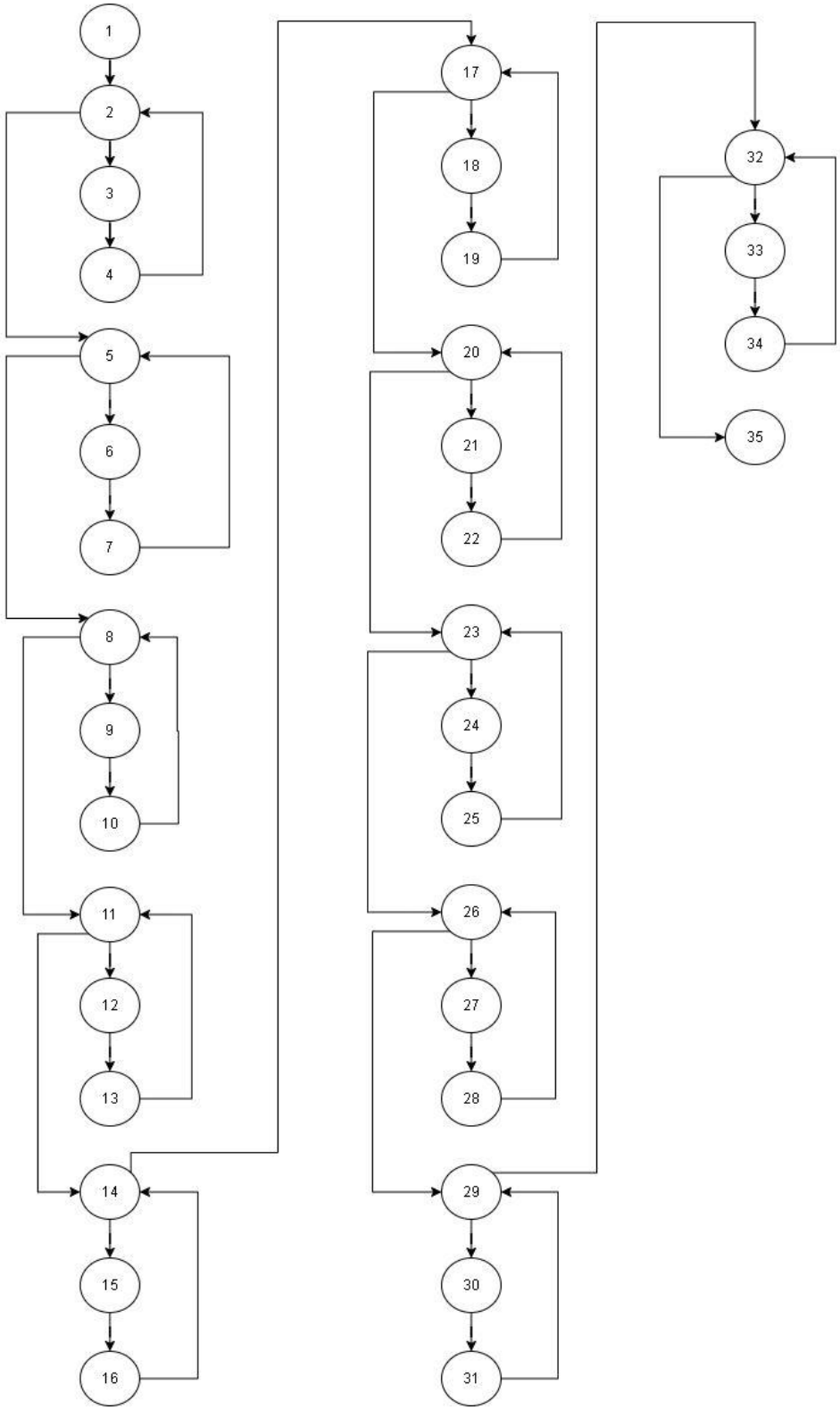
```
15
    System.out.println("Enter age that is 18 years old or
    above");
    // Input age
    _____
16
}

System.out.println("Enter your Email Address");
email=in.next();
//loop to check if the mail contains '.' and '@'    17
while (!email.contains(".") || !email.contains("@"))
// if not then it goes into the loop
{
18
    System.out.println("Invalid Email");//Display error message
    //Continuous Prompt for valid mail
    System.out.println("Enter a Valid Email Address");
    //input valid mail
    email=in.next();

//Break the loop
19
}
```

```
String username = new String ();//Declare username variable
System.out.println("\nEnter your username");
username = in.nextLine();
20
while (username.length() <9 || username.length()>20)

{ 21
System.out.println("Username should be between 9 to 20 characters");
System.out.println("Enter your username");
username = in.nextLine();
22
}
// is used to check if the characters in the username are valid and are all
letters    23
while (!(isUpper(username)))

{  24
//Display error message
System.out.println("First Character should be of UPPERCASE");
// Continuous prompt for valid characters
System.out.println("Enter a valid username");
//input username
username= in.nextLine();
```

```
25
}

System.out.println("Enter Password");
password = in.nextLine();
26
while (password.length() <8 || password.length()>14)
{
27
System.out.println("Password should be between 8 to 14 characters");
System.out.println("Enter a valid password");
password = in.nextLine();

28
}

29
while (!isUppertotal(password))
{
```

```
30

                System.out.println("Password should contain at least 1 UPPERCASE");

                System.out.println("Enter a valid password");

                password = in.nextLine();

31
            }
32
        while (!isspecial(password))

{ 33
                System.out.println("Password should contain at least 1 special
                character('#','_', '&')");

                System.out.println("Enter a valid password");

                password = in.nextLine();

34
            }

        }

        //otherwise it will return false

        return false;

        }

        }
```

2.1 Control flow graph



2.1.1 Cyclomatic complexity

Using no. of closed region V(G) = 12
Using no. of predicate nodes + 1 V(G) = D + 1 = 11 + 1 = 12
Using V(G) = e - n + 2 V(G) = e - n + 2 = 45 - 35 + 2 = 10 + 2 = 12

