**Name: Hassan Rasheed**

**Registration No: 00000126900**

**Class: BSCS5**

**Section: B**

**Advanced Programming Assignment 1**

**Date: 20th September 2017**

Table of Contents

Introduction………………………………………………………………………………………………… 2

Learning………………………………………………………………………………………………………. 2

Sample Runs………………………………………………………………………………………………… 3

Maven…………………………………………………………………………………………………………. 9

Profiling………………………………………………………………………………………………………. 11

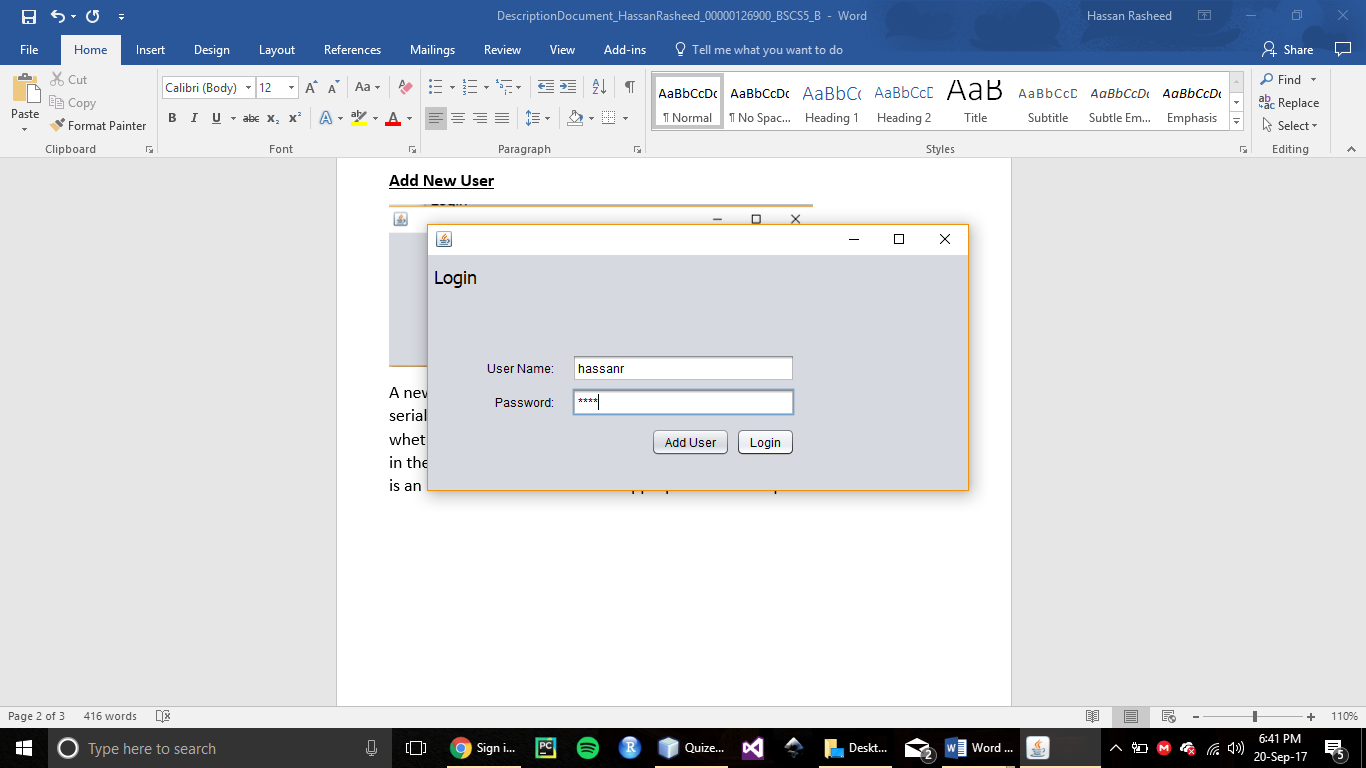
GitHub………………………………………………………………………………………………………… 13

**Introduction:**This assignment required us to create a quiz application which allows a user based on their role (if they are an instructor) create a quiz and (if they are a student) take the created quizzes. For this purpose first of all a user class was created in which required variables were stored such as their username, password, role. Next a structure was needed to store the data of the quizzes that were being created so a quiz class was created that stored the required components of a quiz i.e. the quiz title, a short description of the quiz, list of questions (objects of question type) and the total marks that the quiz carries. The questions further are of 3 types, MCQs, True/False and Numeric so they also required a structure to be stored so a question class was also created. Question class contained question type, the question statement, expected answer of the question, and 4 options for the MCQs in a string array. Then different views were created using JAVA GUI to make an interactive graphical user interface for the users to easily use the application which will be shown in this document later.

**Learning:**The main concepts that I learned during the phase of doing this assignment was the concept of serialization, use of maven, and to use the graphical user interface of JAVA to work with the classes and other OOP concepts that were learned earlier. The MVC (Model View Controller) model which has been implemented in the code by keeping the data storage, the classes and other functionality separate from the view or the graphical user interface.

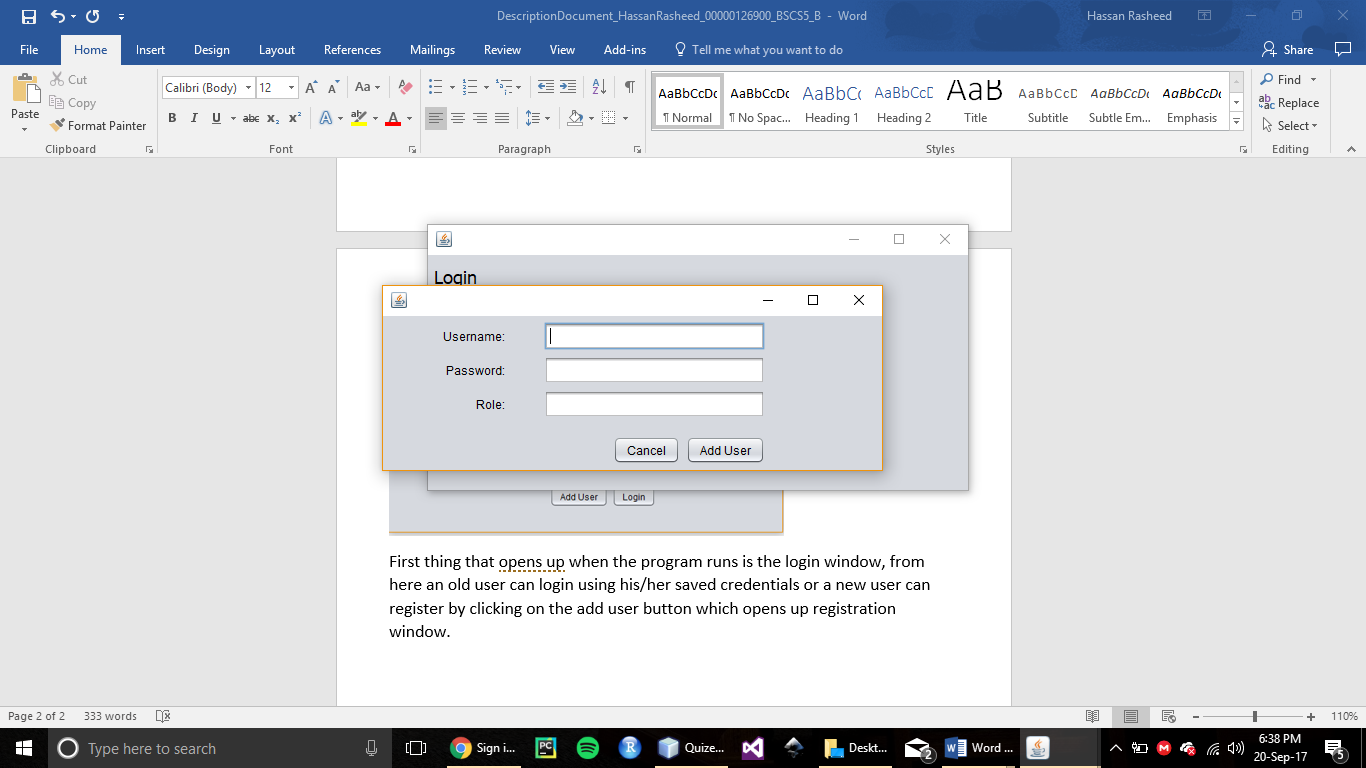
**Sample Runs:**

**Login**



First thing that opens up when the program runs is the login window, from here an old user can login using his/her saved credentials or a new user can register by clicking on the add user button which opens up registration window.

