

CHAPTER 1

INTRODUCTION

What's your story is a project developed using the concept of object-oriented programming in real time and analysing and implementing the same using Java language. Many object-oriented programming features like class, objects, inheritance and encapsulation have been used. Many classes have been declared and defined which are called based on the needs of the user. This project is built after analysing and understanding the technological needs to develop an application.

1.1 PROBLEM STATEMENT

Developing an application that will be a destination for everyone to tell their life experiences and stories along with the option to post their technical and non-technical queries. This project is implemented using the concept of object-oriented programming in real time and analysing and implementing the same using Java language.

1.2 OBJECTIVES

Analyse and understand the technological needs to support tools and develop an application which will provide a platform for the societal needs to ask and answer the queries of any individual on all technical and non-technical aspects. It will also provide an outlet to people's emotions and provide a base for sharing their life stories. This project is designed to provide solutions and help understand the human nature of emotions.

1.3 EXPECTED OUTCOMES

The user will be able to either log in to their already existing account or sign up to create a new account after which they'll be able to log in. Once they are logged in, they will be presented with three options – posting an answer to the already existing questions, adding a

question to the question bank and browsing through the feed of already answered questions. The users will be able to choose one of the three options. If the user chooses to post a question, a list of already existing questions is displayed along with the text area to enter another question the user wishes to add. If the user chooses to post an answer, then the list of questions is displayed and the user can choose the question he/she wants to answer and then the option to answer that particular question is made available.

1.4 METHODOLOGY TO BE FOLLOWED

The concept of object-oriented programming is implemented in the development of this project. Concepts of classes, object, inheritance, exception handling, abstraction are used to come up with an effective and efficient solution of the given problem statement. Each functionality of the program is broken down to form multiple classes whose data members and member functions are accessed by other classes either by creating objects of the intended class or by inheriting the properties of the base class.

1.5 REQUIREMENT SPECIFICATIONS

1.5.1 Software Requirement

- Operating System
- Java Runtime Environment (JRE) or Java Development Kit (JDK) version 1.6 or higher.

1.5.2 Hardware Requirements

- 1 GB RAM (2 GB+ recommended)
- 9-58 GB of free hard disk space,
- Basic GPU
- Intel Pentium or compatible processor 1.6 GHz minimum (2GHz+ recommended)
- 1024x768 or higher resolution monitor.

CHAPTER 2

OBJECT ORIENTED CONCEPTS

Object-Oriented Programming or OOPs refers to languages that uses objects in programming. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism etc in programming. The main aim of Object-Oriented programming is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

2.1 CLASS

A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. In general, class declarations can include these components, in order:

- Access Modifiers: A class can have public, private or protected access.
- Class name: The name should begin with an initial letter (capitalized by convention) and it should not be a keyword.
- Superclass (if any): The name of the class's parent (superclass), if any, preceded by the keyword extends. A class (subclass) can only extend one parent.
- Interfaces (if any): A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.
- Body: The class body surrounded by braces, { }.

There are various types of classes that are used in real time applications such as nested classes, anonymous classes, lambda expressions.

2.2 OBJECT

It is a basic unit of Object-Oriented Programming and represents the real-life entities. A typical Java program creates many objects, interact by invoking methods. An object is an instance of a class. An object consists of:

- **State:** It is represented by attributes of an object. It also reflects the properties of an object.
- **Behaviour:** It is represented by methods of an object. It also reflects the response of an object with other objects.
- **Identity:** It gives a unique name to an object and enables one object to interact with other objects.

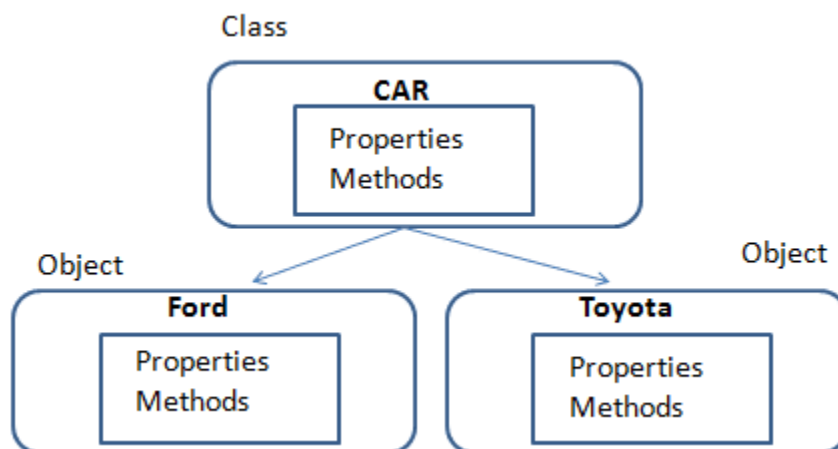


Fig 2.1: Diagrammatic representation of class and object.

