

**QUIZ GAME**

**OBJECT ORIENTED PROGRAMMING**

**PROJECT**

**Submission to:**

**Miss Tooba Rashid**

**Miss Rukhsana Majeed**

**Submitted By**:

**Haris Bin Irfan (37888) – MAIN CAMPUS**

**TABLE OF CONTENTS**

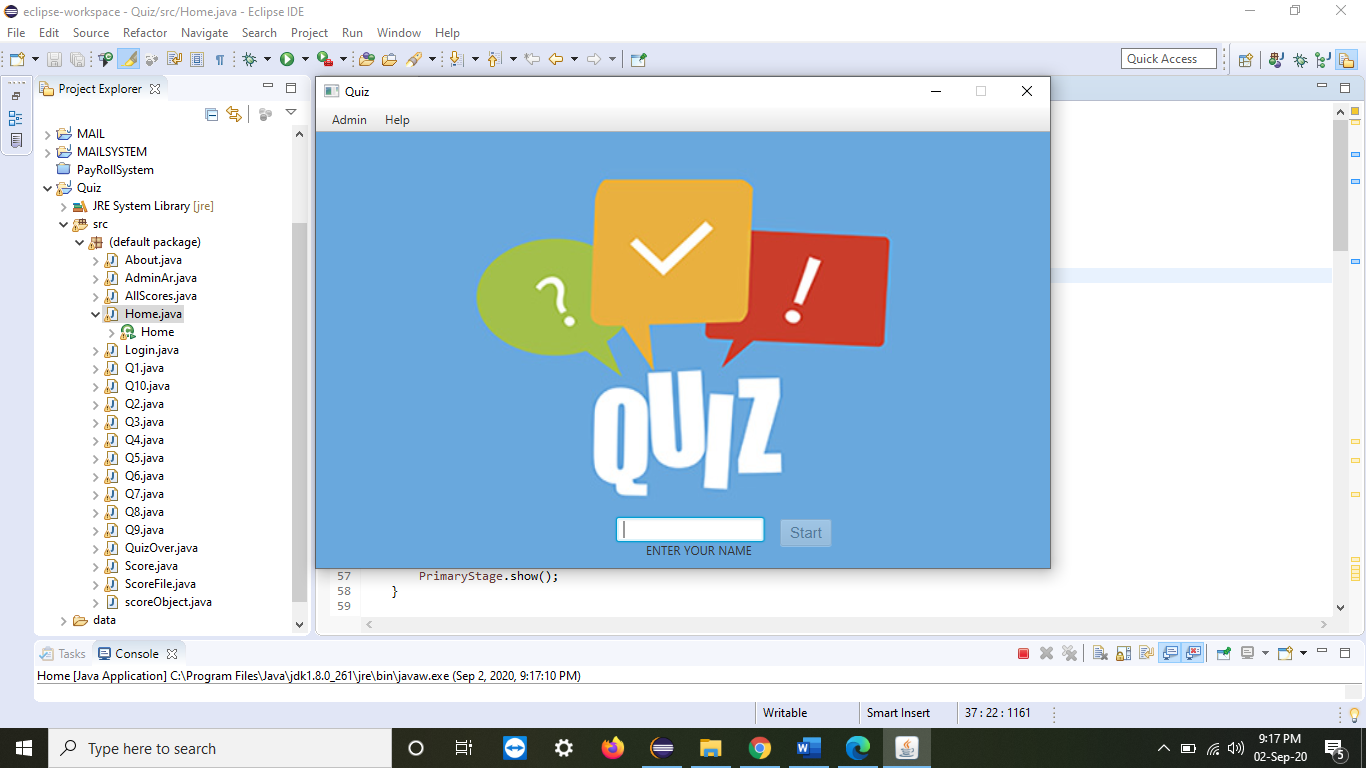
1. INTRODUCTION -------------------------------------- PAGE 2
2. PROJECT DESCRIPTION ---------------------------- PAGE 3
3. RUNNING SNAPS --------------------------------------PAGE 3-6
4. SYSTEM OVERVEW ---------------------------------- PAGE 7
5. FUNCTIONAL REQUIREMENTS ------------------- PAGE 8
6. NON-FUNCTIONAL REQUIREMENTS ----------- PAGE 8
7. CLASS DIAGRAM -------------------------------------- PAGE 9
8. CODE SNIPPETS ---------------------------------------- PAGE 10
9. CONCLUSION ------------------------------------------- PAGE 15