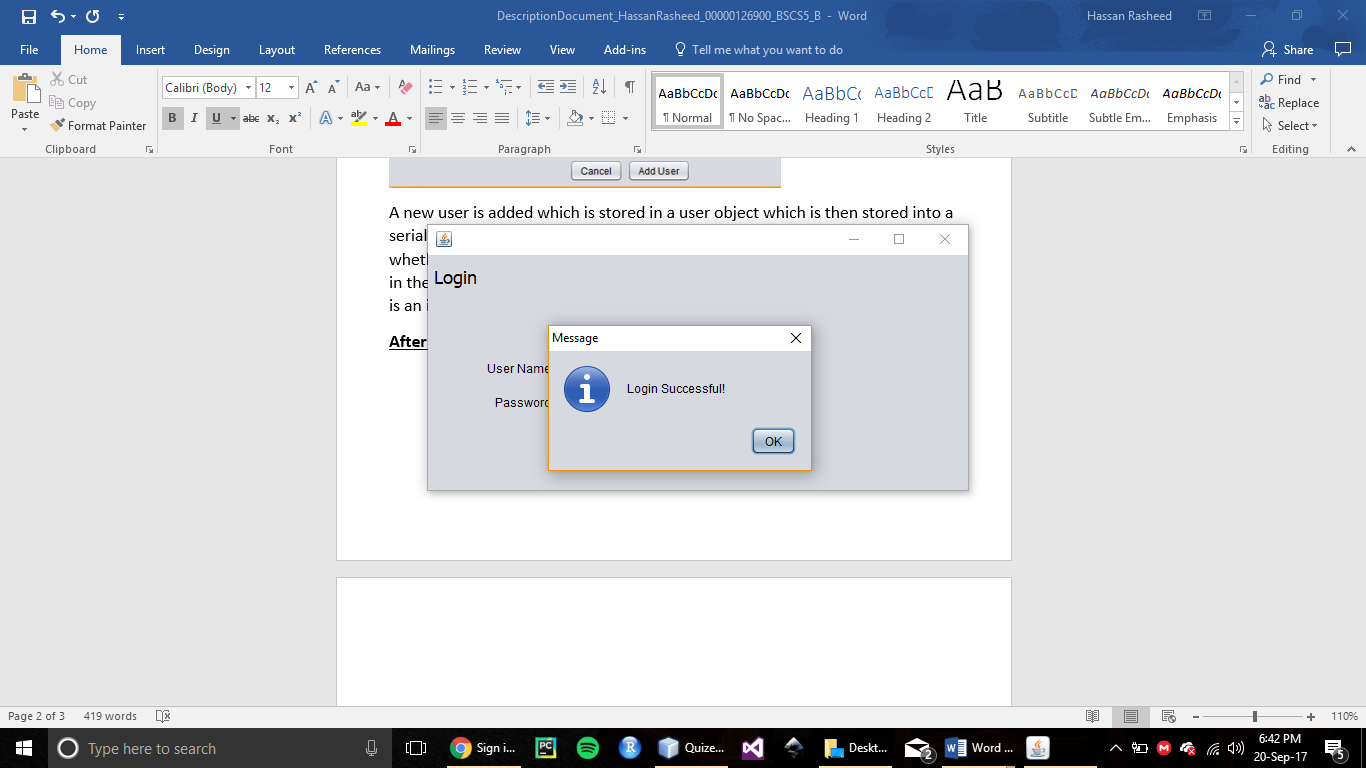
**Add New User**



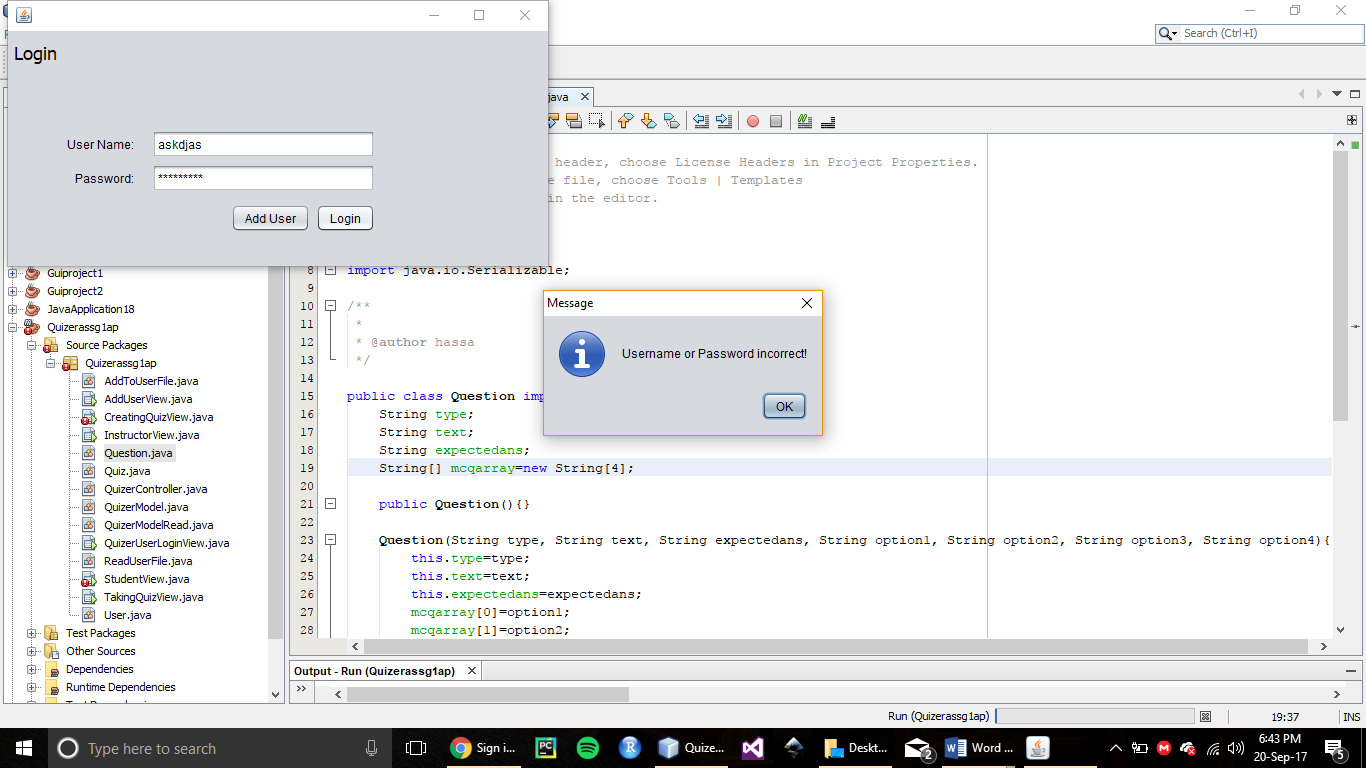
A new user is added which is stored in a user object which is then stored into a serial file, from this file data can later be retrieved to check for login details whether a username matches the password or if they even exist. It also takes in the role so that at login time the application can check if the user logging in is an instructor or a student and appropriate view is opened for each of them.

**Logging In**

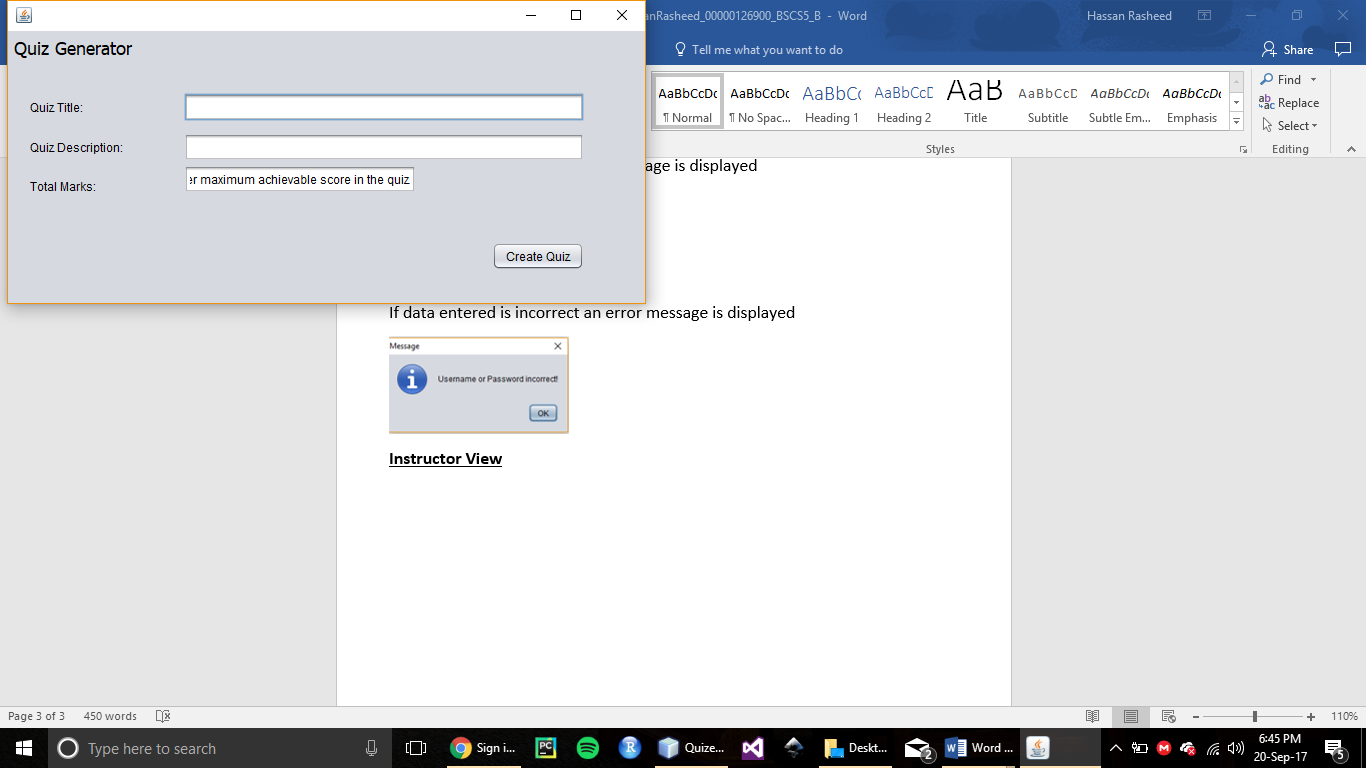
Message dialogues are displayed according to the details entered by the user if details are correct verification message is displayed