In the above figure, the class name is CAR whose data members are – Properties, Methods. Two objects of the class have been created which have their own values for the data members specified. An instance of a class is created every time we declare its object. The only way to access the non-static data members of a class is via its objects. The static data members can be accessed directly without creating an object.

2.3 INHERITANCE

Inheritance is an important pillar of OOP (Object Oriented Programming). It is the mechanism in java by which one class is allow to inherit the features (data members and member functions) of another class. The keyword used for inheritance is extends.

- **Super Class:** The class whose features are inherited is known as super class (or a base class or a parent class).
- **Sub Class:** The class that inherits the other class is known as sub-class (or a derived class, extended class, or child class). The sub-class can add its own data members and member functions in addition to the super-class data members and member functions.
- **Reusability:** Inheritance supports the concept of reusability, i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the data members and member functions of the existing class.

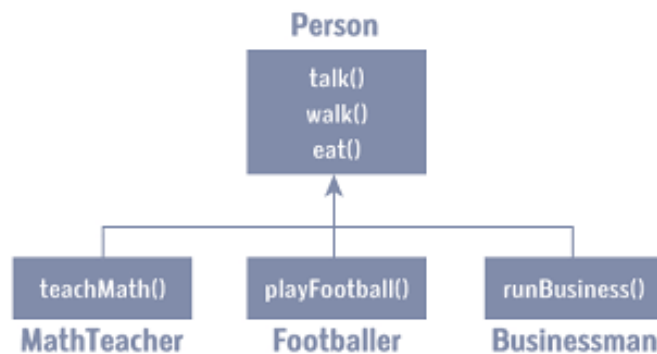


Fig 2.2: Diagrammatic representation of inheritance.

In the above figure, the base class is Person with member functions – `talk()`, `walk()`, `eat()`. The three sub classes inherit these properties from the base class and have their own additional features that are added.

2.4 POLYMORPHISM

Polymorphism is considered as one of the important features of object-oriented programming languages. Polymorphism allows us to perform a single action in different ways. In other words, polymorphism allows you to define one interface and have multiple implementations. The word “poly” means many and “morphs” means forms. Hence, polymorphism means “many forms”.

In Java polymorphism is mainly divided into two types -

- Compile time Polymorphism - It is also known as Static Polymorphism. This type of polymorphism is achieved by method overloading or operator overloading.
- Runtime Polymorphism - It is also known as Dynamic Method Dispatch. It is a process in which a function call to the overridden method is resolved at runtime. This type of polymorphism is achieved by method overriding.

2.5 ABSTRACTION

Data Abstraction is the property by virtue of which only the essential details are displayed to the user and the background details and information is hidden. The trivial or the non-essentials units are not displayed to the user. Data Abstraction may also be defined as the process of identifying only the required characteristics of an object ignoring the irrelevant details. The properties and behaviours of an object differentiate it from other objects of similar type and also help in classifying/grouping the objects.

Abstract classes and Abstract methods -

- An abstract class is a class that is declared with abstract keyword.
- An abstract method is a method that is declared without an implementation or body.
- An abstract class may or may not have all abstract methods. Some of them can be abstract while the rest could be concrete methods.
- There can be no object of an abstract class. That is, an abstract class cannot be directly instantiated with the new operator.

2.6 ENCAPSULATION

Encapsulation is defined as the wrapping up of data under a single unit. It is the mechanism that binds together code and the data it manipulates. Encapsulation can also be explained as a protective shield that prevents the data from being accessed by the code outside this shield. Technically in encapsulation, the variables or data of a class is hidden from any other class and can be accessed only through any member function of own class in which they are declared. Encapsulation can be achieved by: Declaring all the variables in the class as private and writing public methods in the class to set and get the values of variables.

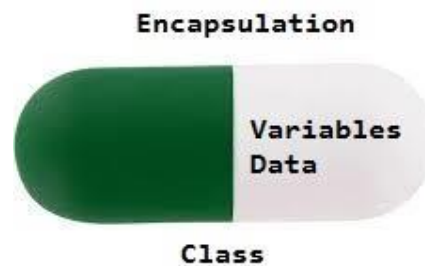


Fig 2.3: Diagrammatic representation of Encapsulation.

2.7 MULTITHREADING

Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU. Each part of such program is called a thread. So, threads are light-weight processes within a process.