# PROJECT DESCRIPTION

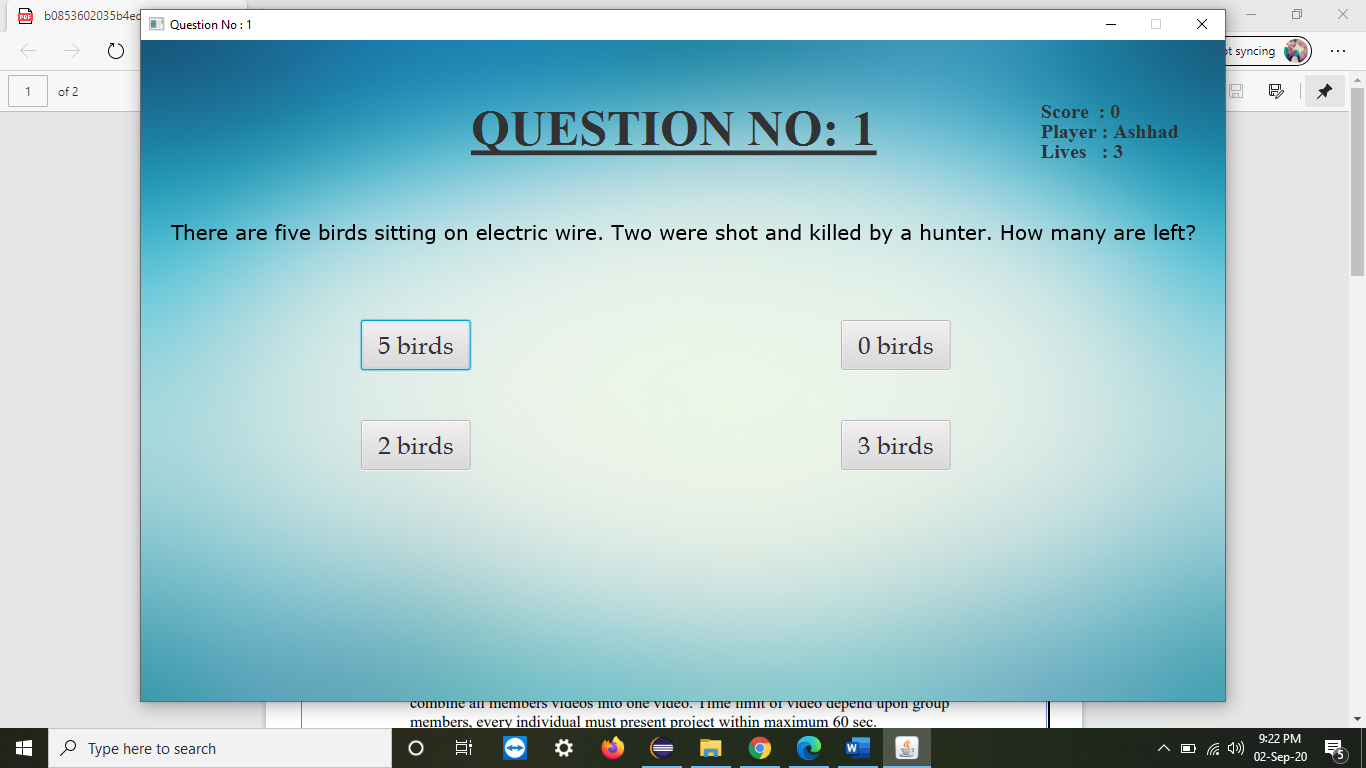
Our Quiz Management System is based on some general knowledge and IQ questions, that any user can get entertained with. It is developed on Java and JavaFX has been used to carry out the project. The Quiz Management System has been developed on the concept of CMS and wide graphical tools have been integrated. This project covers all pillars of Object-Oriented Programming. Any version of eclipse can be used to run this Quiz Management System.

## SYSTEM OVERVIEW

Our system consists of some fun informational questions for user to get entertained with. The user will be asked to enter the name first:

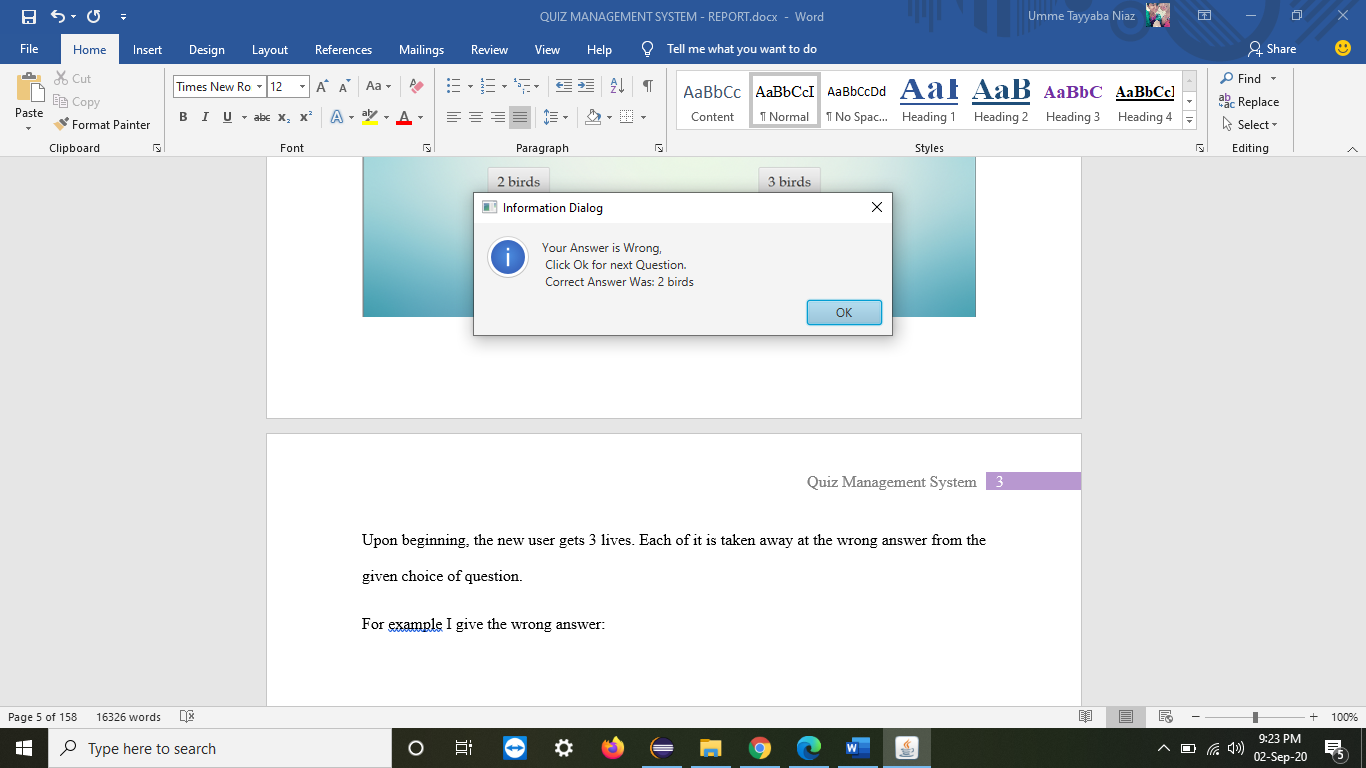


I have entered my name and pressed START button:

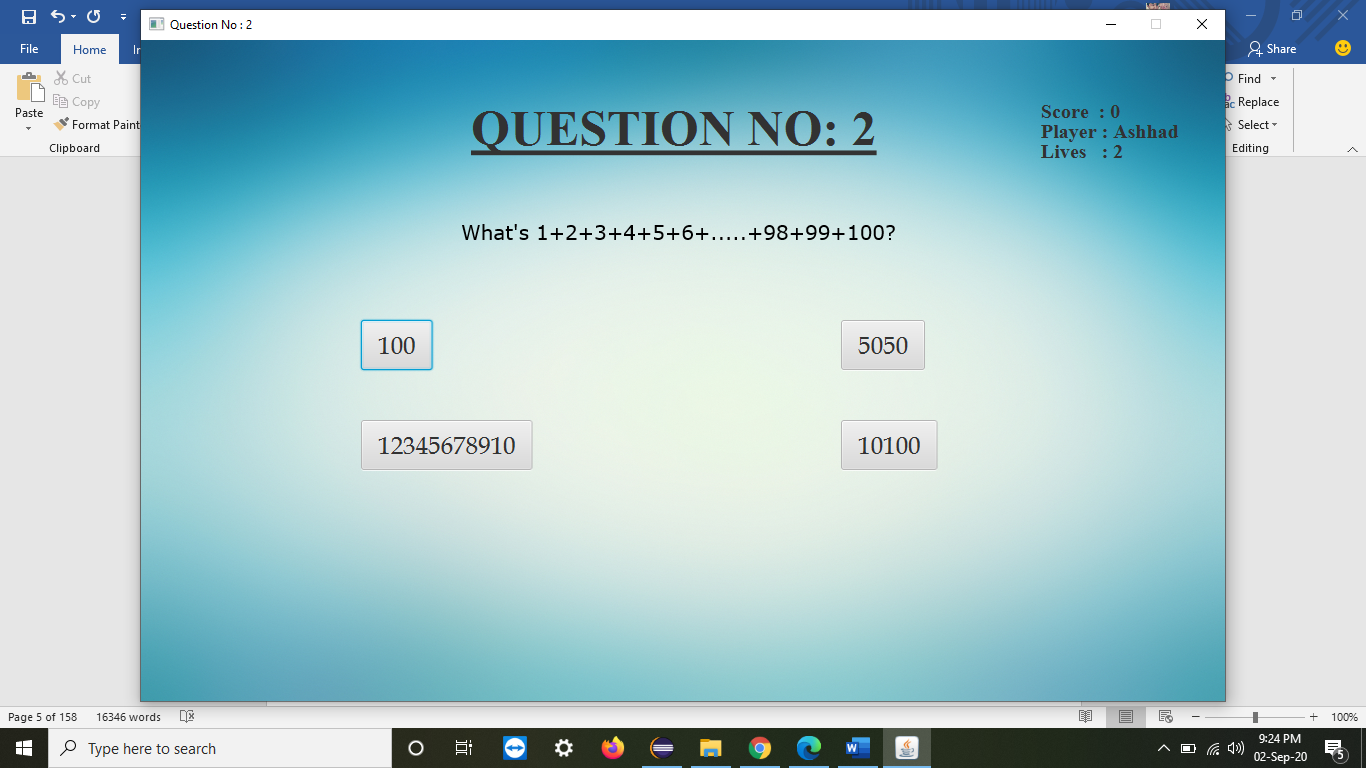


Upon beginning, the new user gets 3 lives. Each of it is taken away at the wrong answer from the given choice of question.

For example, I give the wrong answer:



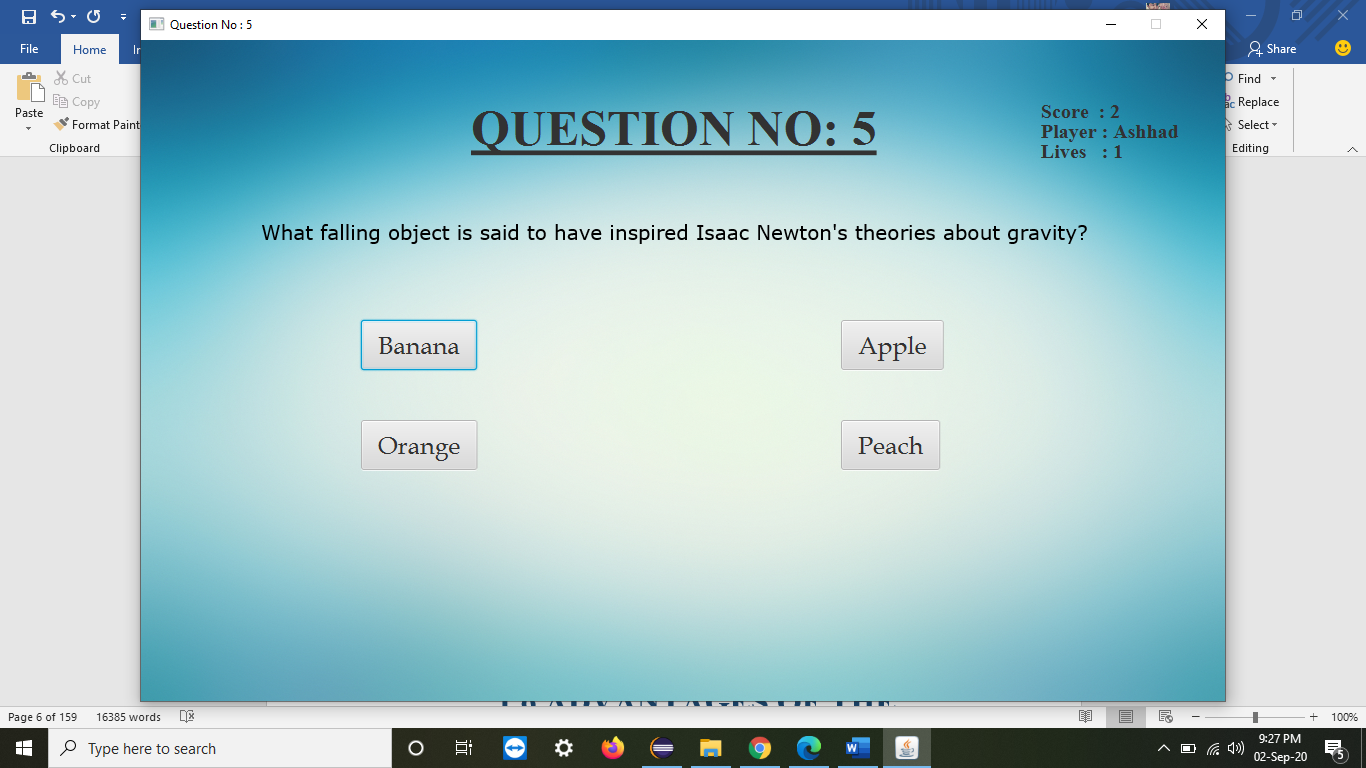
Now, on pressing okay I am directed back to the dashboard when I can see a deduction in my lives:

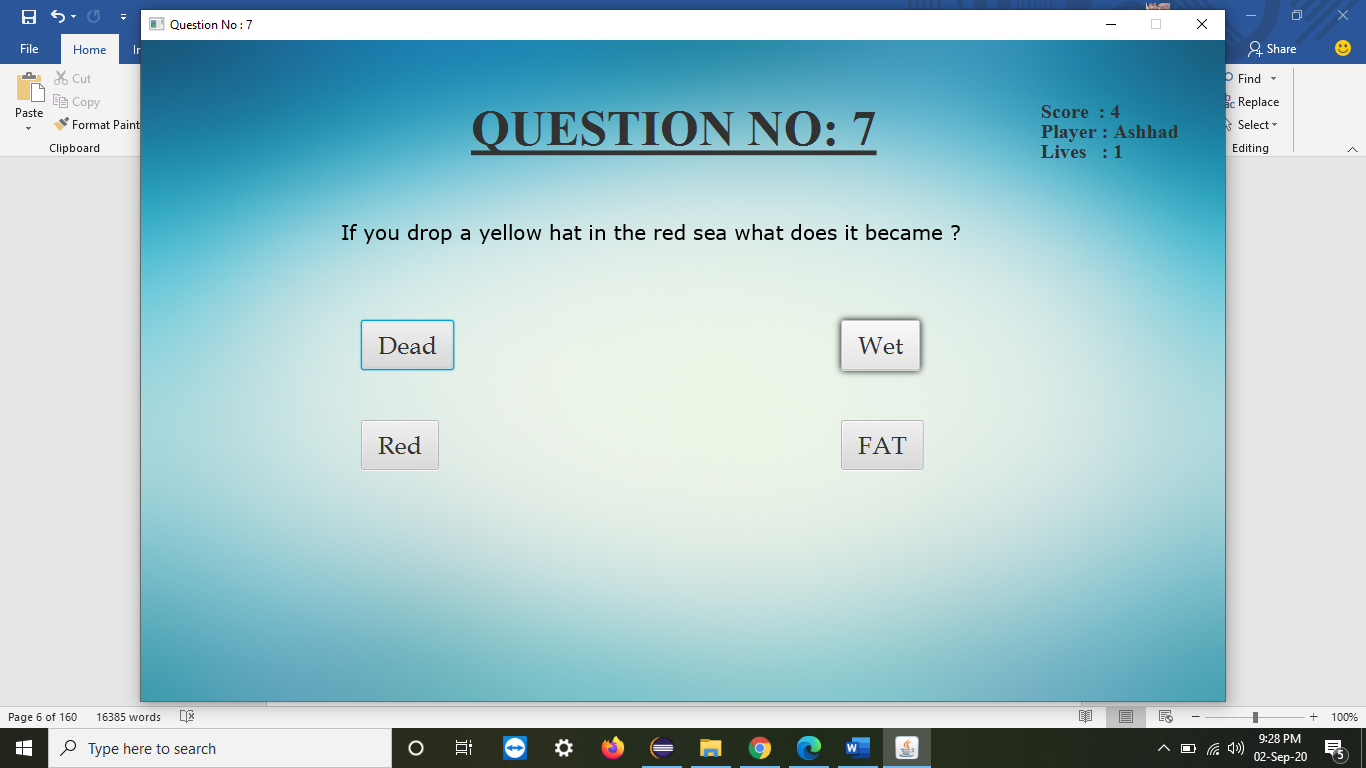


Now I will give the correct answer:

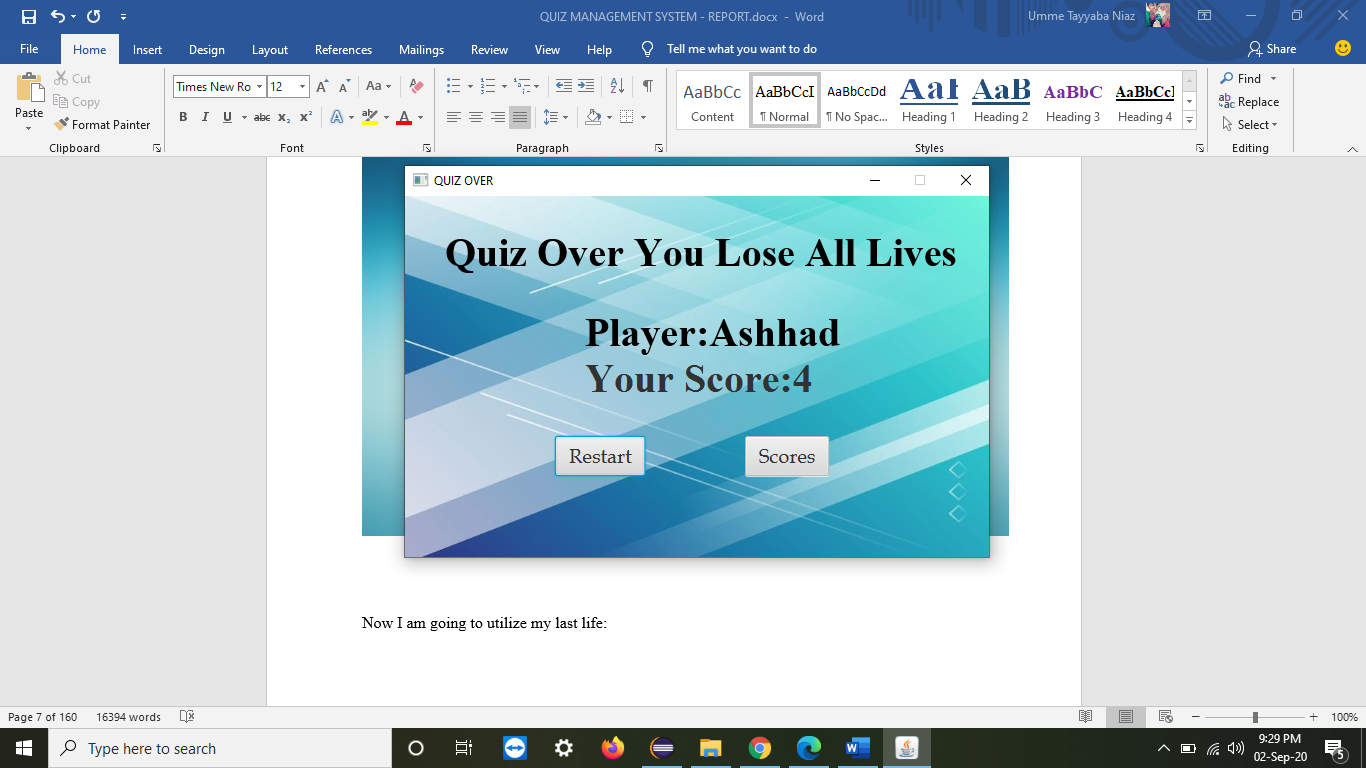


Now coming back to the board for the next question. Similarly, a list of limited question goes on unless the questions come to an end or all three lives are being used.



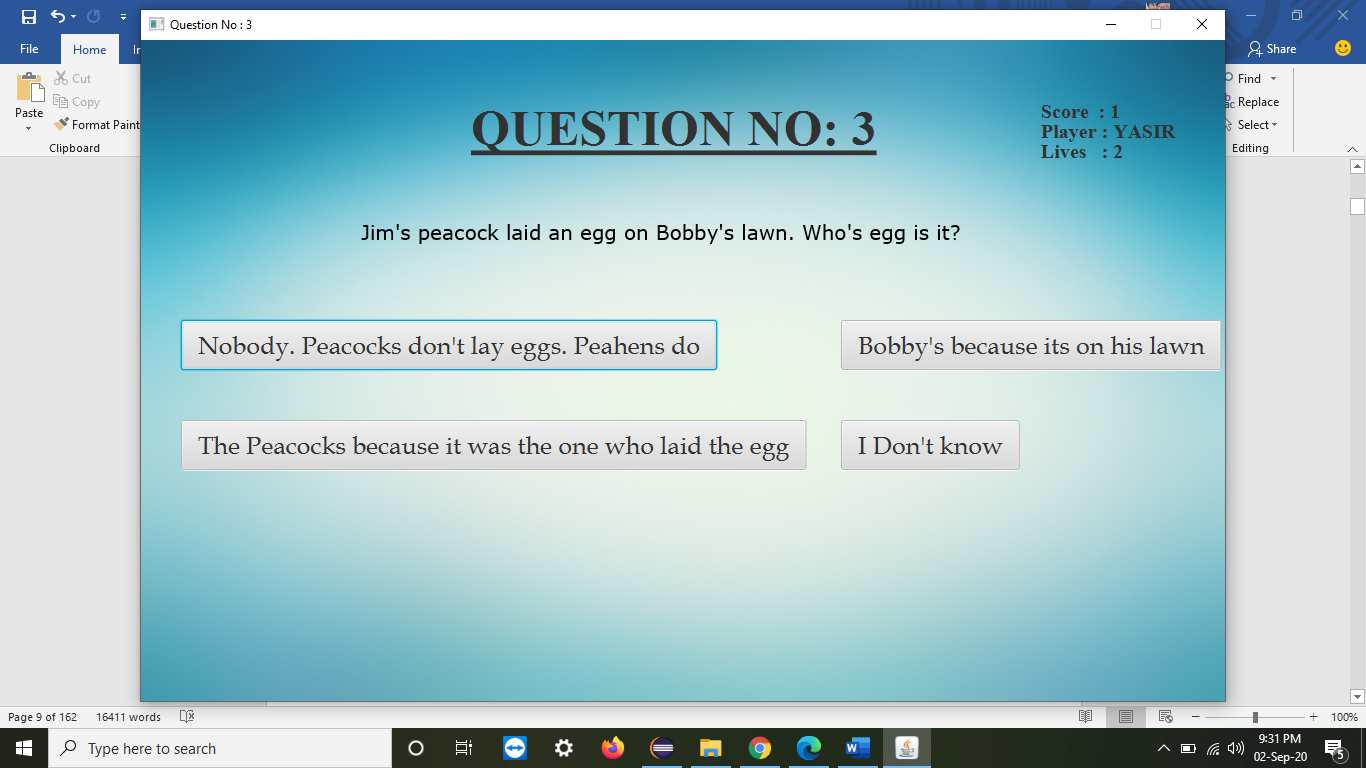


Now I am going to utilize my last life:



Upon restarting, you can again start the quiz:

Now I am doing the quiz with another user:



## FUNCTIONAL REQUIREMENTS

* The same patterns of question, choices and answers can be applied and implemented to other systems like Survey as well.
* Click on the id of quiz and ready to start it just clicking on a button.
* After completing all questions, result will be displayed automatically.
* Maintain the scores
* Front end: Eclipse with GUI, JDK version 1.8.0
* Backend: Classes, constructors, parameterized constructors, JavaFX, etc.
* LIBRARIES from JavaFX used:
* **import** javafx.application.Application;
* **import** javafx.beans.binding.Bindings;
* **import** javafx.beans.binding.BooleanBinding;
* **import** javafx.collections.ObservableList;
* **import** javafx.event.ActionEvent;
* **import** javafx.event.EventHandler;
* **import** javafx.geometry.Insets;
* **import** javafx.scene.Group;
* **import** javafx.scene.Scene;
* **import** javafx.scene.control.Button;
* **import** javafx.scene.control.Label;
* **import** javafx.scene.control.Menu;
* **import** javafx.scene.control.MenuBar;
* **import** javafx.scene.control.MenuItem;
* **import** javafx.scene.control.TextField;
* **import** javafx.scene.effect.DropShadow;
* **import** javafx.scene.image.Image;
* **import** javafx.scene.image.ImageView;
* **import** javafx.scene.input.MouseEvent;
* **import** javafx.scene.layout.GridPane;
* **import** javafx.scene.paint.Color;
* **import** javafx.scene.paint.ImagePattern;
* **import** javafx.scene.text.Font;
* **import** javafx.scene.text.FontWeight;
* **import** javafx.scene.text.Text;
* **import** javafx.stage.Stage;

## NON- FUNCTIONAL REQUIREMENTS

1. Easy to use, once started, user can get a pile of different questions covering all aspects of nature, no matter whatever the interest user has.
2. Enhances Knowledge.
3. Security and usability, which are observable during operation (at run time).
4. Evolution qualities, such as testability, maintainability, extensibility and scalability,  
   which are embodied in the static structure of the system.