If data entered is incorrect an error message is displayed

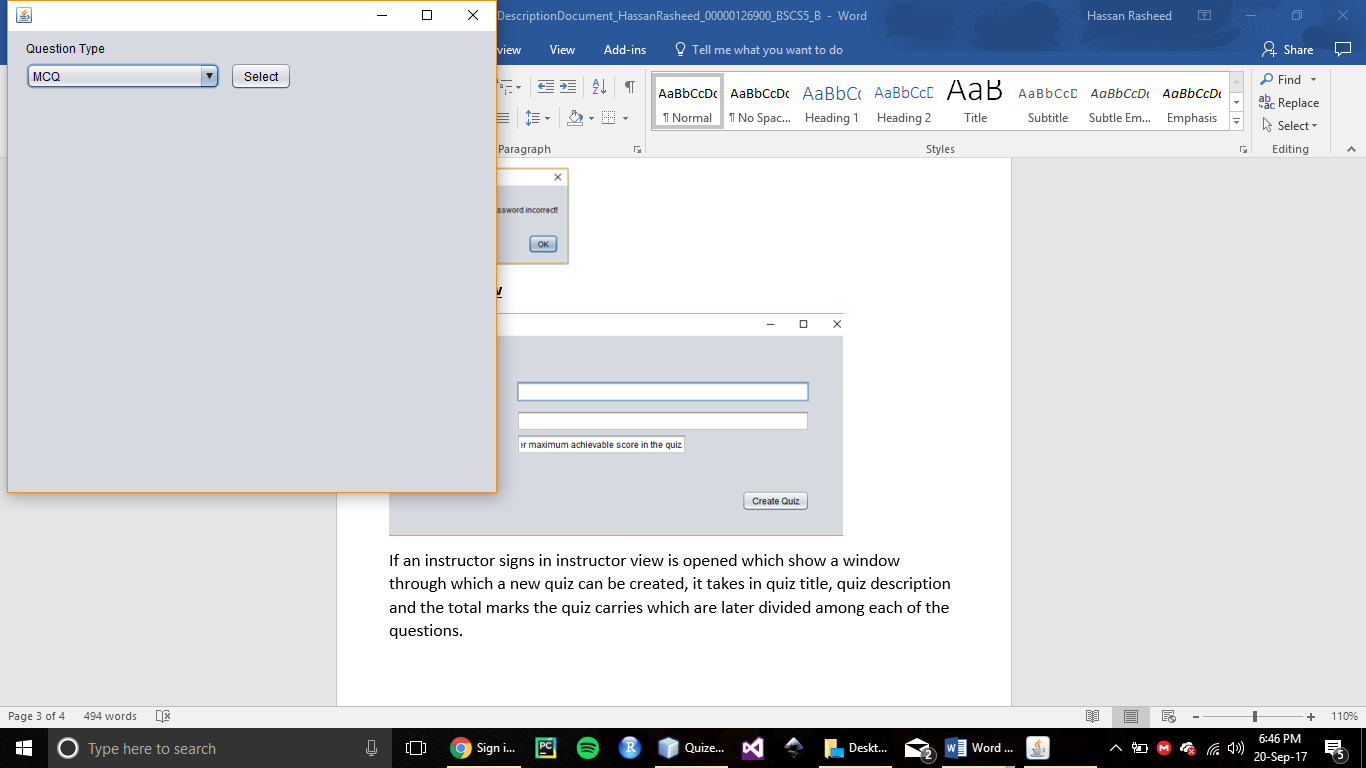


**Instructor View**

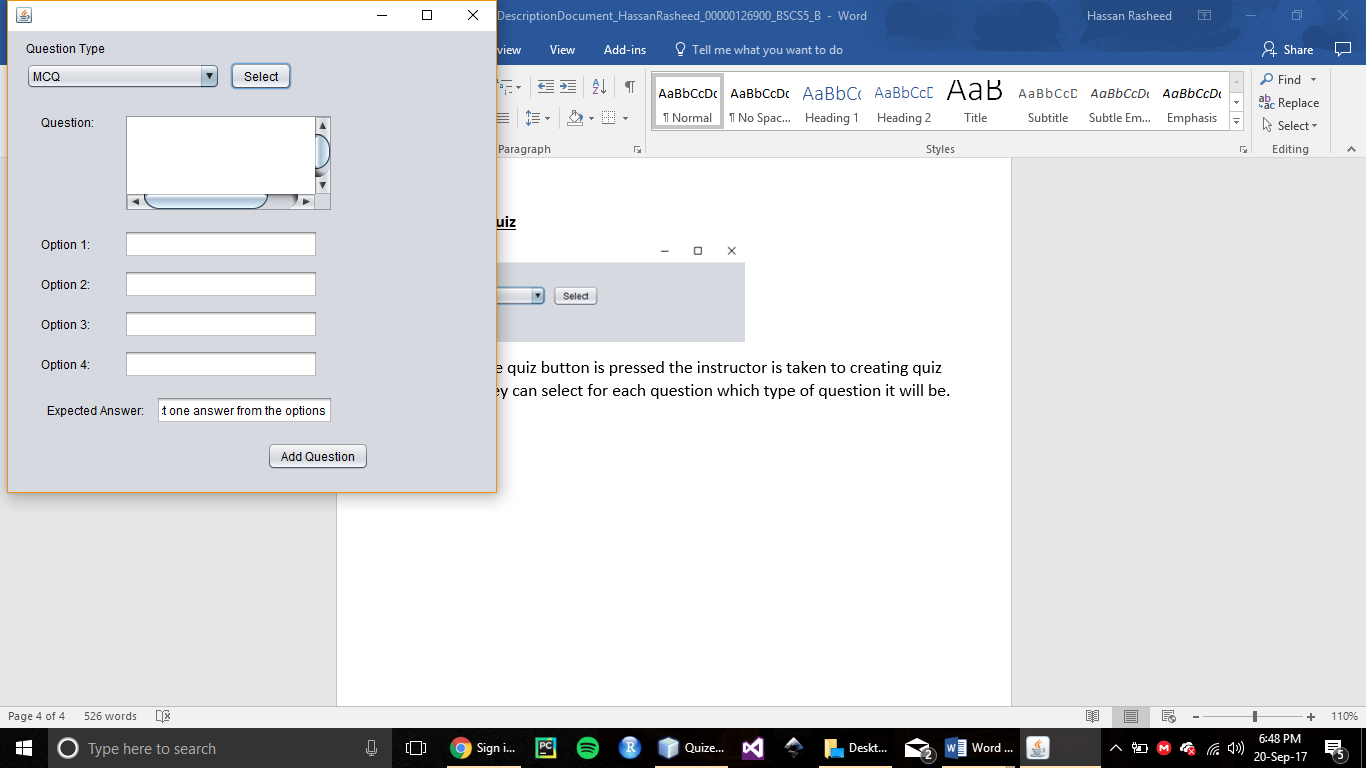


If an instructor signs in instructor view is opened which show a window through which a new quiz can be created, it takes in quiz title, quiz description and the total marks the quiz carries which are later divided among each of the questions.

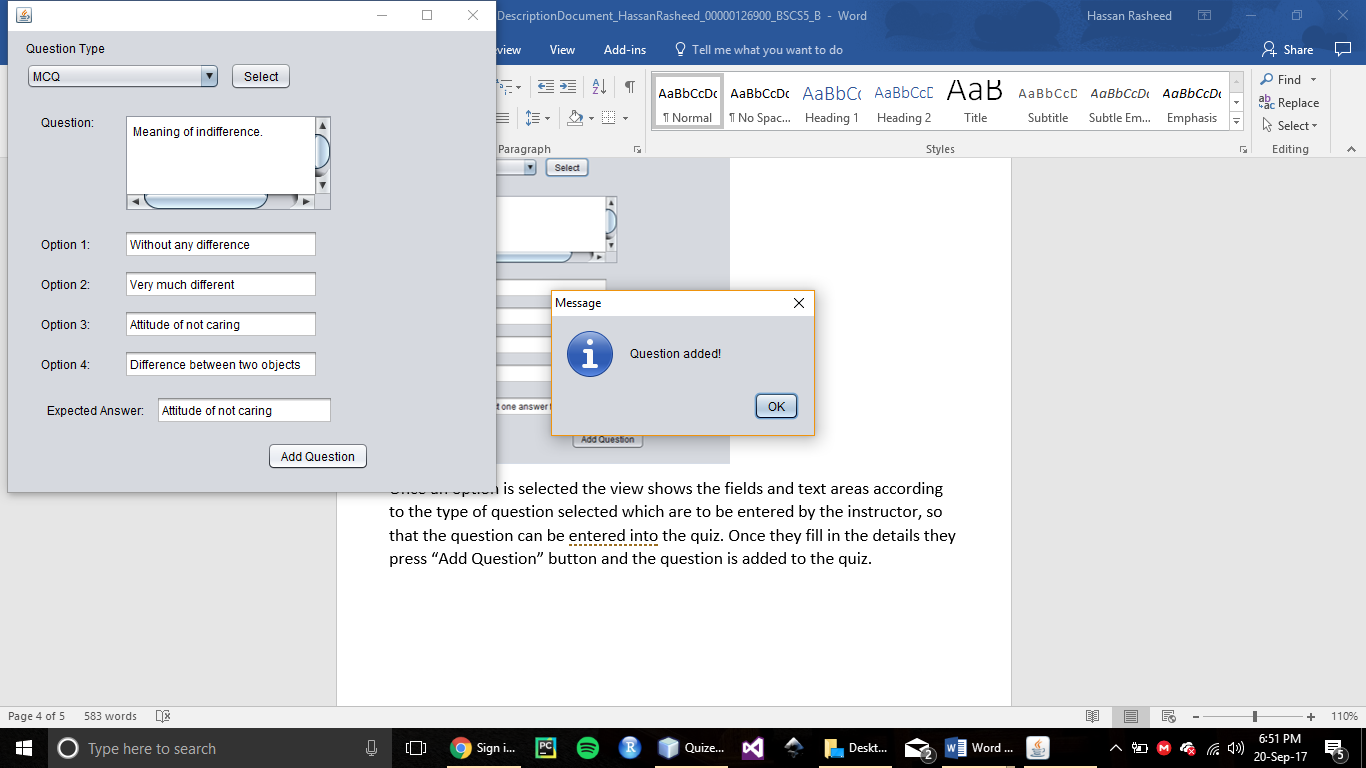
**Creating the Quiz**



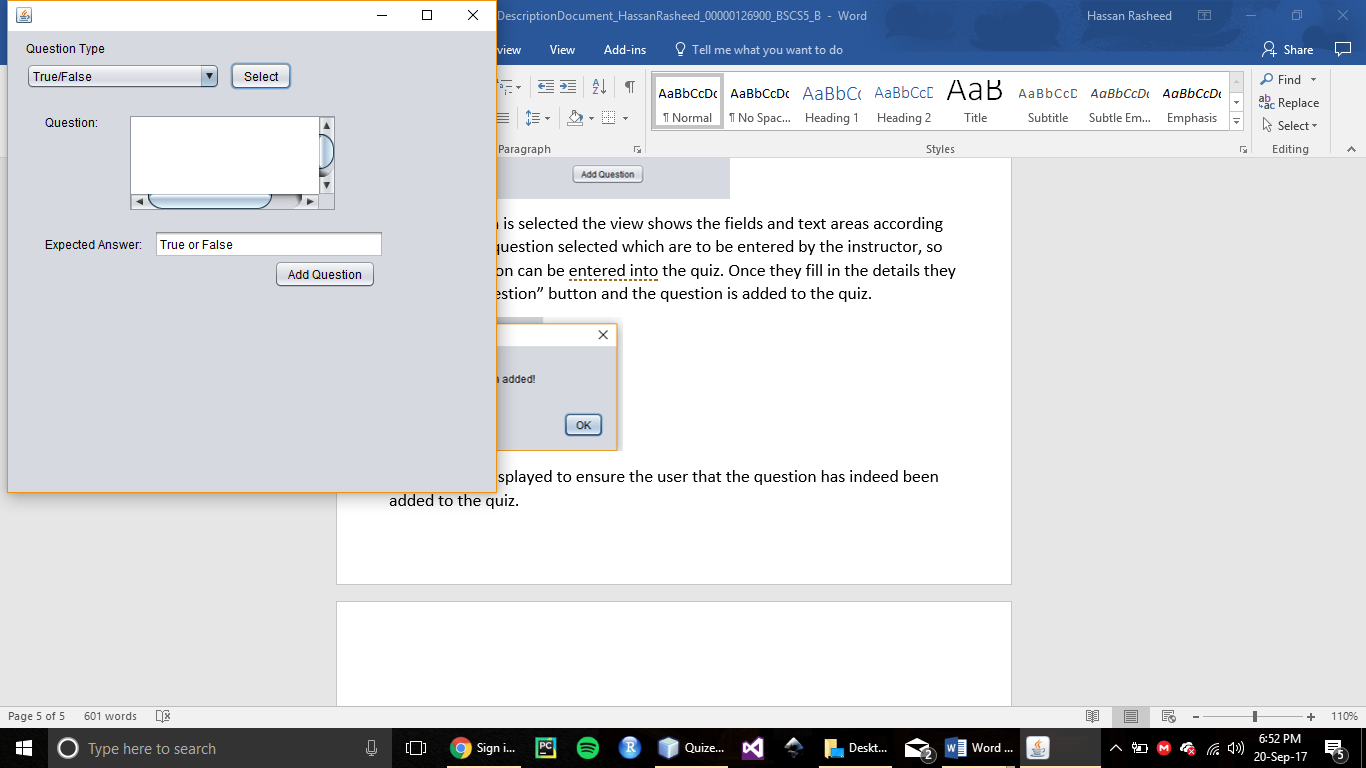
After the create quiz button is pressed the instructor is taken to creating quiz view where they can select for each question which type of question it will be.