2.1.2 Independent path

Path 1: 1-2-5-8-11-14-17-20-23-26-29-32-35

Path 2: 1-2-3-4-2-5-8-11-14-17-20-23-26-29-32-35

Path 3: 1-2-3-4-2-5-6-7-5-8-11-14-17-20-23-26-29-32-35

Path 4: 1-2-3-4-2-5-6-7-5-8-9-10-8-11-14-17-20-23-26-29-32-35

Path 5: 1-2-3-4-2-5-6-7-5-8-9-10-8-11-12-13-11-14-17-20-23-26-29-32-35

Path 6: 1-2-3-4-2-5-6-7-5-8-9-10-8-11-12-13-11-14-15-16-14-17-20-23-26-29-32-35

Path 7: 1-2-3-4-2-5-6-7-5-8-9-10-8-11-12-13-11-14-15-16-14-17-18-19-17-20-23-26-29-32-35

Path 8: 1-2-3-4-2-5-6-7-5-8-9-10-8-11-12-13-11-14-15-16-14-17-18-19-17-20-21-22-20-23-26-29-32-35

Path 9: 1-2-3-4-2-5-6-7-5-8-9-10-8-11-12-13-11-14-15-16-14-17-18-19-17-20-21-22-20-23-24-25-23-26-29-32-35

Path 10: 1-2-3-4-2-5-6-7-5-8-9-10-8-11-12-13-11-14-15-16-14-17-18-19-17-20-21-22-20-23-24-25-23-26-27-28-26-29-32-35

Path 11: 1-2-3-4-2-5-6-7-5-8-9-10-8-11-12-13-11-14-15-16-14-17-18-19-17-20-21-22-20-23-24-25-23-26-27-28-26-29-30-31-29-32-35

Path 12: 1-2-3-4-2-5-6-7-5-8-9-10-8-11-12-13-11-14-15-16-14-17-18-19-17-20-21-22-20-23-24-25-23-26-27-28-26-29-30-31-29-32-33-34-32-35

2.1.3 Test data for paths

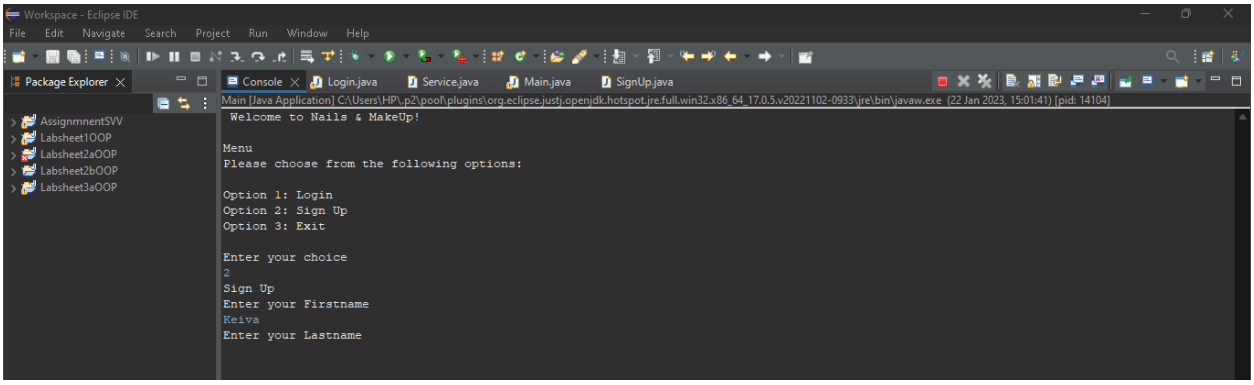
Test Case ID	Path	Test Data	Expected output	Output Validation
1	Path 1	First Name: ke Last Name: Ch5 Age: 10 Username: keiva Password: Char# Email Address: keivacharlottegmail.com	Sign up unsuccessful	Invalid
2	Path 2	First Name: ke Last Name: charlotte Age: 25 Username: keiva10 Password: keivaaaaaaaa& Email Address: keivacharlotte@gmail.com	Sign up unsuccessful	Invalid
3	Path 3	First Name: Keiva Last Name: Ch5 Age: 25 Username: keiva10 Password: Keivaaaaaaaa& Email Address: keivacharlotte@gmail.com	Sign up unsuccessful	Invalid
4	Path 4	First Name: keiva Last Name: charlotte Age: 10 Username: keiva10 Password: Keivaaaaaaaa& Email Address: keivacharlotte@gmail.com	Sign up unsuccessful	Invalid

5	Path 5	First Name: keiva Last Name:charlotte Age: 25 Username: keiva10 Password: Keivaaaaaaaa& Email Address: keivacharlottegmailcom	Sign up unsuccessful	Invalid
6	Path 6	First Name: keiva Last Name: charlotte Age: 25 Username: keiva Password: Keivaaaaaaaa& Email Address: keivacharlotte@gmail.com	Sign up unsuccessful	Invalid
7	Path 7	First Name: Keiva Last Name: Charlotte Age: 25 Username: Keiva10 Password: Char# Email Address: keivacharlotte@gmail.com	Sign up unsuccessful	Invalid
8	Path 8	First Name: Kei5 Last Name: Char5 Age :11 Username: keiva Password: keivaaaaaaaa& Email address: keivacharlottegmailcom	Sign up unsuccessful	Invalid
9	Path 9	First Name: keiva Last Name: Charlotte Age: 25 Username: keiva10 Password: keivaaaaaaaa&	Sign up unsuccessful	Invalid