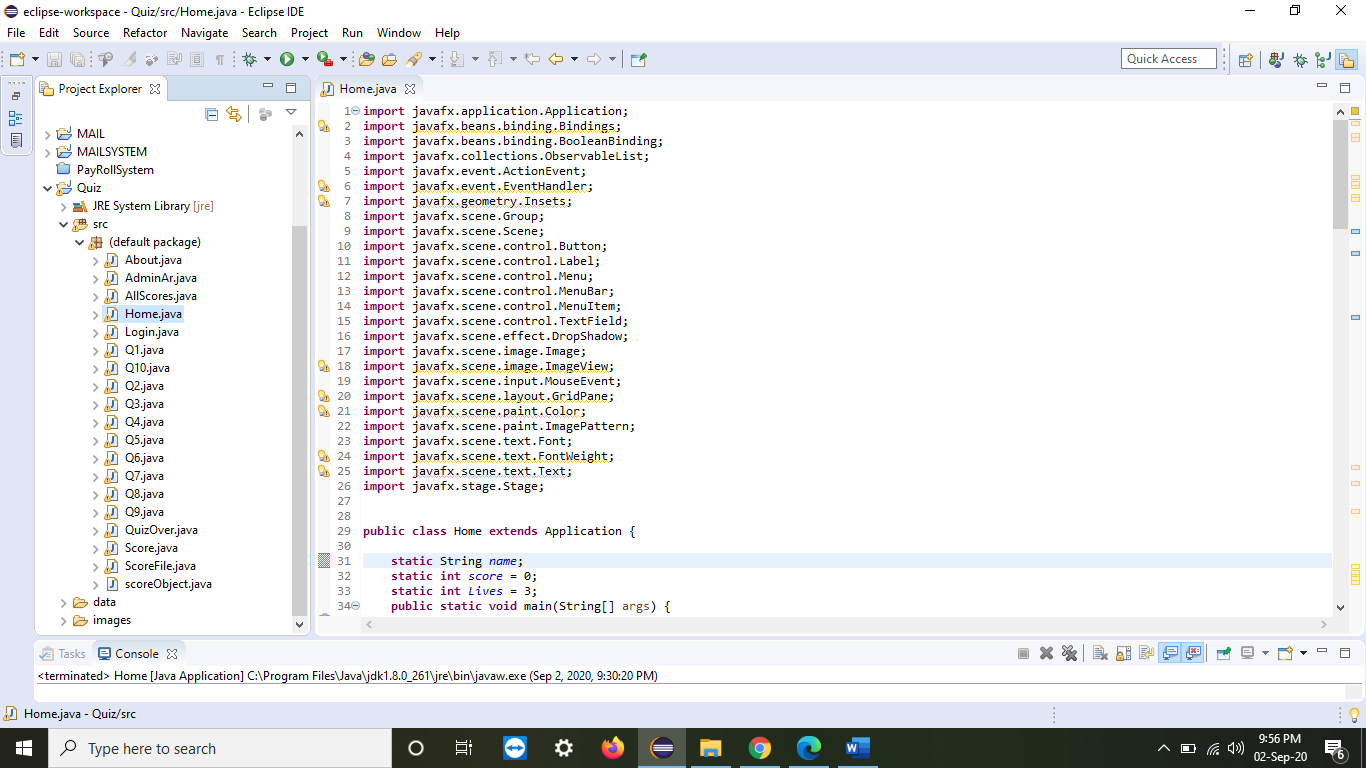
## CLASS DIAGRAM



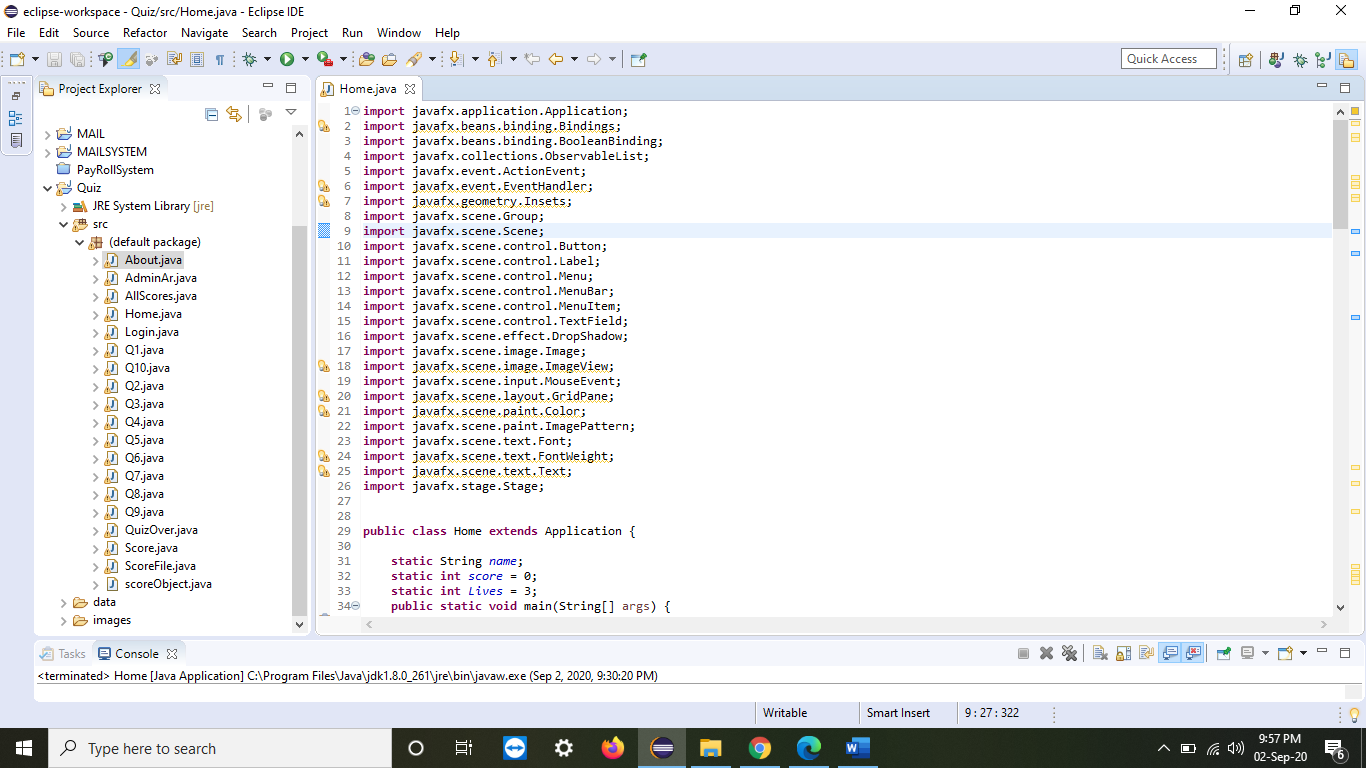