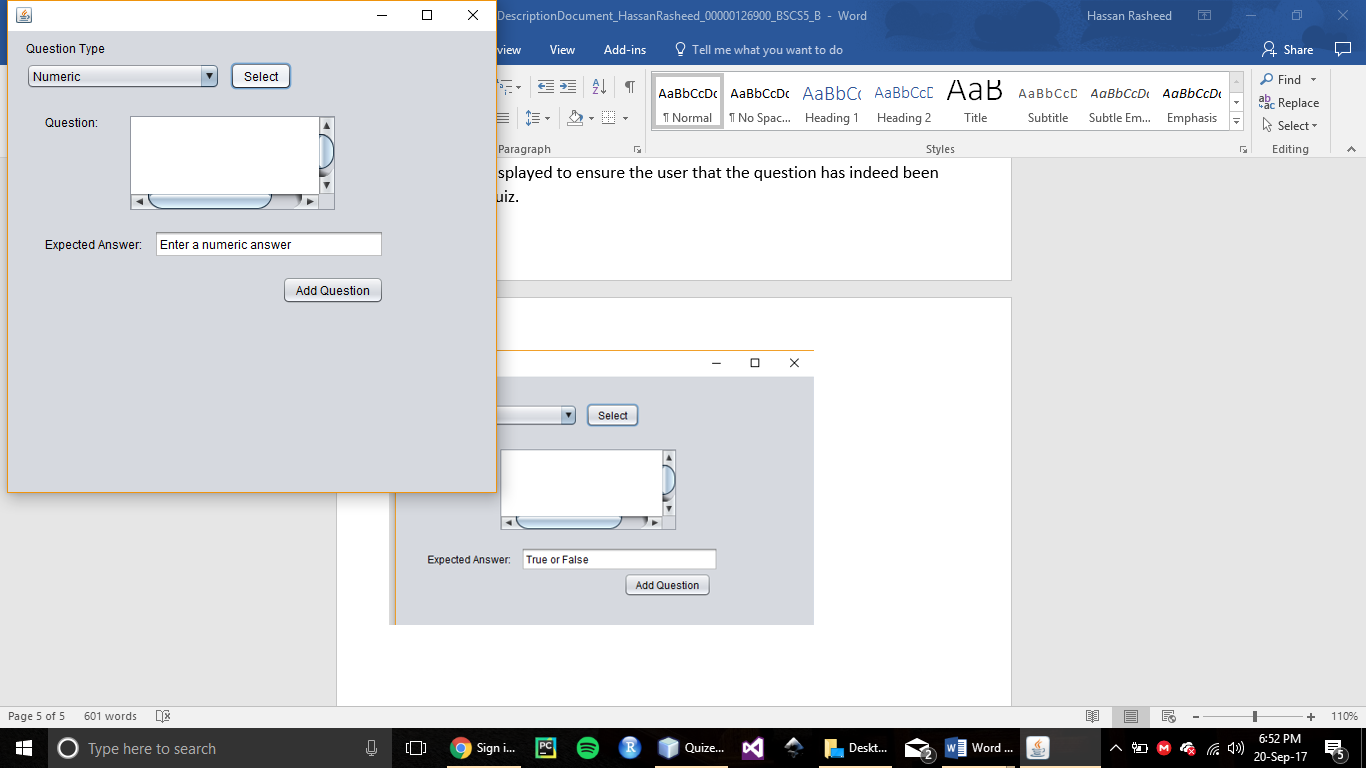


Once an option is selected the view shows the fields and text areas according to the type of question selected which are to be entered by the instructor, so that the question can be entered into the quiz. Once they fill in the details they press “Add Question” button and the question is added to the quiz.

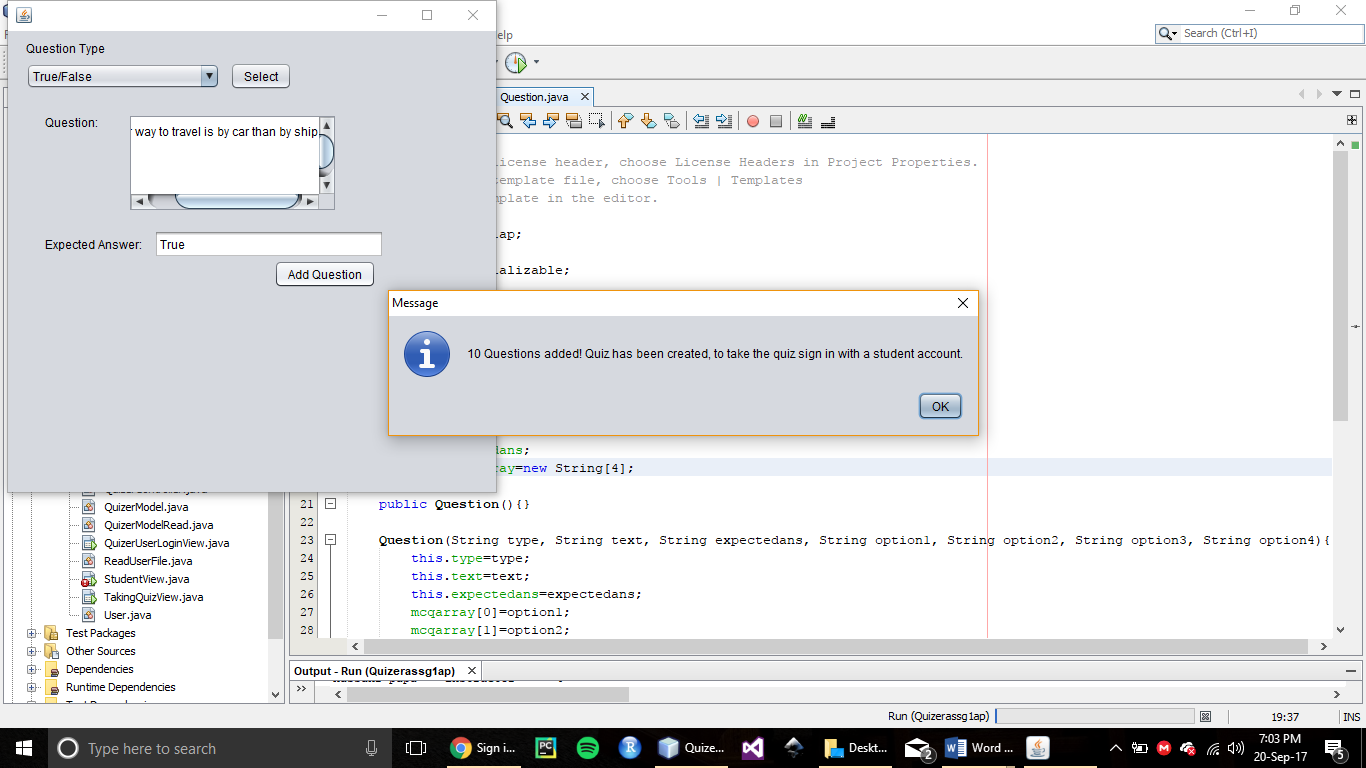


A message is displayed to ensure the user that the question has indeed been added to the quiz.