Threads can be created by using two mechanisms –

- Extending the Thread class.
- Implementing the Runnable Interface.

If we extend the Thread class, our class cannot extend any other class because Java doesn't support multiple inheritance. But, if we implement the Runnable interface, our class can still extend other base classes.

2.8 I/O FUNCTIONS

Java I/O (Input and Output) is used to process the input and produce the output. Java uses the concept of a stream to make I/O operation fast. The java.io package contains all the classes required for input and output operations. We can perform file handling in Java by Java I/O API.

A stream can be defined as a sequence of data. There are two kinds of Streams –

- Input Stream – The Input Stream is used for reading data from a source.
- Output Stream – The Output Stream is used for writing data to a destination.

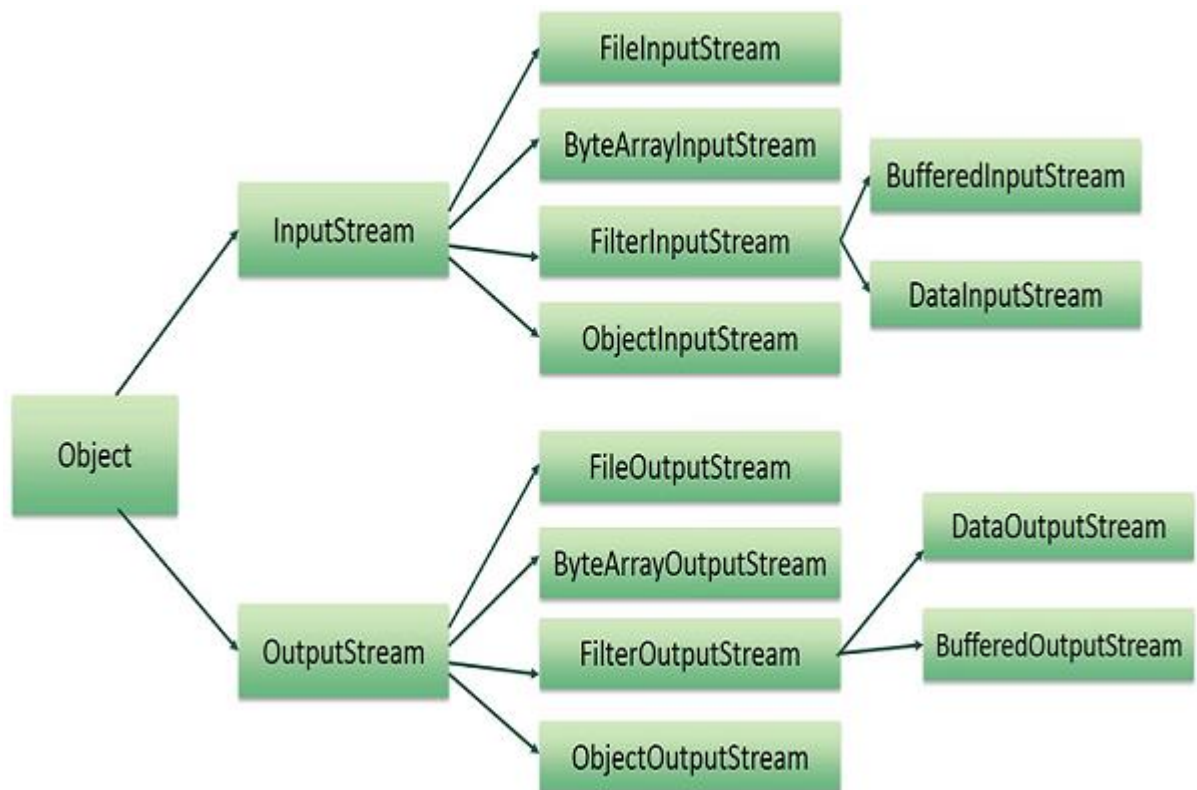


Fig 2.4: Hierarchy of classes of I/O Stream.

2.9 JAVA PACKAGES

A java package is a group of similar types of classes, interfaces and sub-packages. Packages can be considered as data encapsulation (or data-hiding). Package in java can be categorized in two form, built-in package and user-defined package. There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

Packages are used for:

- Preventing naming conflicts - For example, there can be two classes with name Employee in two packages, college.staff.cse.Employee and college.staff.ise.Employee.
- Making searching/locating and usage of classes, interfaces, enumerations and annotations easier.
- Providing controlled access - protected and default have package level access control. A protected member is accessible by classes in the same package and its subclasses. A default member (without any access specifier) is accessible by classes in the same package only.

2.10 EXCEPTION HANDLING

An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e., at run time, that disrupts the normal flow of the program's instructions.