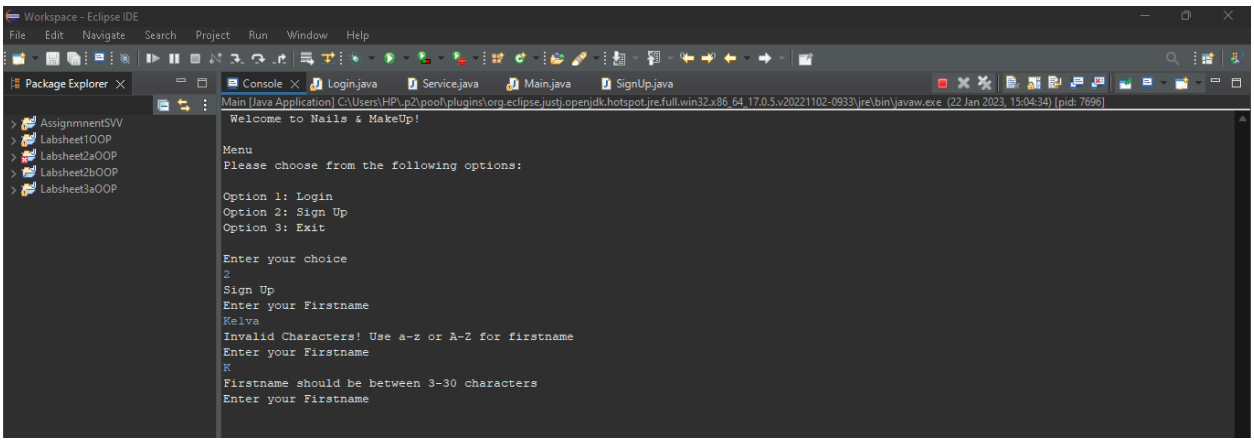
		Email Address: keivacharlotte@gmail.com		
10	Path 10	First Name: keiva Last Name: Charlotte Age: 25 Username: keiva10 Password: keivaaaaaaaa& Email Address: keivacharlotte@gmail.com	Sign up unsuccessful	Invalid
11	Path 11	First Name: keiva Last Name: Charlotte Age: 25 Username: keiva10 Password: Keivaaaaaaaa Email Address: keivacharlotte@gmail.com	Sign up unsuccessful	Invalid
12	Path 12	First Name: Keiva Last Name :Chorlotte Age:25 Username: Keivachar10 Password: Keivaaaaaaaa& Email address: keivacharlotte@gmail.com	Sign up Successful	Valid