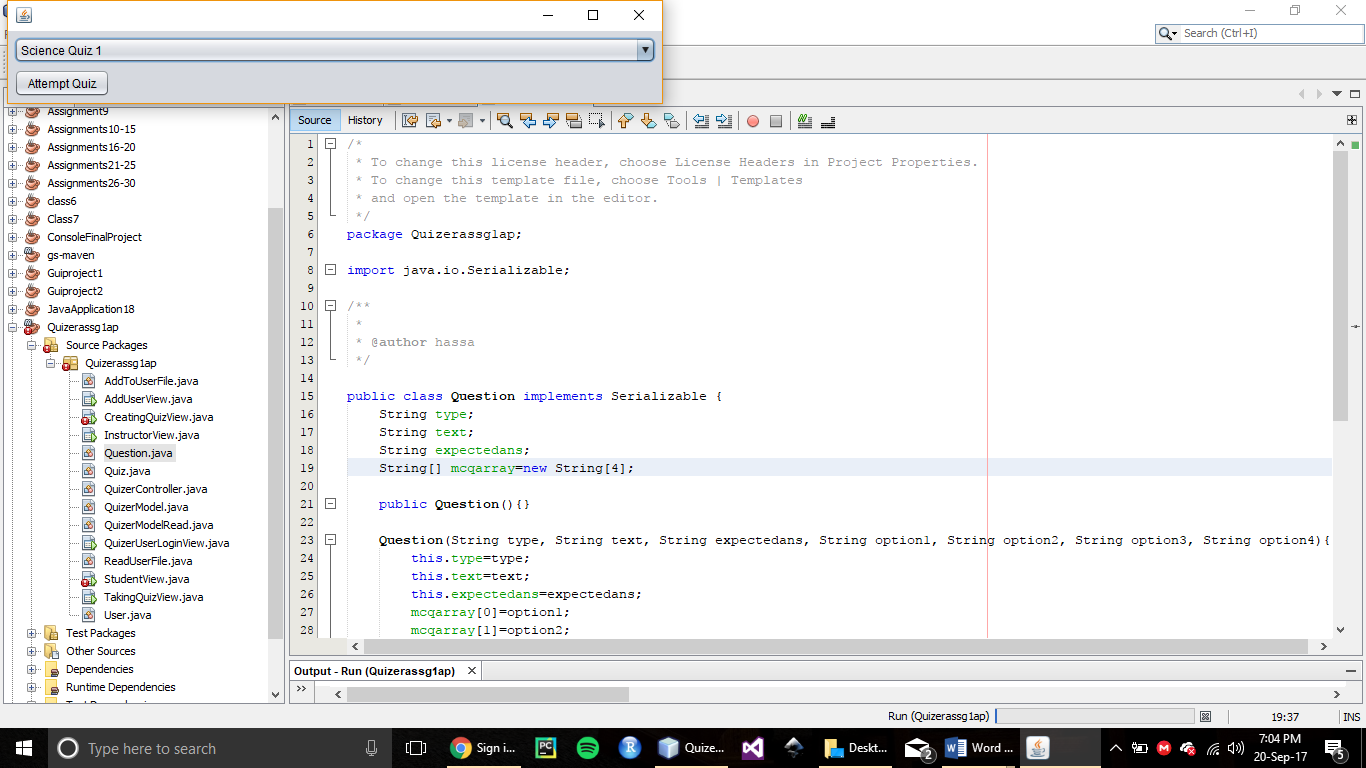


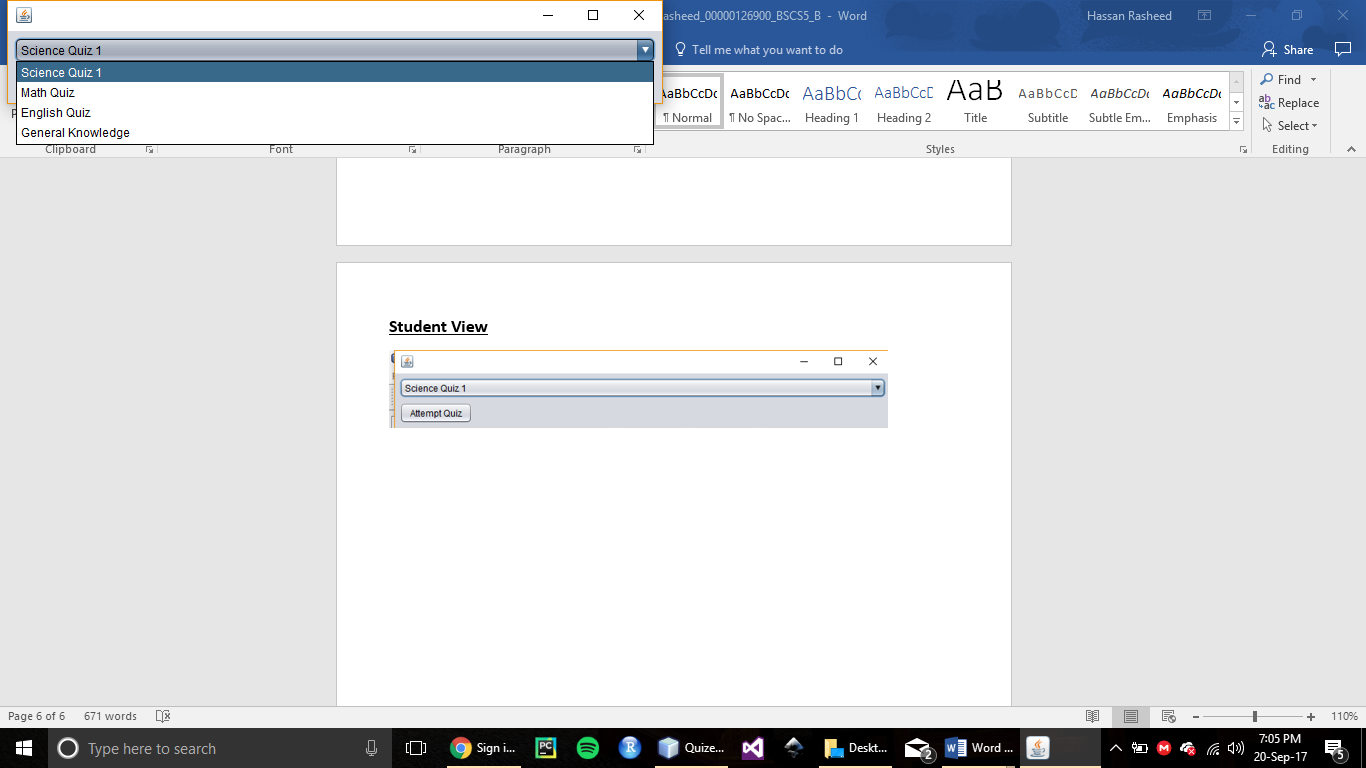


In a similar way the users can select question types of numeric or true/false, for each the view is different and the fields in which data is to be entered are also not the same.

  
After 10 questions are added (a limit set due to the requirements of the assignment) a message is shown that the quiz has been created and can be taken from a student view.

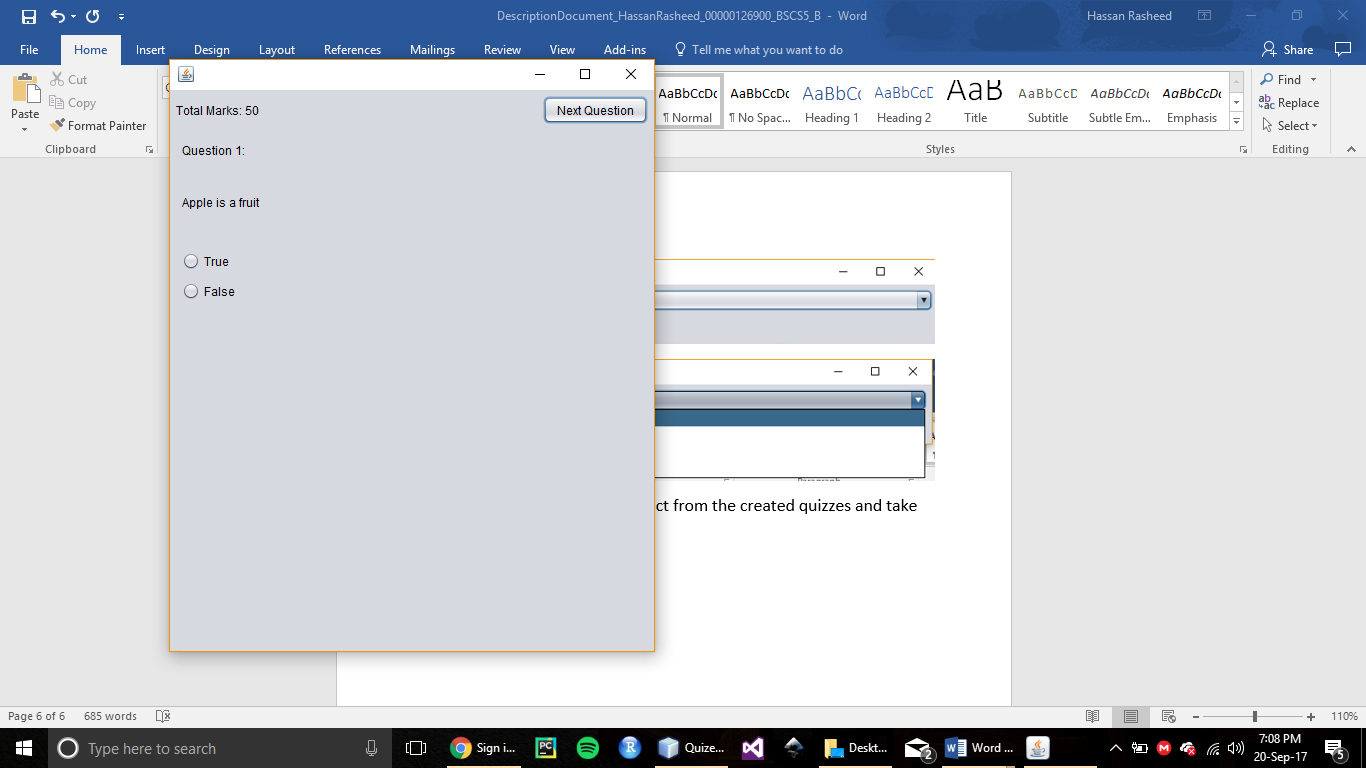
**Student View**





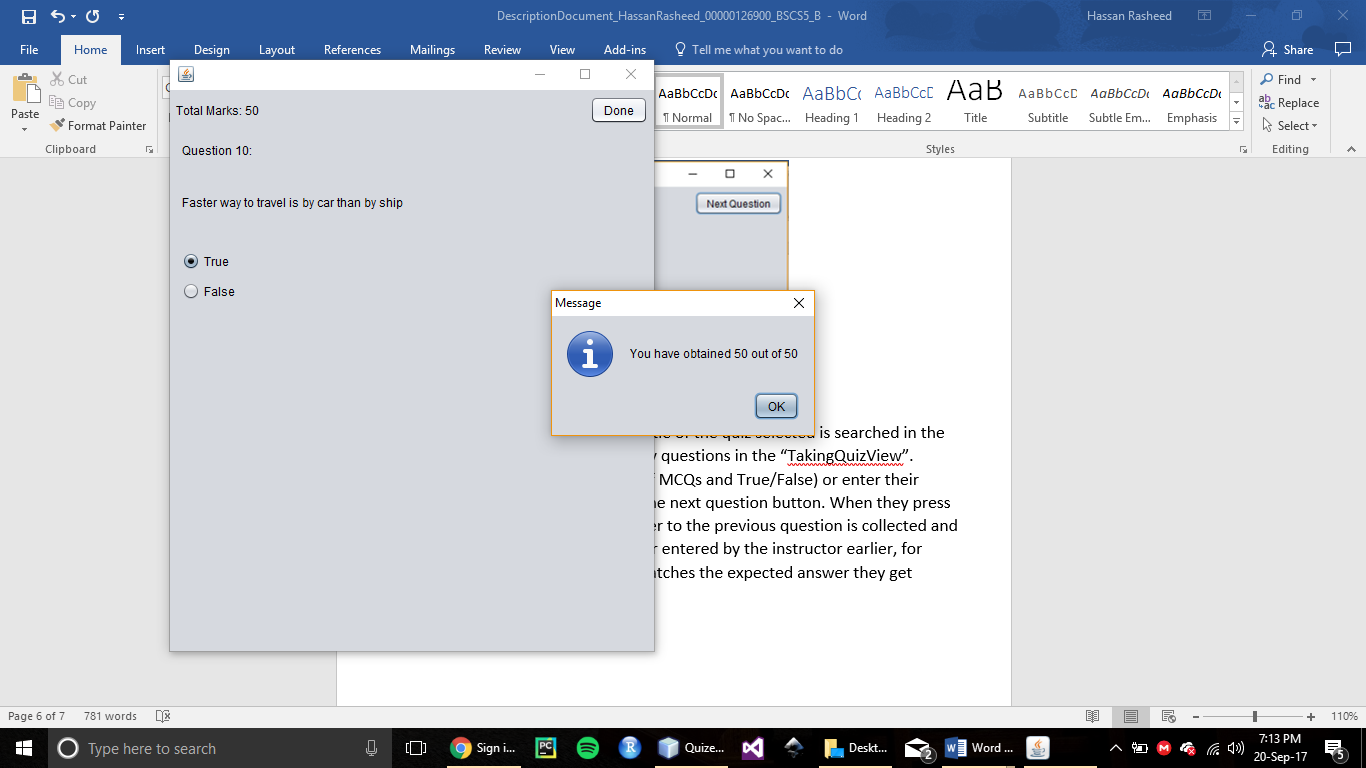
In the student view students can select from the created quizzes and take them.