Error vs Exception -

Error - An Error indicates serious problem that a reasonable application should not try to catch.

Exception - Exception indicates conditions that a reasonable application might try to catch.

Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, etc. The core advantage of exception handling is to maintain the normal flow of the application. An exception normally disrupts the normal flow of the application that is why we use exception handling.

CHAPTER 3

DESIGN AND ALGORITHM

An algorithm is a well-defined procedure that allows a computer to solve a problem. Another way to describe an algorithm is a sequence of unambiguous instructions. The use of the term 'unambiguous' indicates that there is no room for subjective interpretation. Every time you ask your computer to carry out the same algorithm, it will do it in exactly the same manner with the exact same result.

3.1 DESIGN

This project is designed using the concept of object-oriented programming in Java language. Some of the operations the functions in this program are designed to do are:

- Asking the user to either log in to their respective account or create a new account if they are a new user.
- Once the user logs in, they will be presented with three main categories to choose from, from where they will be directed towards the respective pages based on their choice.
- The user will be able to either post an answer to the already existing questions in the software or add a question in the existing question bank. They will also be able to browse through the already answered questions along with their responses.
- If the user wishes to answer a question, they will be presented with the question bank. They'll have to enter the number of the question they want to answer and then they will be directed towards the text area where they can type and edit their answer.
- If the user wishes to post a question, they will again be presented with the question bank. They can go through it once before posting their question to ensure that there is no similar type of questions in order to avoid confusion.
- The user can log out of their account once they have completed their task and they will be directed towards the home page after that.

3.2 ALGORITHM

STEP 1: Start the program.

STEP 2: Create a J-Frame class and design the Home Page of the software.

STEP 3: Create two options in the Home Page. One for log-in of already existing customers and another for signing up of new users by creating a new account.

STEP 4: Create objects of the respective classes to open the corresponding J-Frames.

STEP 5: If the user chooses to log-in, go to Step 8.

STEP 6: If the user chooses to sign-up, accept the user details and store them into a file.

STEP 7: Direct the user towards Home Page for them to log-in with the newly created account.

STEP 8: Accept the username and password of the user.

STEP 9: If the name and password details match and the log-in is successful, go to Step 11.

STEP 10: If the details do not match, display an appropriate error message and tell the user to try again. Go to Step 8.

STEP 11: Open the J-Frame class for the user to choose from the option of posting a question, posting an answer and browsing through the feed.

STEP 12: Accept the user input and open the corresponding J-Frame class by creating an object of the class.

STEP 13: If the user chooses to post a question, the object of that class is created and the J-Frame class is opened. Else, go to Step 16.

STEP 14: Display all the existing questions in the question bank in a text area. In a separate text area provide an option to enter the question the user wants to post.

STEP 15: As soon as the user clicks on the submit button, the question is added to the question bank list of the software. Go to Step 23.

STEP 16: If the user chooses to post an answer, the object of that class is created and the J-Frame class is opened. Else, go to Step 21.

STEP 17: Display all the questions in the question bank in a text area. Provide a text field for the user to enter the question number he/she wants to answer.

STEP 18: Accept the user input. If the input is a valid question number, go to Step 17. Else, go to Step 19.

STEP 19: The question number entered by the user is displayed in a text area along with the question they want to answer. A separate text area is also provided to enter the user perspective on the question.

STEP 20: As soon as the user clicks on the submit button, the question along with the answer will be added to the feed of the software. Go to Step 23.

STEP 21: If the user chooses to browse through the feed, the object of that class is created and the J-Frame class is opened.

STEP 22: All the questions that have been answered by the users are displayed in a text area along with their answers.

STEP 23: After the user has finished their purpose, they can choose to either go through the options of using the application again or to log out of their respective accounts.

STEP 24: If the user wishes to use the options of the application again, Go to Step 11.

STEP 25: If the user wishes to log out of their account, Go to Step 3.

STEP 26: Exit from the program.

CHAPTER 4

IMPLEMENTATION

Each aspect of this application has been divided into modules which are then implemented by using classes. There is a total of seven classes in this application, hence the seven module functionalities and their implementation have been described in detail.

4.1 MODULE 1 IMPLEMENTATION

Each aspect of this application has been divided into modules which are then implemented by using classes. The first class that needs to be created is the Home Page. It is J-Frame class that implements graphic user interface based on our needs. An object of the Home Page class is created in the main function of a class. This class has a main function which has the object to start the running of this application. The object of the Home Page class is used to set the visibility of the J-Frame class as “true”, so that it is visible to the user. Every time someone wants to access this application, they will have to run this class.