3.0 Sample screenshots

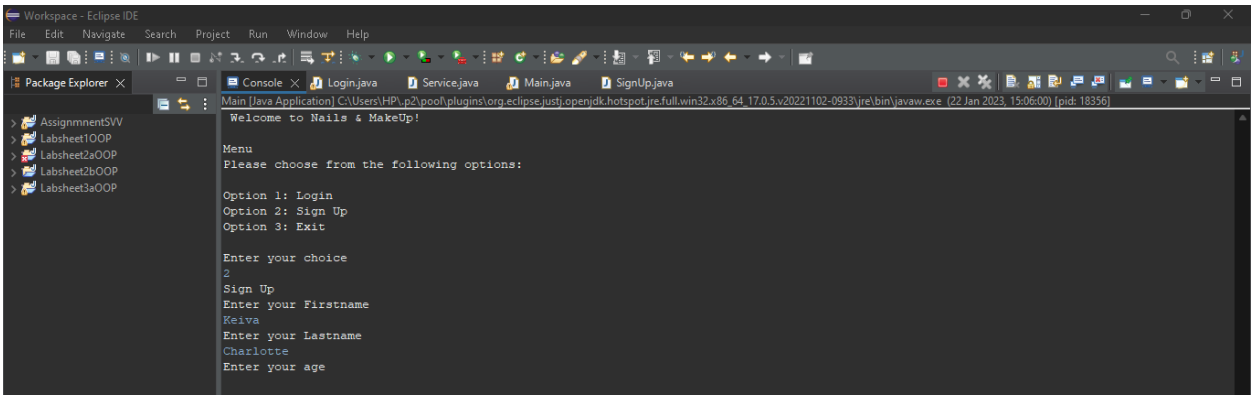
Valid first name



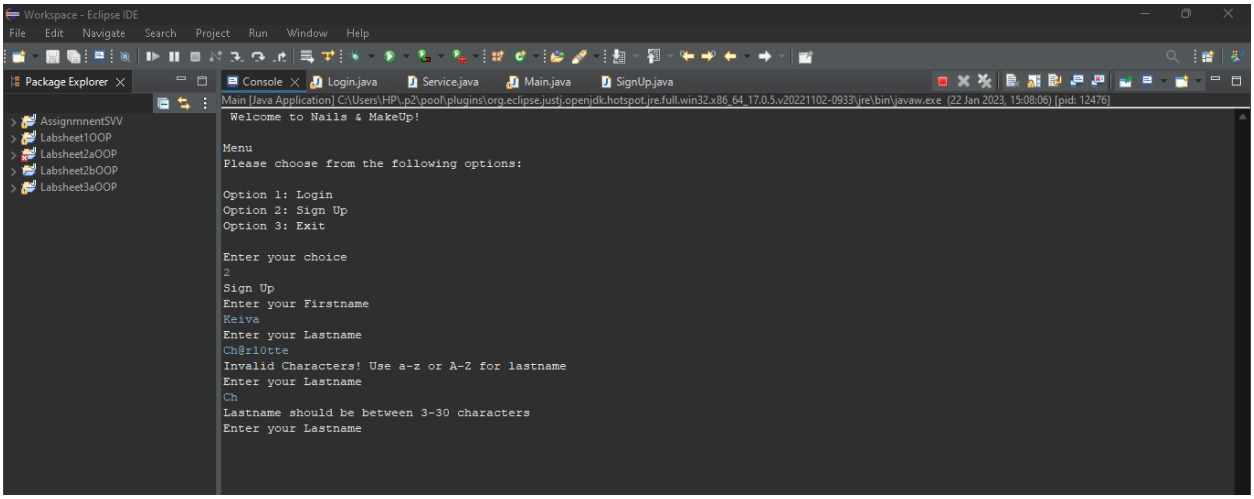
Invalid first name



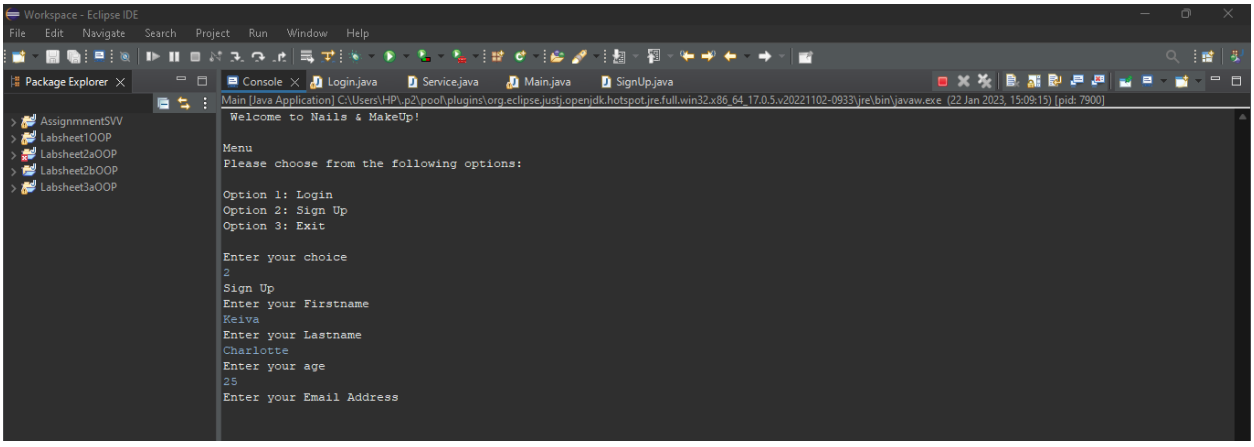
Valid last name



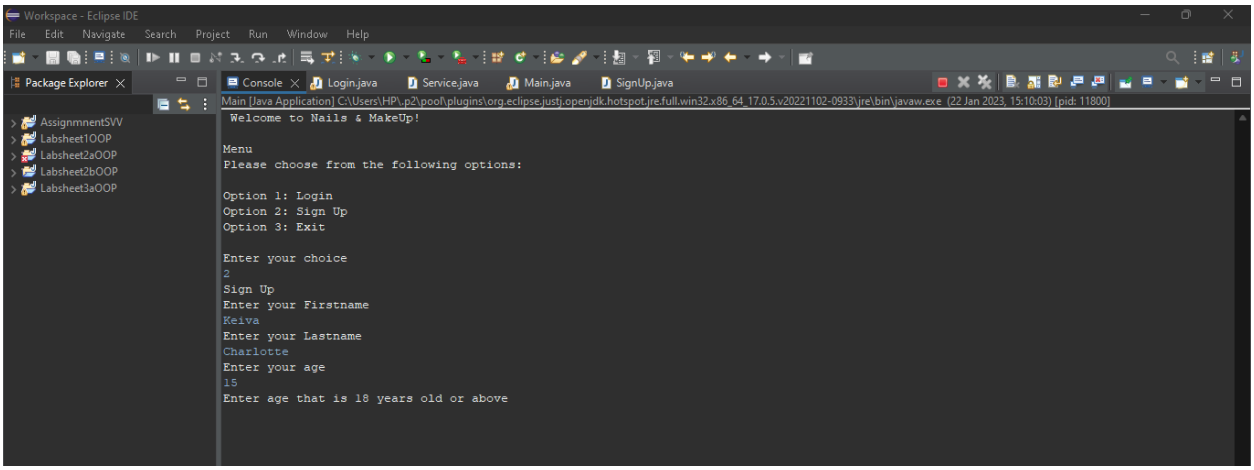
Invalid last name



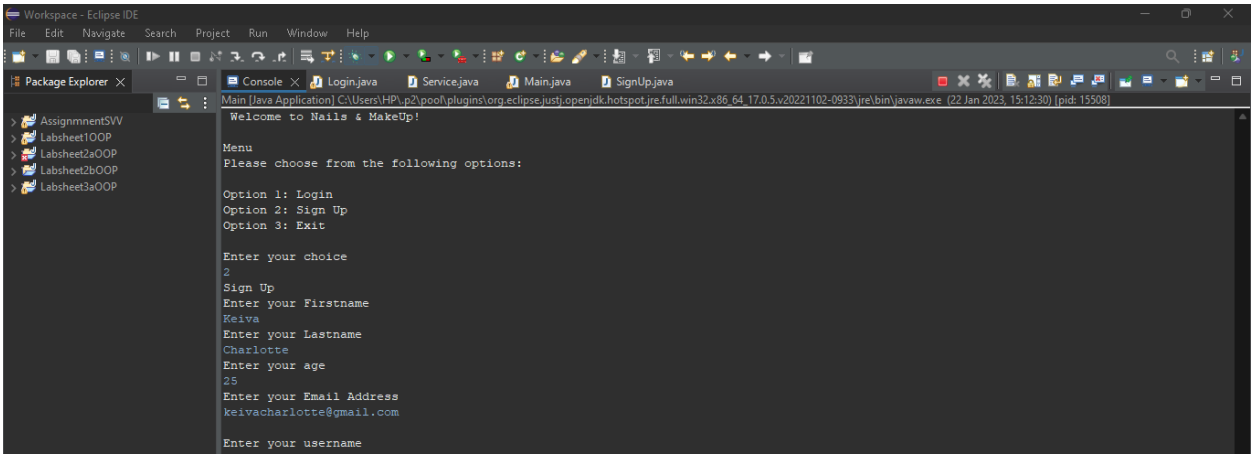
Valid age



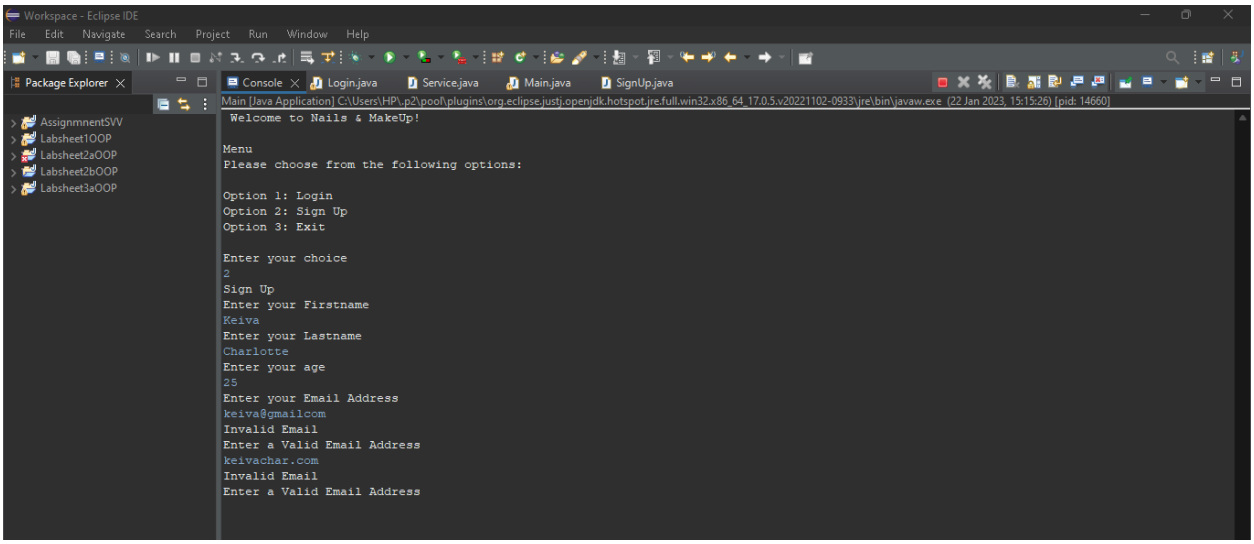
Invalid age



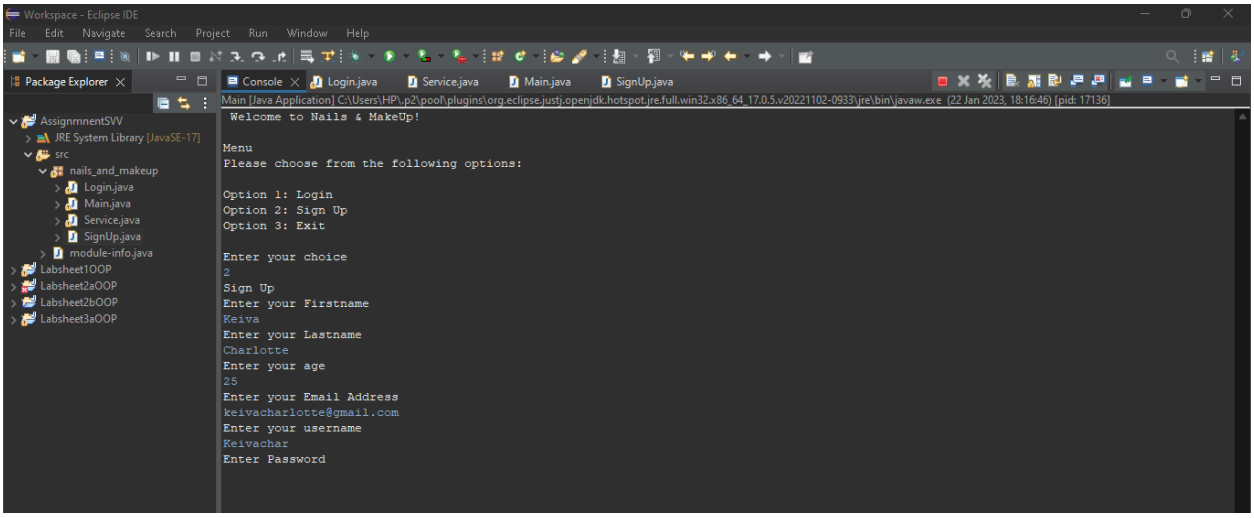
Valid email address



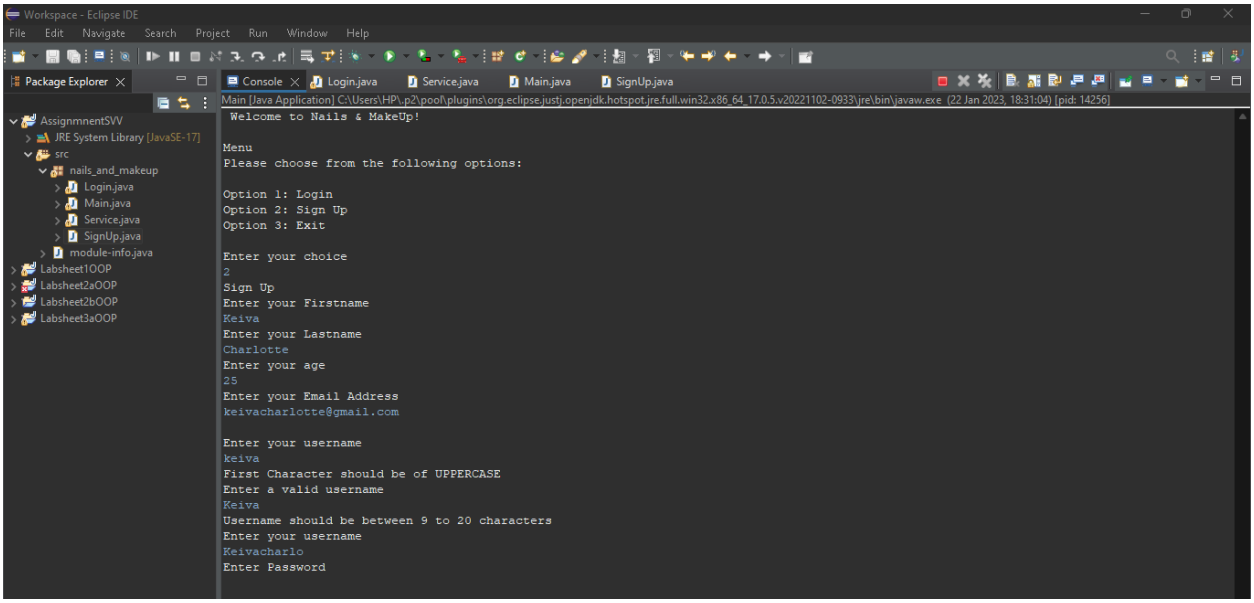
Invalid email address



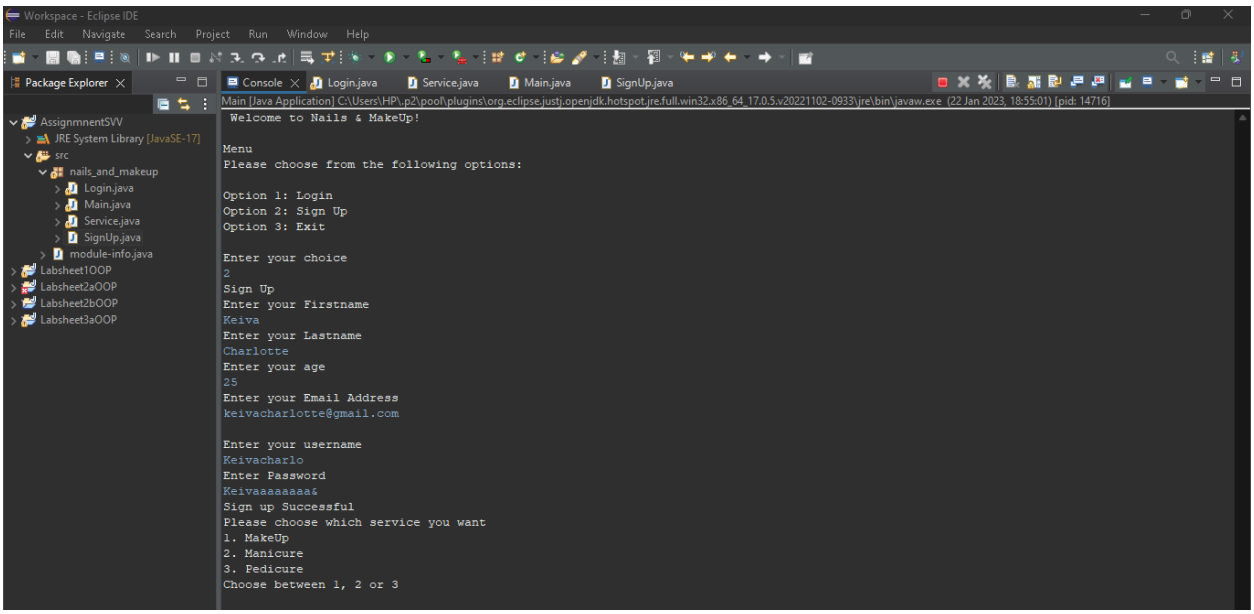
Valid user name



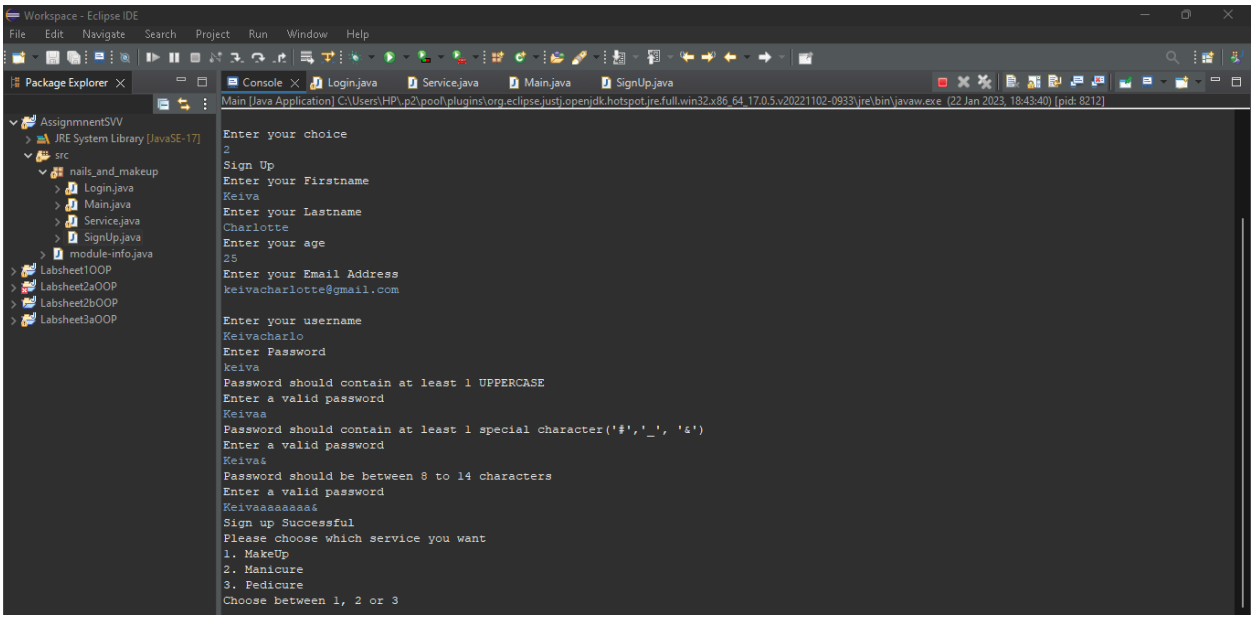
Invalid user name



Valid password



Invalid password



4.0 Appendix

Login

```
package nails_and_makeup;

import java.util.*;

public class Login

{

public void login ()

{

String username = new String();

String password = new String();

Scanner in = new Scanner(System.in);

System.out.println("Enter Username");

username = in.nextLine();

//System.out.println("Successful");