**Taking the Quiz**

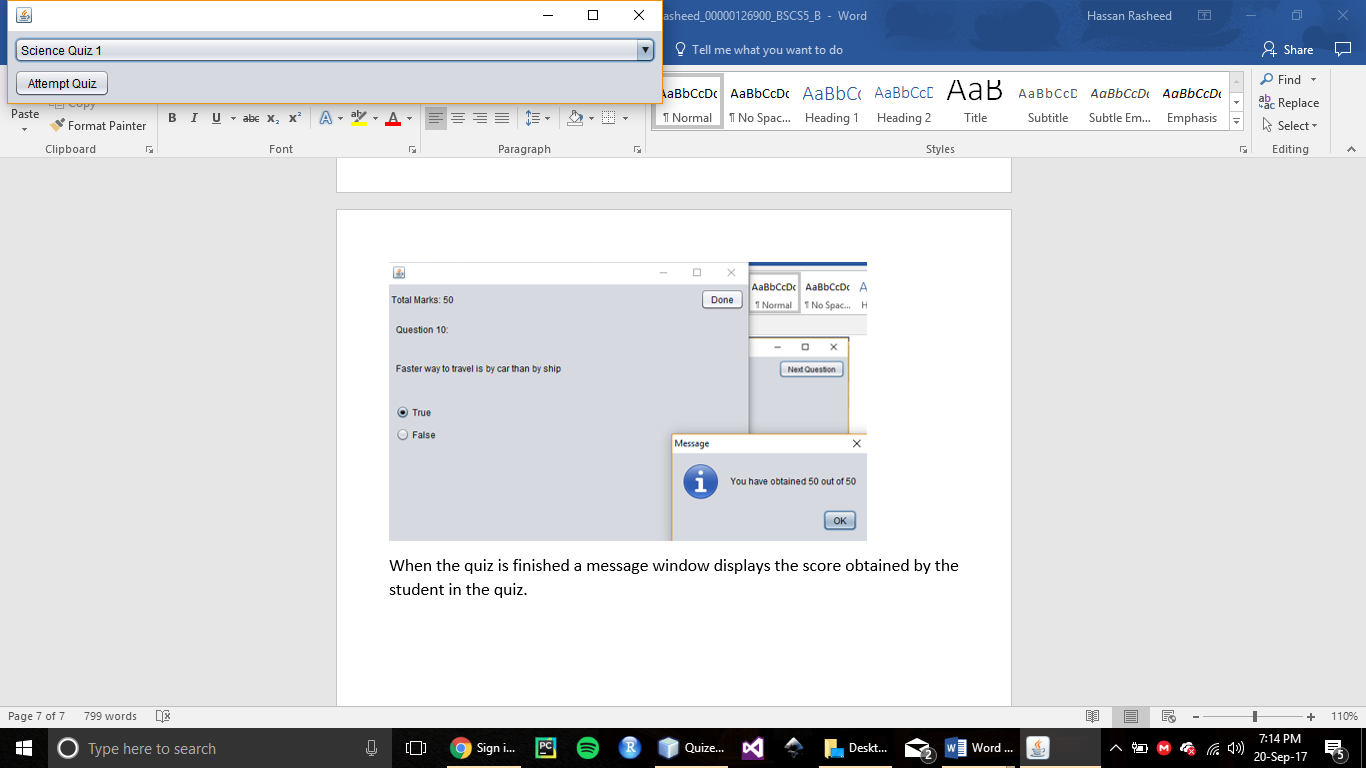


When attempt quiz is pressed. The title of the quiz selected is searched in the serial file and it is retrieved to display questions in the “TakingQuizView”. Students select the option (in case of MCQs and True/False) or enter their answer in the given field and press the next question button. When they press the next question button their answer to the previous question is collected and checked against the expected answer entered by the instructor earlier, for each question if student’s answer matches the expected answer they get marks for the question.

**Quiz is Finished**

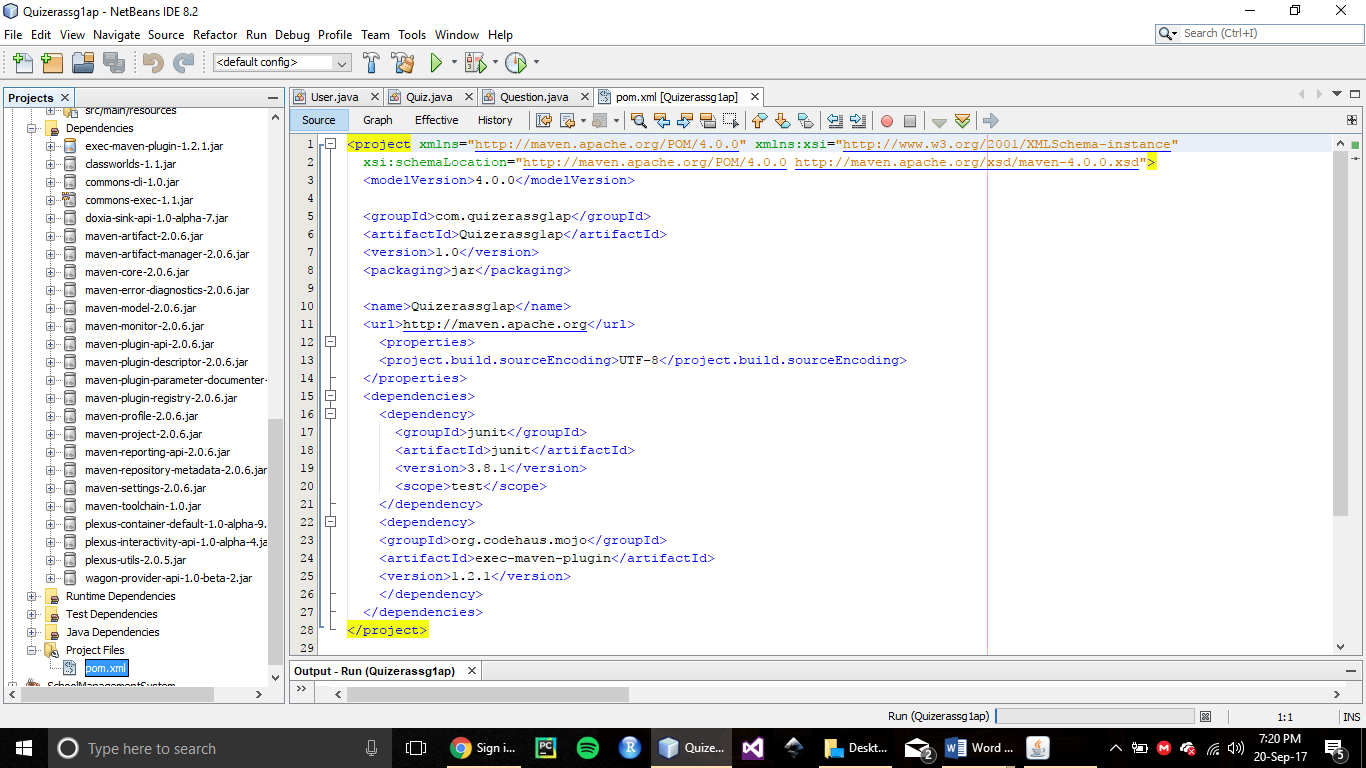


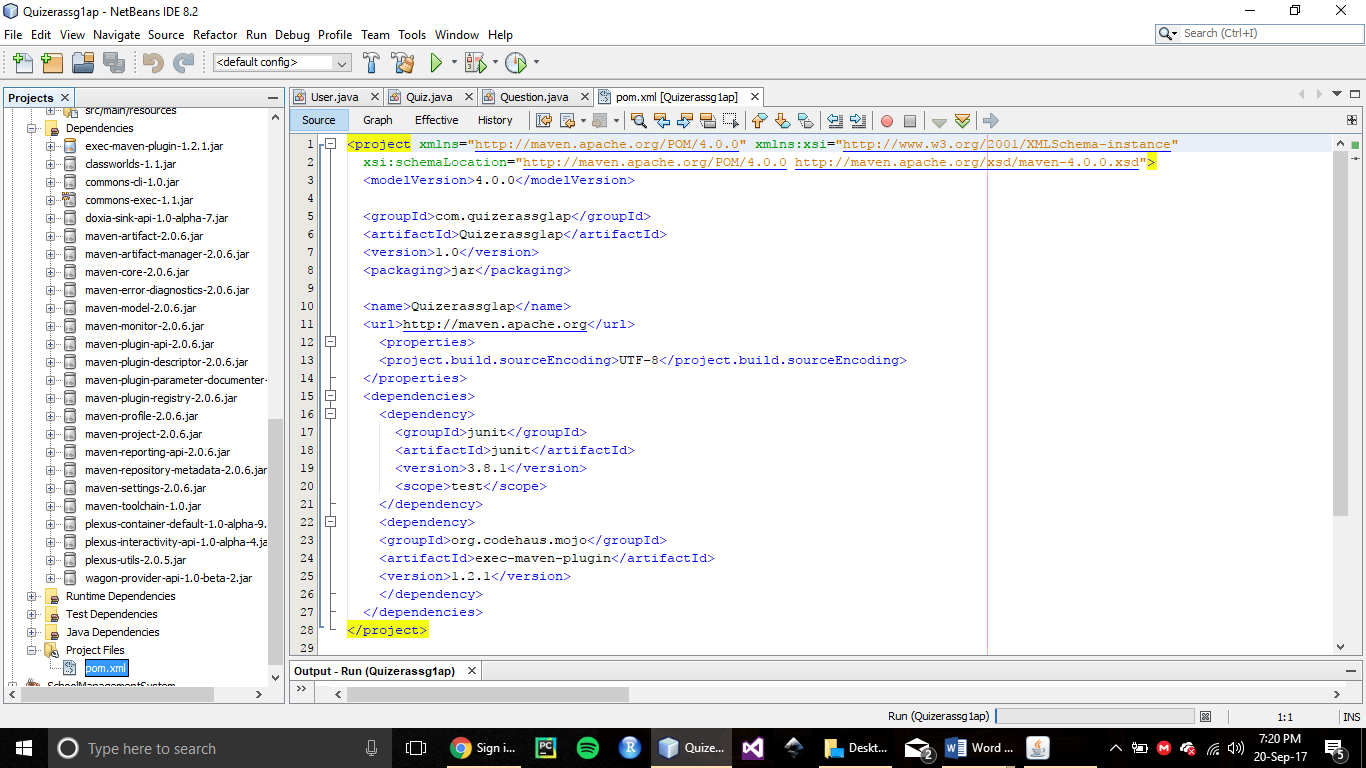
When the quiz is finished a message window displays the score obtained by the student in the quiz.