The Home Page serves to distinguish new users from already existing customers. It provides any user who runs the application with two options – Existing user Log-In, Sign Up option for creating a new account for new users. If the if the user chooses to Log-In, they are directed towards the Log-In page class for them to enter their details. If the user chooses the Sign-Up option, they are directed towards the Sign-Up page class where they are provided with the option to create a new account.

When a user after completing their work on the application, chooses to Log-out or wishes to exit their account, they are provided with a Log-Out option. Once the user clicks on this button, they are directed towards the Home Page preparing the application for a new user.

4.2 MODULE 2 IMPLEMENTATION

The user is directed towards the Log-In page if they choose that option in the Home Page class. This is done by creating an object of the Log-In page class. The object is then used to set the visibility of the Log-In page as “true”. Once the Log-In page is visible to the user, the Home Page needs to be set visible as “false”. The Log-In page is also a J-Frame class that implements graphic user interface.

The Log-In page consists of two text fields. One of them is to enter the username of the user and the another is a password text field to enter the password. The username of the person is unique and no two users can have the same username. The password field is encrypted and the user will not be able to see the text entered in the field as it will be masked by “*” for security reasons. Both the username and password are case sensitive, so the user will have to enter the exact details he gave while creating his/her account.

Once the user is done tying his/her details and click on the submit button, they entered details are compared with the user database of the application. The user database is a file that contains all the usernames along with their passwords. If the user enters a valid username that already exists in the file with the correct corresponding password, the Log-In is considered to be successful and the user is directed towards the Query class for them to perform their task. If the user enters an invalid username that does not exist in the user database, an appropriate error message is displayed stating that the account is not found. If the user enters a valid username but the password entered is incorrect, an error message is displayed stating the same.

The user is also provided with an option to go back to the Home Page from the Log-In page in case the user wishes to do so. There is also an “Clear” option that erases the already typed data in username and password text fields in case the user has typed in incorrect data by mistake.

4.3 MODULE 3 IMPLEMENTATION

The user is directed towards the Sign-Up page if they choose that option in the Home Page class. This is done by creating an object of the Sign-Up page class. The object is then used to set the visibility of the Sign-Up page as “true”. Once the Sign-Up page is visible to the user, the Home Page needs to be set visible as “false”. The Sign-Up page is also a J-Frame class that implements graphic user interface.

The Sign-Up page consists of three text fields. One of them is to enter a username. The other two are password text fields. One is for the password and the other one is to confirm the entered password so that there is no discrepancy in the users mind. The password fields are encrypted and the user will not be able to see the text entered in the fields as they will be masked by “*” for security reasons. Username and passwords are case sensitive and the user must remember their details for future Log-In purposes.

Once the user types his/her details and clicks on the Submit button, the details are first matched with the already existing details in the user database. If the username given by the user already exists in the database, an error message is displayed requesting the user to select a new username. Username of all the users need to be unique in order of them to access their accounts. After making sure that the username selected is valid, the password entered is matched with the option where the user is asked to re-enter the password. If both the passwords match, the user details (username and password) are added to the user database. The user then has to go back to the Home Page in order to Log-In to their newly created account and use the application according to their needs. The option to go back to Home Page is provided in the form of a button. There is also an “Clear” option that erases the already typed data in username and password text fields in case the user has typed in incorrect data by mistake.

4.4 MODULE 4 IMPLEMENTATION

After the user has logged in to their account successfully, a J-Frame class is opened that presents the user with the three main aspects of this application. The J-Frame class that is opened is set visible as “true” when the log-in is done successfully. The three options that are presented to the user are –

- Post a question.
- Post an answer.
- Browse through the feed.

The three options are accompanied with a radio button each. The radio buttons are grouped into a single Button Group. This is done to ensure the user cannot select more than one option at a time. If the user tries to select an option after he/she has already chosen one, the previous option is automatically deselected and the latest option will be selected. The corresponding class that has been selected is opened by creating its object.

4.5 MODULE 5 IMPLEMENTATION

If the user chooses the “post a question” option, they are directed towards a J-Frame class that will display all the existing question in a text area. There will be an addition text area where the user can type the question, they want to add to the question bank of the application. The existing questions are shown for the reference of the user so that the questions of similar type are not repeated. Once they user has typed and submitted their query, the question will be added to the question list of the application. It is also indicated by the display of appropriate message on the screen. The user can submit multiple questions. After each submission, the user will have to clear he existing question from the text field with the help of a button provided. After clearing the text area, the user can type in the next question. The user is also provided with the option to go back to the Query page if they want to perform any other task. If they are done with their tasks, they can choose to log-out from their respective accounts.