## IMPLEMENTATION

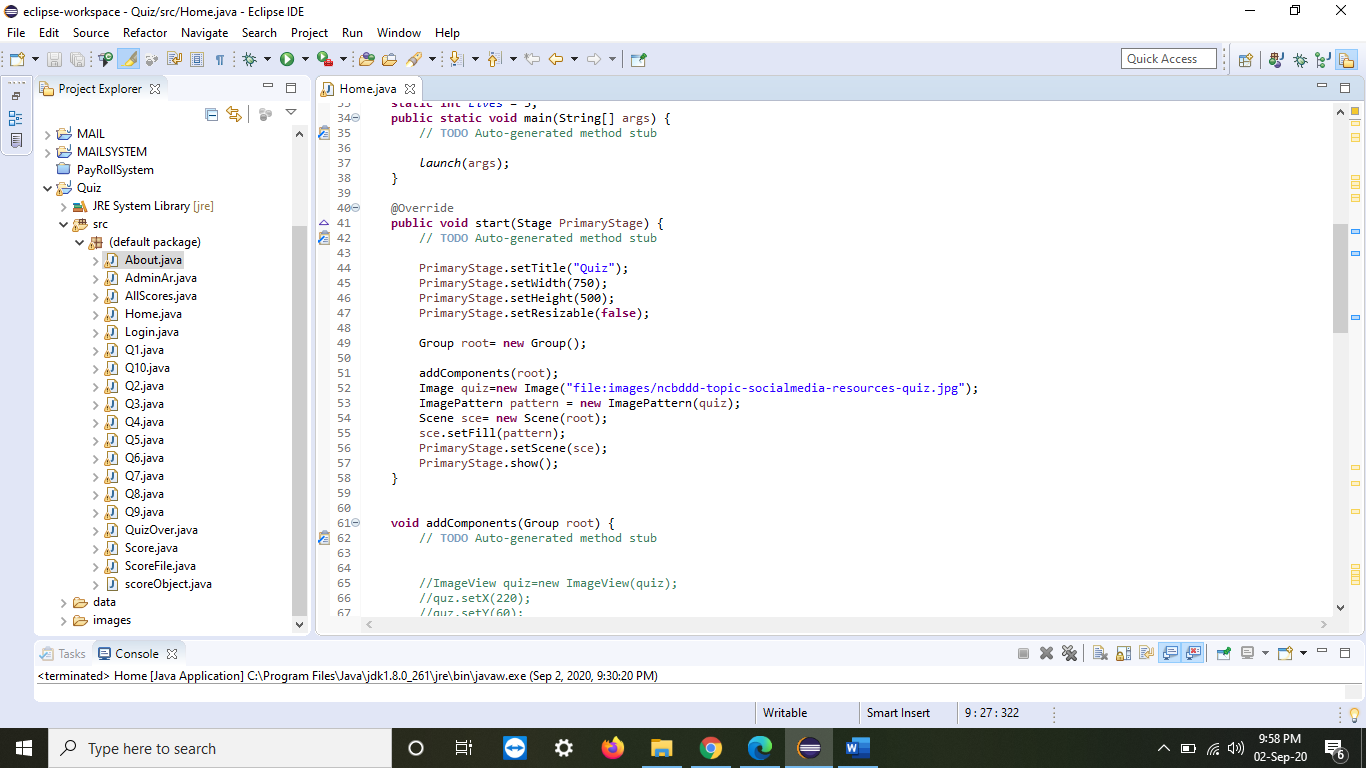
CODE SNIPPETS:

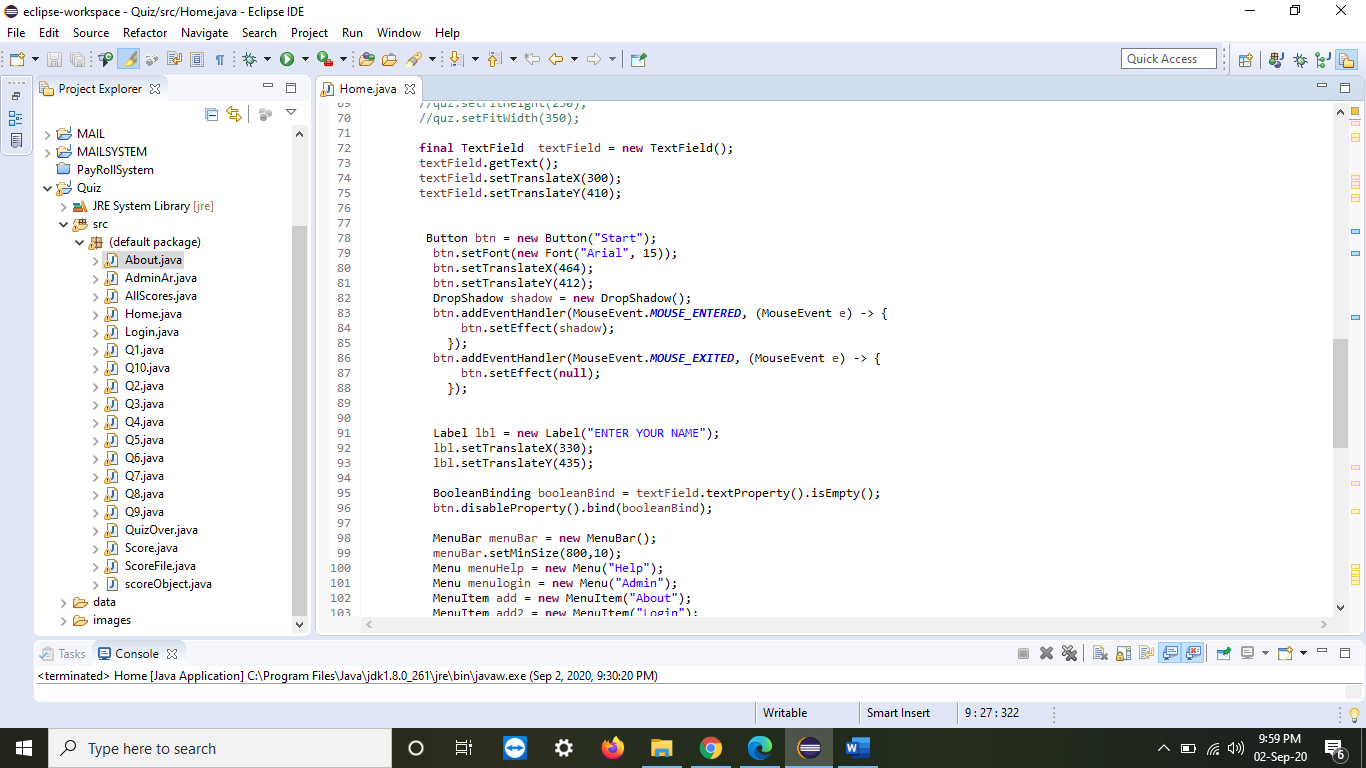
The snapshots of Code from Main class, is given. Rest of the code is provided with the project.

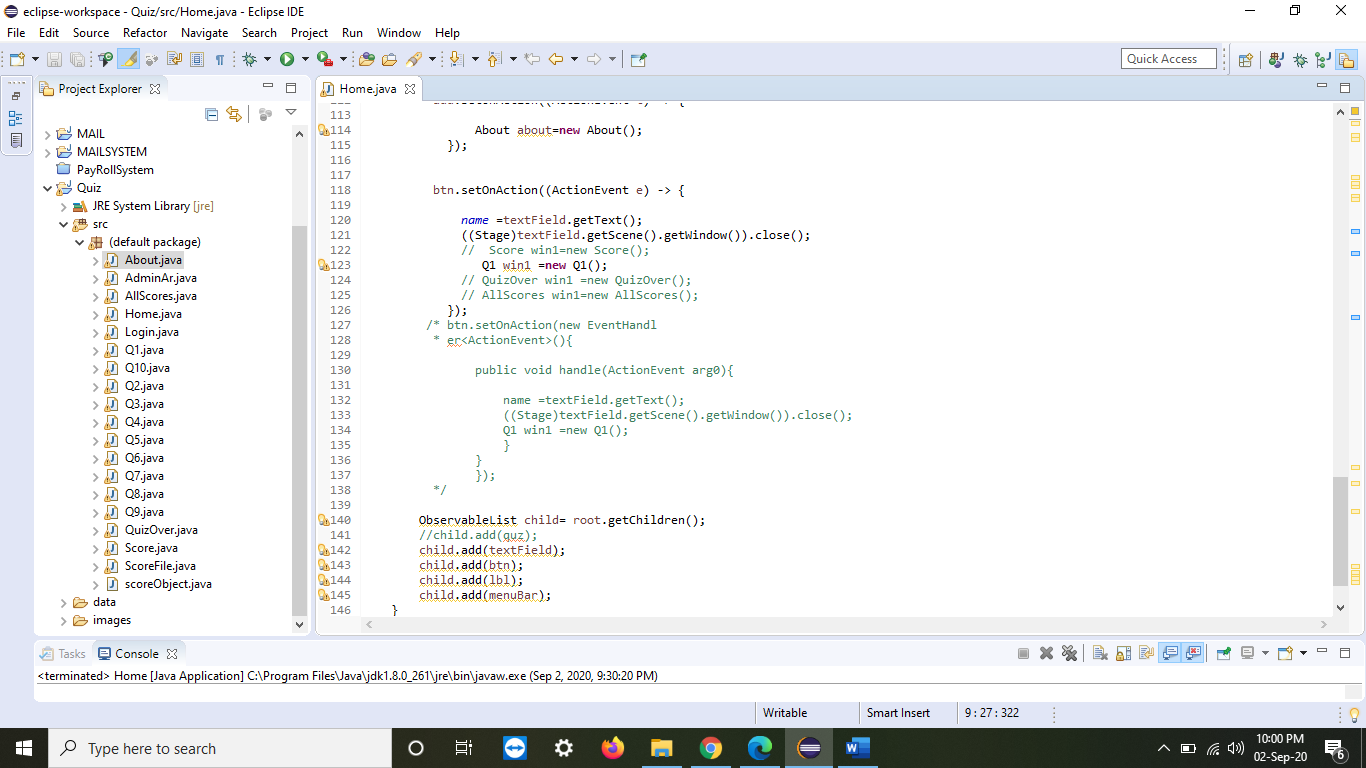


HOME.JAVA:









## CONCLUSION

## This quiz system can easily be implemented in any form such as it can provide facility to conduct examination. It saves time as it allows number of students to give the exam at a time and displays the results as the test gets over, so no need to wait for the result. User can register, login and give the test with his specific id, and can see the results as well