// check the user name length and continue to prompt if it is outside the
accepted range

while (username.length() <9 || username.length()>20)

{

System.out.println("Username should be between 9 to 20 characters");

System.out.println("Enter your username");

username = in.nextLine();

}

while (!(isUpper(username)))// is used to check if the characters in the user
name are valid and are all letters

{
```

```
System.out.println("First Character should be of UPPERCASE");//Display error
message

// Continuous prompt for valid characters

System.out.println("Enter a valid username");

//input user name

username= in.nextLine();

}

System.out.println("Enter Password");

password = in.nextLine();

while (password.length() <8 || password.length()>14)

{

System.out.println("Password should be between 8 to 14 characters");

System.out.println("Enter a valid password");

password = in.nextLine();

}

while (!isUppertotal(password))

{

System.out.println("Password should contain at least 1 UPPERCASE");

System.out.println("Enter a valid password");

password = in.nextLine();

}

while (!isspecial(password))

{

System.out.println("Password should contain at least 1 special
character('#','_', '&')");

System.out.println("Enter a valid password");

password = in.nextLine();
```

```
}

System.out.println("Login Successful");

}

// to test if the name are of letters only

public boolean isAlphabet (String name)

{

for (int i=0; i< name.length();i++ )// String name can be seen as array of
characters also hence it becomes an array

{

if (!Character.isLetter(name.charAt(i)/*Character at position i in the name
string*/))

/*is tested using the build in function Character.isLetter () and in this
case if all the characters are letters

* it returns true */

{

return false;

}

}

//otherwise it will return false

return true;

}

public boolean isUpper (String name)

{

for (int i=0; i< name.length();i++ )// String name can be seen as array of
characters also hence it becomes an array

{

if (!Character.isUpperCase(name.charAt(0)/*Character at position i in the
name string*/))
```

```
/*is tested using the build in function Character.isLetter () and in this
case if all the characters are letters

* it returns true */

{

return false;

}

}

//otherwise it will return false

return true;

}

public boolean isUppertotal (String name)

{

for (int i=0; i< name.length();i++ )// String name can be seen as array of
characters also hence it becomes an array

{

if (Character.isUpperCase(name.charAt(i)/*Character at position i in the name
string*/))

/*is tested using the build in function Character.isLetter () and in this
case if all the characters are letters

* it returns true */

{

return true;

}

}

//otherwise it will return false

return false;

}

public boolean isspecial (String name)
```

```
{

for (int i=0; i< name.length();i++ )// String name can be seen as array of
characters also hence it becomes an array

{

if (( name.charAt(i) == '#'|| name.charAt(i) == '_'|| name.charAt(i)=='&'))

{

return true;

}

}

//otherwise it will return false

return false;

}
```