4.6 MODULE 6 IMPLEMENTATION

If the user chooses “post an answer” option from the Query class, they are first directed towards a J-Frame class where all the questions in question bank of the application are displayed. The user will be asked to enter the corresponding question number they want to answer. Each question is accompanied with a question number in the text area where the questions are displayed. The user will have to type the question number of the question they want to post their facts or opinions about in the text field provided. If the question number entered is invalid or the question number does not exist in the list displayed, an error message is displayed stating the same and the text field becomes empty. This process repeats until the user enters a valid question number. The user should click on the submit button after typing in the question number so that the option to post their answer is made available. The user is also provided with the option to go back to the Query class J-Frame if they want to perform another task. They can also choose to log—out from their accounts if they wish to do so.

After the user enters a valid question number and submits, the users are directed towards another J-Frame class where they are provided with the option to type in their answers. A text area displaying the question that they selected is shown. Below that is the text area to type and edit their answer. After writing their answers, they can click on the submit button provided. Once they click on the submit button, the question along with their answer is added to the feed of the application. The user can choose to go back to the question page and choose to answer another question with the help of the “Go Back” button provided. This way a single user can post multiple answers to questions. There is also a clear option provided to the user that erases all the text typed for the answer and the user can start afresh. There is also a log-out option provided, which can be used by the user after they are done with all their tasks. Clicking on this option leads the user to the Home Page and a new user can use this application.

4.7 MODULE 7 IMPLEMENTATION

If the user chooses the “Browse through the Feed” option, they are directed towards a J-Frame class which consists of all the questions that have been answered along with their questions. A text area is used to display the details. The text area consists of a scroll bar that should be scrolled up to view the older answers. The answers are displayed according to the date and time they were submitted. The user can read through the answers and use the scroll option to view more answers. After the user is done reading, he/she can either choose to go back and perform another task or log-out from their respective account. This can be done with the help of buttons provided below the text area.

CHAPTER 5

RESULTS

The below screenshots are the working results of the program.

1. Home Page of the program.

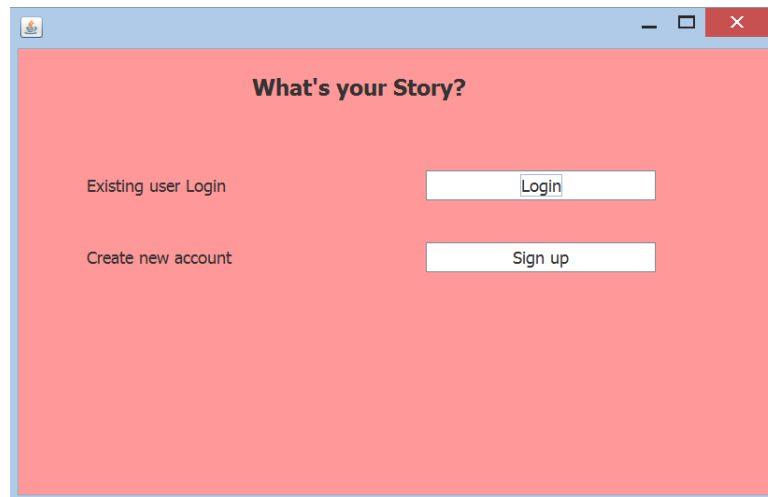


Fig 5.1: Result Screenshot – 1.

2. Log-in Page.



Fig 5.2: Result Screenshot – 2.

3. Error in the Log-in page.

Invalid username error –



Fig 5.3: Result Screenshot – 3.