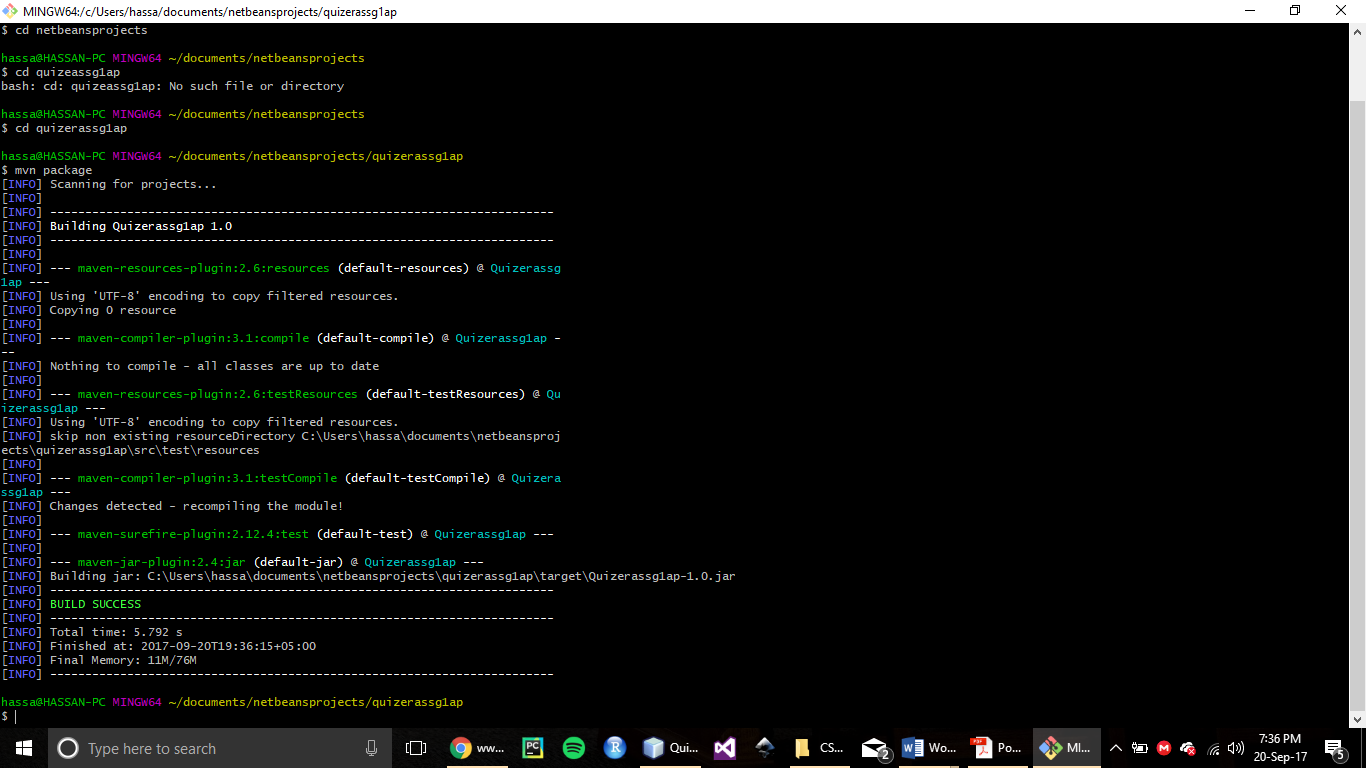
When they press the ok button on the score message they are taken back to the student view window where they can select to attempt another quiz or close the application.

**Maven**





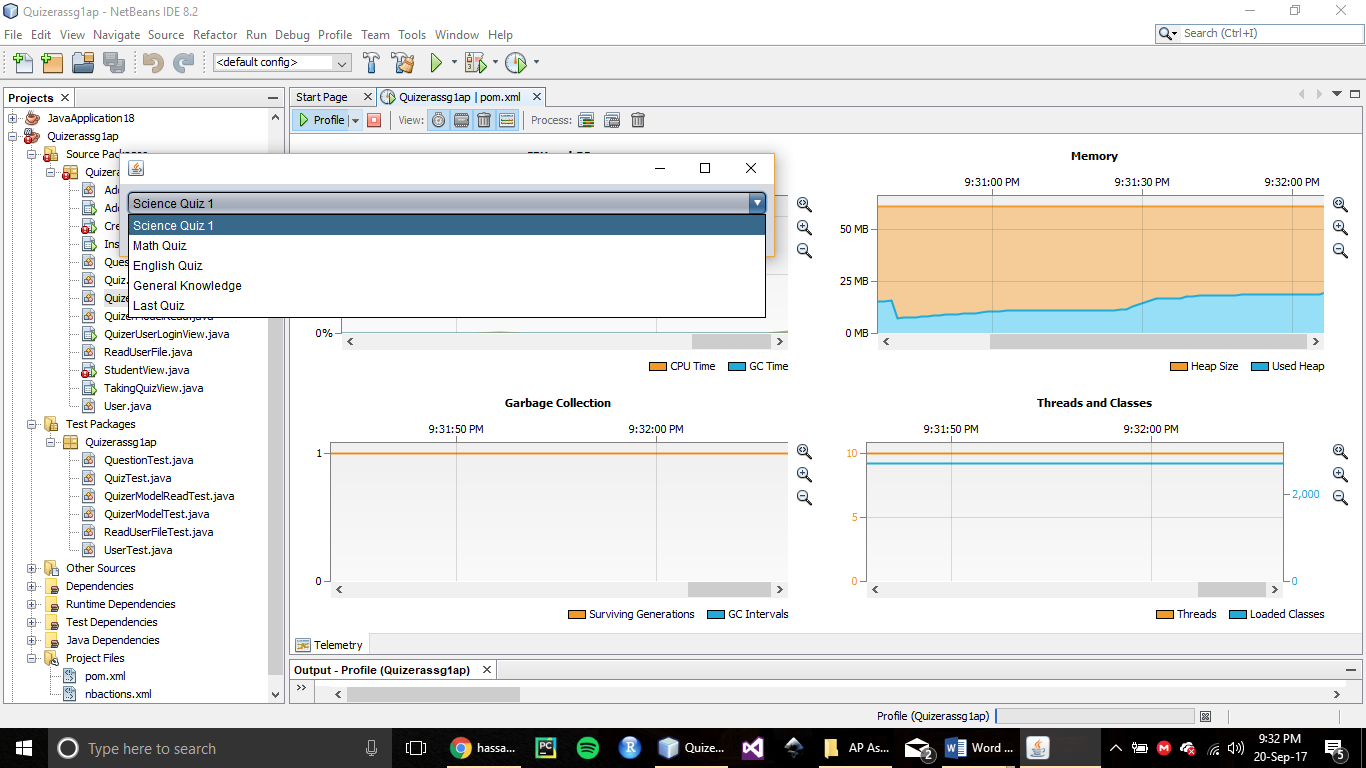
Project is created as a Maven archetype project in NetBeans so the dependencies are included at the runtime (only if they are not already installed) if any are needed.

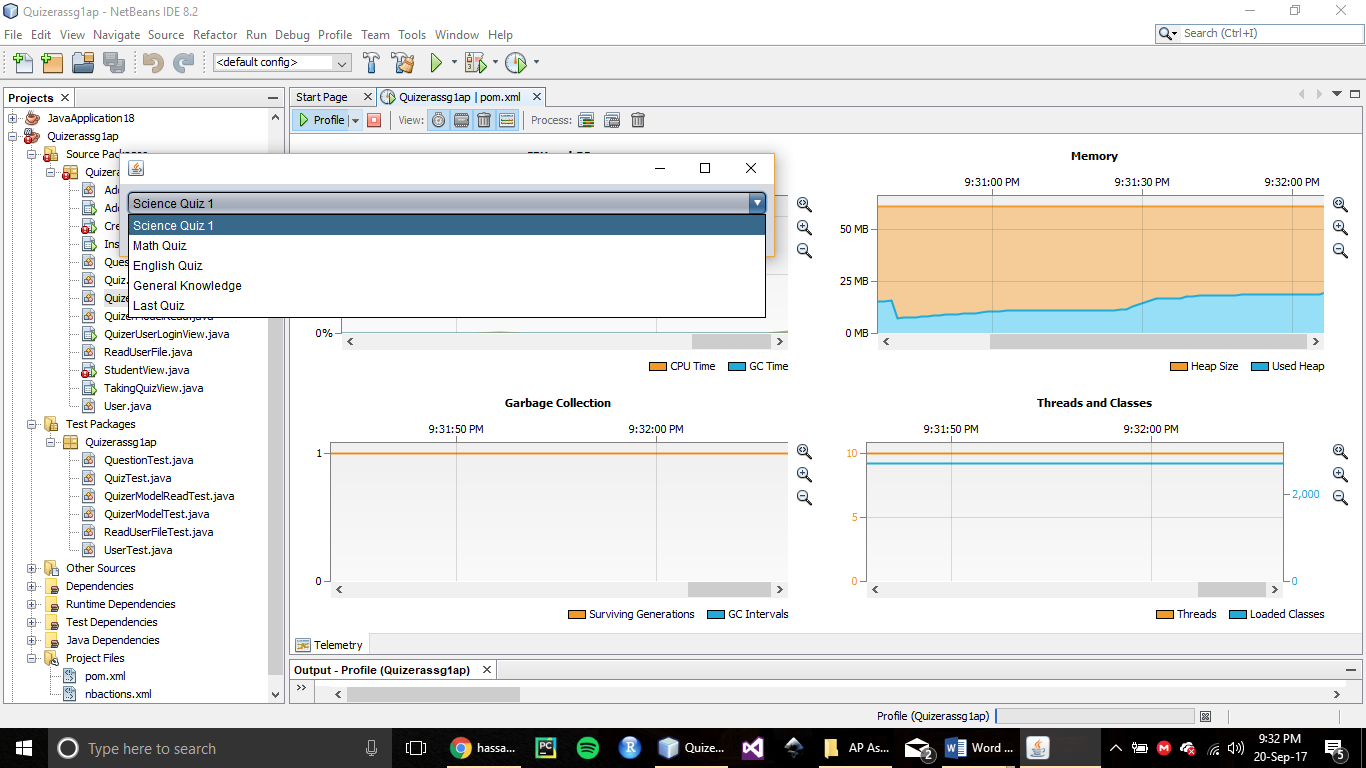


The project is packaged and compiled with maven, no dependencies were needed to be imported manually, all were installed by maven.