Sign up

```
package nails_and_makeup;

import java.util.Scanner;

public class SignUp

{

Scanner in = new Scanner (System.in);

public void signup ()

{

String fname = new String();

String lname = new String();

String email = new String();

String password = new String();

int age;

//Prompt for first name

System.out.println("Enter your Firstname");

//Input first name

fname=in.nextLine();



while ( !(isAlphabet(fname)))// is used to check if the characters in the
first name are valid and are all letters

{

System.out.println("Invalid Characters! Use a-z or A-Z for
firstname");//Display error message

// Continuous prompt for valid characters

System.out.println("Enter your Firstname");

//input first name
```

```
fname=in.nextLine();



}

// if the characters are all alphabets then check the first name range if it
is between 3-30



while (fname.length()<3||fname.length()>30)

{

//if it is not in accepted range,

System.out.println("Firstname should be between 3-30 characters");// an error
message is displayed

// Continuous prompt using the while loop

System.out.println("Enter your Firstname");

//input first name within the range for the loop to break

fname=in.nextLine();



}

//Prompt for Last name input

System.out.println("Enter your Lastname");

//Input last name

lname= in.nextLine();



while (!(isAlphabet(lname)))// is used to check if the characters in the last
name are valid and are all letters

{

System.out.println("Invalid Characters! Use a-z or A-Z for
lastname");//Display error message

// Continuous prompt for valid characters
```

```
System.out.println("Enter your Lastname");

//input last name

lname= in.nextLine();



}

// if the characters are all alphabets then check the last name range if it
is between 3-30

while (lname.length()<3||lname.length()>30)

{

//if it is not in accepted range,

System.out.println("Lastname should be between 3-30 characters");// an error
message is displayed

// Continuous prompt using the while loop

System.out.println("Enter your Lastname");

//input last name within the range for the loop to break

lname= in.nextLine();



}

// Prompt to enter age

System.out.println("Enter your age");

//Input age

age = in.nextInt();



while (age <17|| age > 100) // check if age < 17

{

// if yes
```

```
// System.out.println("Age should be greater than 17 years old! ");// Display
Error Message

//Continuous Prompt for an age > 17

System.out.println("Enter age that is 18 years old or above");

// Input age

age = in.nextInt();



}

System.out.println("Enter your Email Address");

email=in.next();



while (!email.contains(".") || !email.contains("@"))//loop to check if the
mail contains '.' and '@'

// if not then it goes into the loop

{

System.out.println("Invalid Email");//Display error message

//Continuous Prompt for valid mail

System.out.println("Enter a Valid Email Address");

//input valid mail

email=in.next();

//Break the loop

//System.out.println("Sign Up Successful");

}



String username = new String ();//Declare username variable

System.out.println("\nEnter your username");

username = in.nextLine();
```

```
while (username.length() <9 || username.length()>20)

{

System.out.println("Username should be between 9 to 20 characters");

System.out.println("Enter your username");

username = in.nextLine();



}



while (!(isUpper(username)))// is used to check if the characters in the
username are valid and are all letters

{

System.out.println("First Character should be of UPPERCASE");//Display error
message

// Continuous prompt for valid characters

System.out.println("Enter a valid username");

//input username

username= in.nextLine();



}

System.out.println("Enter Password");

password = in.nextLine();



while (password.length() <8 || password.length()>14)

{

System.out.println("Password should be between 8 to 14 characters");

System.out.println("Enter a valid password");
```

```
password = in.nextLine();



}



while (!isUppertotal(password))

{

System.out.println("Password should contain at least 1 UPPERCASE");

System.out.println("Enter a valid password");

password = in.nextLine();



}



while (!isspecial(password))

{

System.out.println("Password should contain at least 1 special
character('#','_', '&')");

System.out.println("Enter a valid password");

password = in.nextLine();



}

System.out.println("Sign up Successful");

}

// to test if the name are of letters only

public boolean isAlphabet (String name)

{

for (int i=0; i< name.length();i++ )// String name can be seen as array of
characters also hence it becomes an array
```

```
{

if (!Character.isLetter(name.charAt(i)/*Character at position i in the name
string*/))

/*is tested using the build in function Character.isLetter () and in this
case if all the characters are letters

* it returns true */

{

return false;

}

}

//otherwise it will return false

return true;

}

public boolean isUpper (String name)

{

for (int i=0; i< name.length();i++ )// String name can be seen as array of
characters also hence it becomes an array

{

if (!Character.isUpperCase(name.charAt(0)/*Character at position i in the
name string*/))

/*is tested using the build in function Character.isLetter () and in this
case if all the characters are letters

* it returns true */

{

return false;

}

}

//otherwise it will return false
```

```
return true;

}

public boolean isUppertotal (String name)

{

for (int i=0; i< name.length();i++ )// String name can be seen as array of
characters also hence it becomes an array

{

if (Character.isUpperCase(name.charAt(i)/*Character at position i in the name
string*/))

/*is tested using the build in function Character.isLetter () and in this
case if all the characters are letters

* it returns true */

{

return true;

}

}

//otherwise it will return false

return false;

}

public boolean isspecial (String name)

{

for (int i=0; i< name.length();i++ )// String name can be seen as array of
characters also hence it becomes an array

{

if (( name.charAt(i) == '#'|| name.charAt(i) == '_'|| name.charAt(i)=='&'))

{

return true;

}
```

```
}

//otherwise it will return false

return false;

}

// Display Sign up or registration successful

}
```

Service

```
package nails_and_makeup;

import java.util.Scanner;

public class Service {

public void service(){

int choice=0;

String day = new String();

Scanner scanner = new Scanner(System.in);

System.out.println("Please choose which service you want");

System.out.println("1. MakeUp");

System.out.println("2. Manicure");

System.out.println("3. Pedicure");

System.out.println("Choose between 1, 2 or 3 ");

choice = scanner.nextInt();

while (choice<1 || choice>3)

{

System.out.print("Invalid Input. Please enter a value between 1 and 3 ");

choice = scanner.nextInt();

}
```

```
if (choice == 1)

{

System.out.println("MakeUp Selected");

System.out.println("Choose a day between Tuesday and Friday: ");

day = scanner.next();

while (!((day.toLowerCase().equals("tuesday")) ||
(day.toLowerCase().equals("friday"))))

{

System.out.println("Invalid day. Please select a day between Tuesday and
Friday ");

day = scanner.next();

}

System.out.println("Booking is confirmed for " + day.toLowerCase());

}

else if (choice == 2)

{

System.out.println("Manicure Selected");

System.out.println("Choose a day between Monday, Thursday or Saturday ");

day = scanner.next();

while (!((day.toLowerCase().equals("monday")) ||
day.toLowerCase().equals("thursday") ||
day.toLowerCase().equals("saturday")))

{

System.out.println("Invalid day. Please select a day between Monday, Thursday
or Saturday ");

day = scanner.next();

}

System.out.println("Booking is confirmed for " + day.toLowerCase());
```

```
}

else if (choice == 3)

{

System.out.println("Pedicure Selected");

System.out.println("Choose a day between Wednesday and Sunday ");

day = scanner.next();

while (!(day.toLowerCase().equals("wednesday") ||
day.toLowerCase().equals("sunday")))

{

System.out.println("Invalid day. Please select a day between Wednesday and
Sunday ");

day = scanner.next();

}

System.out.println("Booking is confirmed for " + day.toLowerCase());

}

}
```

Main

```
package nails_and_makeup;

import java.util.*;

public class Main

{

public static void main (String[] args)

{
```

```
Scanner in = new Scanner (System.in);

int choice = 0;

System.out.println(" Welcome to Nails & MakeUp!\n\nMenu\nPlease choose from
the following options: \n"

+ "\nOption 1: Login\nOption 2: Sign Up\nOption 3: Exit");

System.out.println("\nEnter your choice");

while (choice ==0)

{

choice= in.nextInt();

if (choice ==1 )

{

System.out.println("Login");

Login lg = new Login();

lg.login();

Service menu = new Service();

menu.service();

}

else

{

if (choice ==2)

{

System.out.println("Sign Up");

SignUp lg = new SignUp();

lg.signup();

Service menu = new Service();

menu.service();
```

```
}
else
{
if (choice ==3)
{
System.out.println("Exit Successful");
System.out.println("Have a great day/night! See you soon.");
}
else
{
System.out.println("Invalid option. Enter option number from the Menu");
System.out.println("Enter your choice");
choice = 0;
}
}
}
}
```