Valid username but wrong password error-

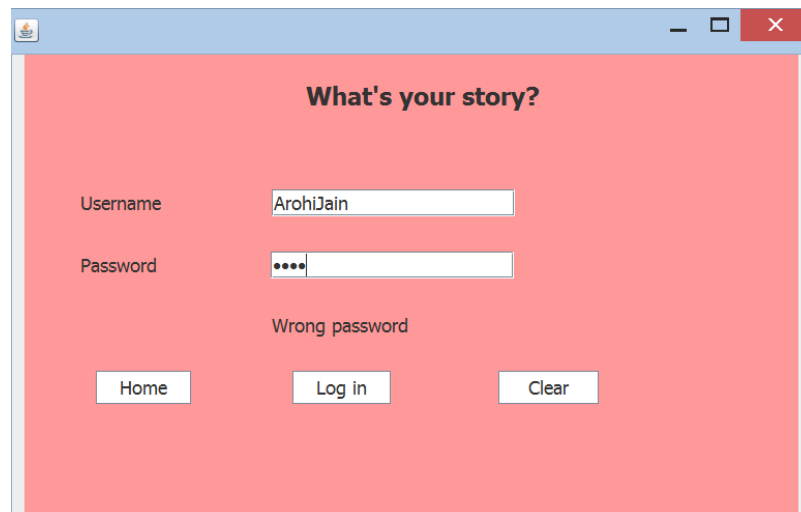
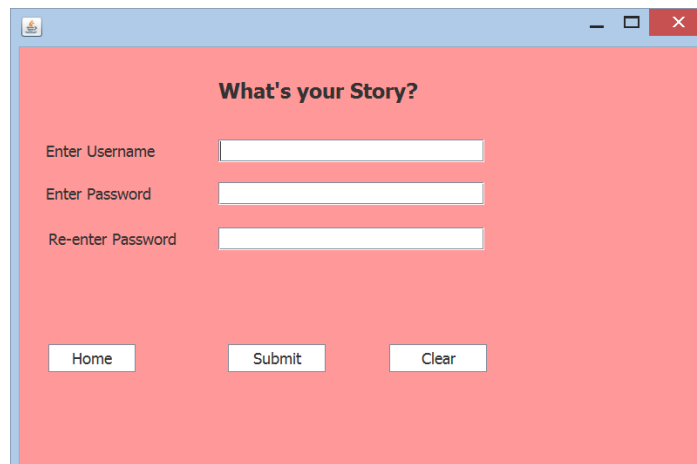


Fig 5.4: Result Screenshot – 4.

4. Sign-up Page.



What's your Story?

Enter Username

Enter Password

Re-enter Password

Fig 5.5: Result Screenshot – 5.

5. Error in the Sign-up page.



What's your Story?

Enter Username

Enter Password

Re-enter Password

Username already exists. Choose a new username.

Fig 5.6: Result Screenshot – 6.



What's your Story?

Enter Username

Enter Password

Re-enter Password

Passwords do not match

Fig 5.7: Result Screenshot – 7.

6. Options of the program after successful Log-in.

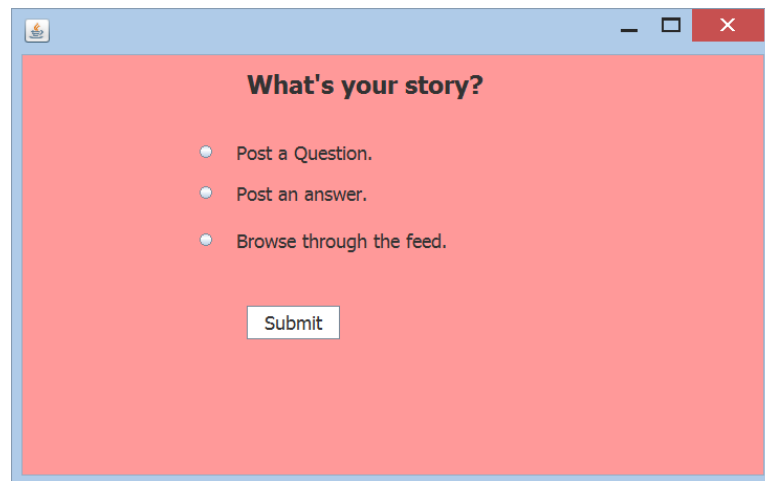


Fig 5.8: Result Screenshot – 8.

7. Post a question Page.



Fig 5.9: Result Screenshot – 9.

8. Write an answer page.



Fig 5.10: Result Screenshot – 10.

The next page after the user chooses the question to be answered.

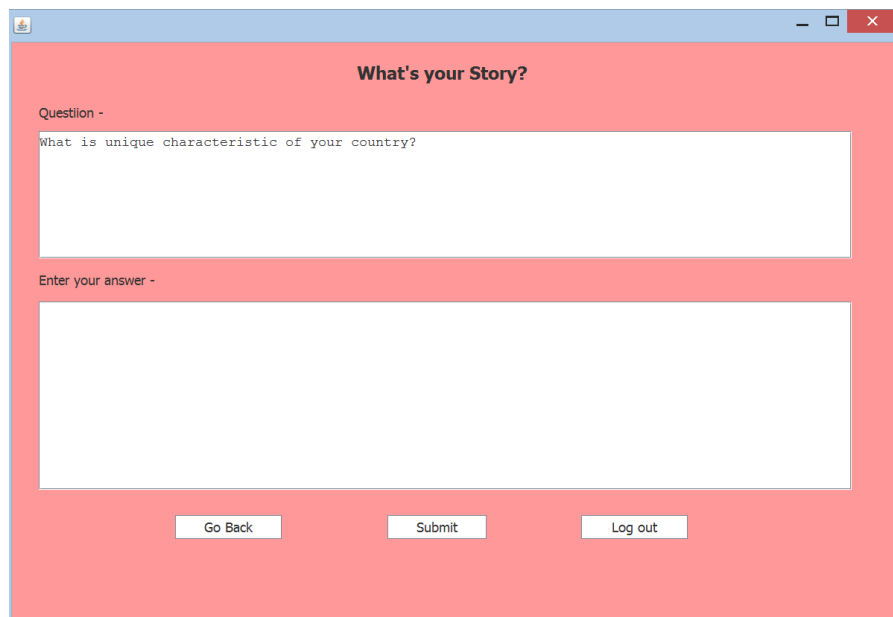


Fig 5.11: Result Screenshot – 11.

9. Browse the Feed Page.

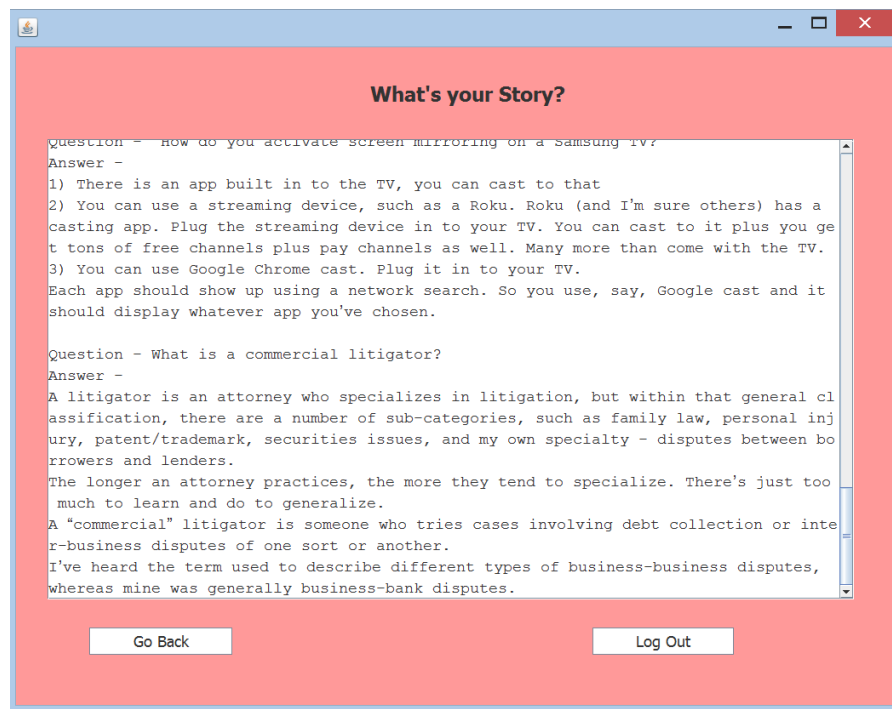


Fig 5.12: Result Screenshot – 12.

CHAPTER 6

CONCLUSION

A platform where people can have access to different opinions on varied topics and where users can resolve their queries is a new way to bridge minds together. What's your Story, is a platform where people can get their facts and also express any opinions they have on any topic. This application gives an individual a chance to express their opinions to the world without the fear of being judged. This project is developed using the concept of object-oriented programming in Java language.

Developing this project has also led me to have a better understanding of the object-oriented programming related topics. Implementing the concepts in real life has made me more confident in the topic and can henceforth be able to develop real time applications with much more clarity in mind. This has also led us to have a better grasp on the Java language. The main objective of understanding and analysing the technological needs of the society has been achieved.

REFERENCES

- [1] Herbert Schildt, Java: The Complete Reference, McGraw-Hill, Tenth Edition, 2018.
- [2] Cay S. Horstmann, Core Java SE 9 for the Impatient, Addison Wesley, Second Edition, 2018.
- [3] SAMS teach yourself Java – 2: 3rd Edition by Rogers Cedenhead and Leura Lemay Pub. Pearson Education.
- [4] Ken Kousen, Modern Java Recipes, O'Reilly Media, Inc., 2017.
- [5] Cay S. Horstmann, Core Java™ Volume I—Fundamentals, Prentice Hall, Tenth Edition, 2015.
- [6] <https://www.javatpoint.com/java-fileinputstream-class>
- [7] <https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/>
- [8] <https://www.guru99.com/java-swing-gui.html>
- [9] https://www.w3schools.com/java/java_oop.asp
- [10] <https://www.javatpoint.com/java-oops-concepts>