**Profiling:**

**Memory Usage after Creation of 5 Quizzes:**

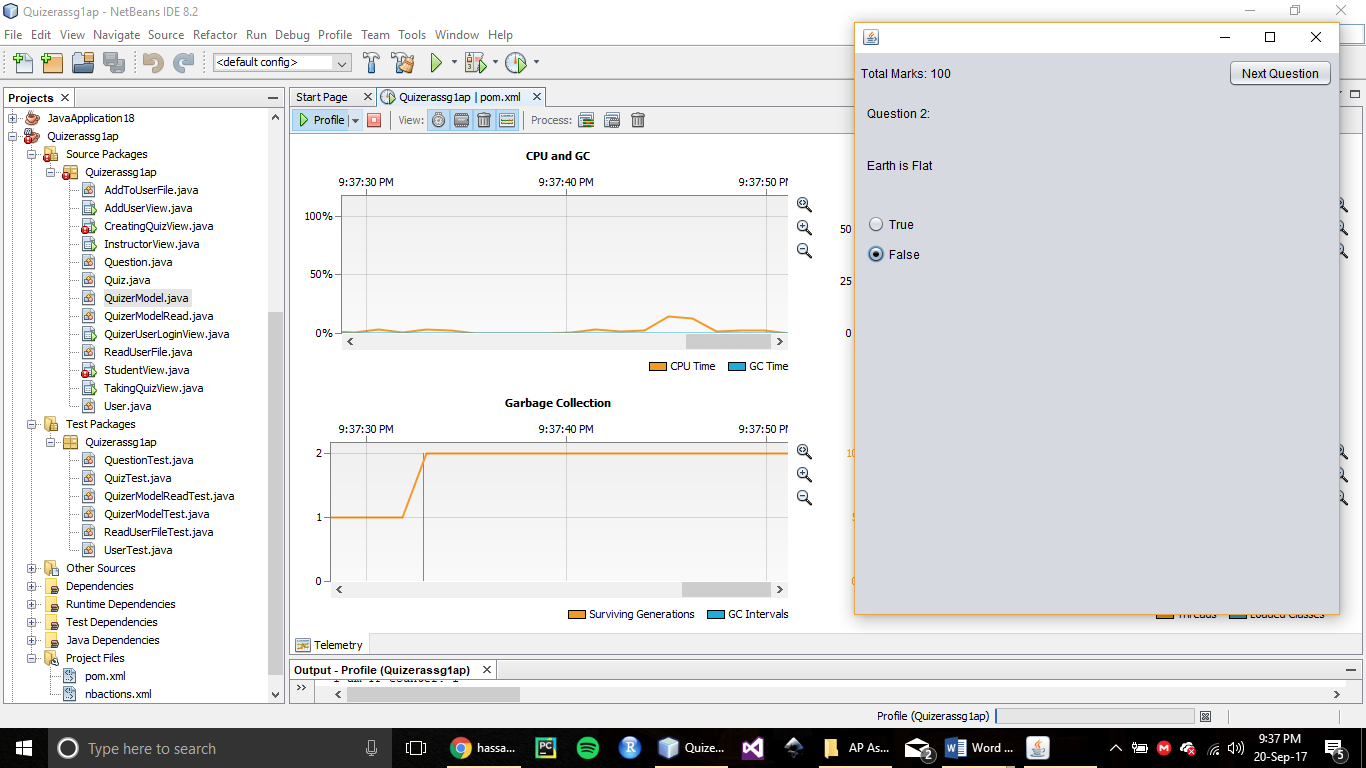


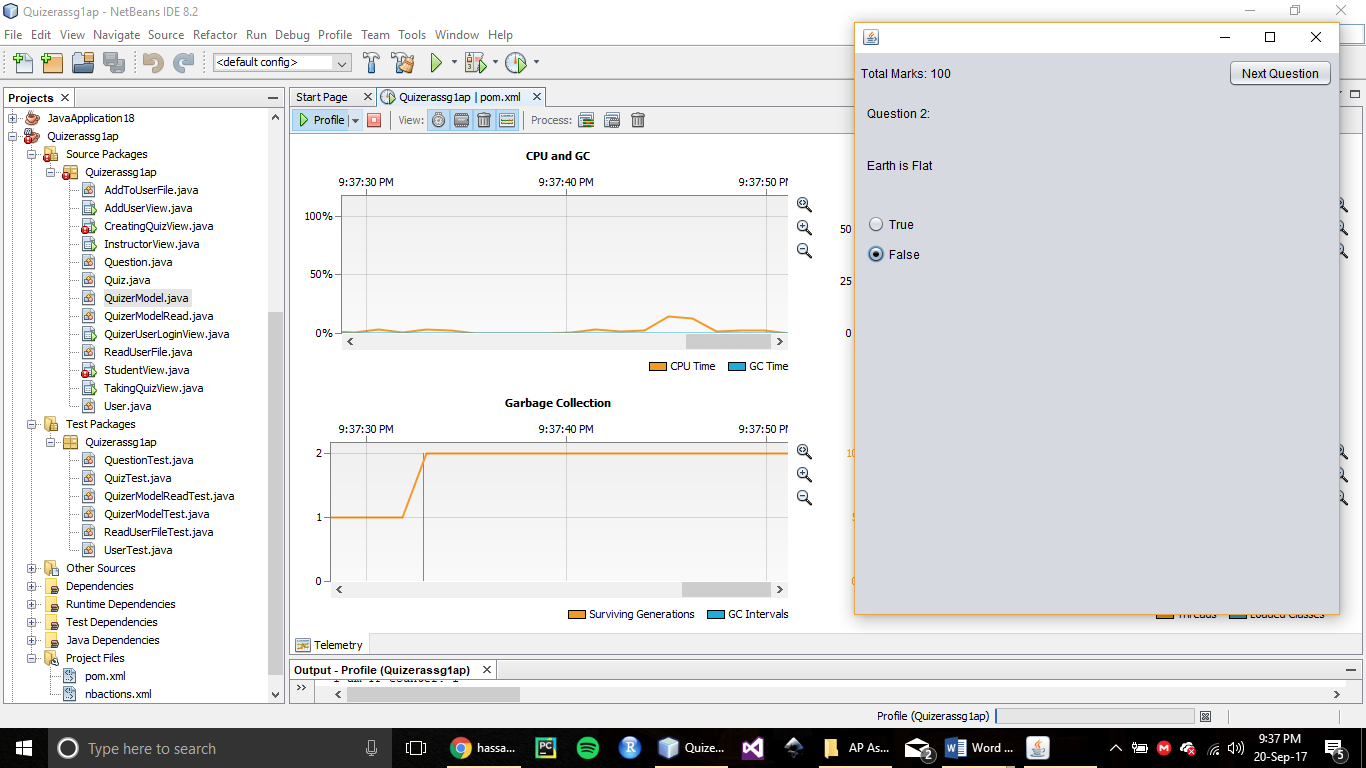


As it is evident from the screenshot that there are 5 quizzes currently added. And because the limit to question was set at 10 so every quiz contains 10 questions each.

Memory heap usage seen from the profiling is around 20MB.

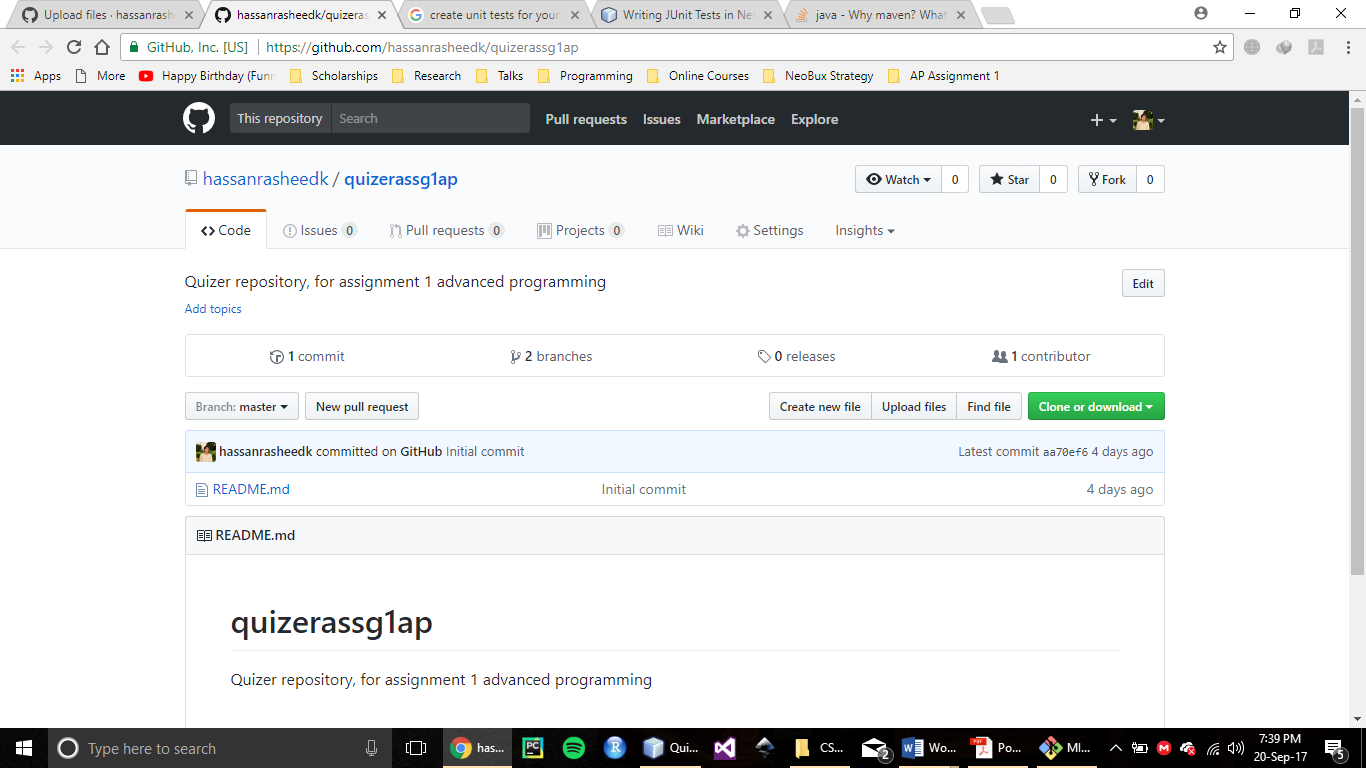
**CPU Usage when a Student is Attempting a Quiz:**



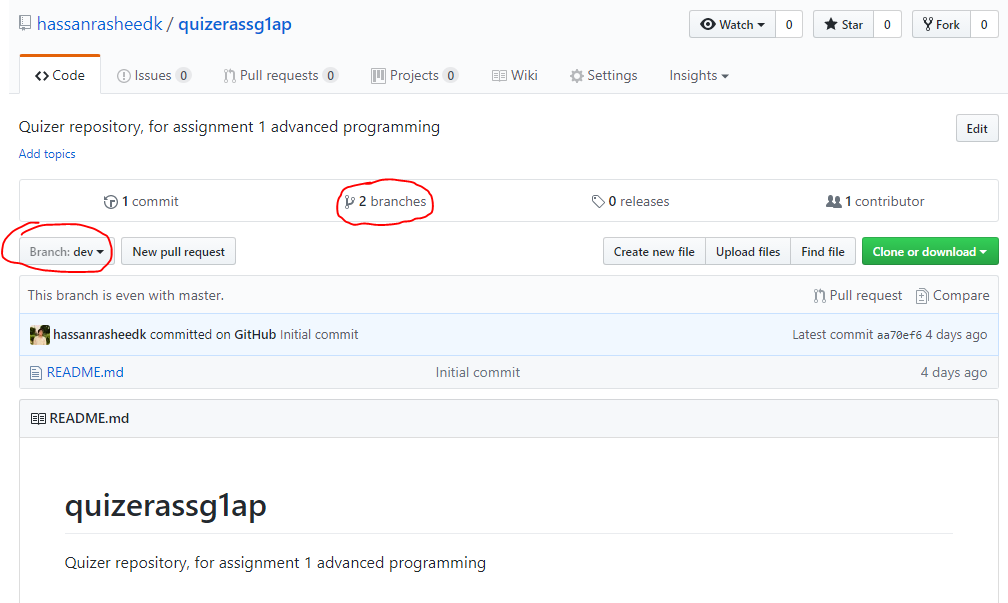


As seen from the profiling results the CPU usage is very low. There is a little hike once in a while, that probably occurs when an answer to a question is submitted.

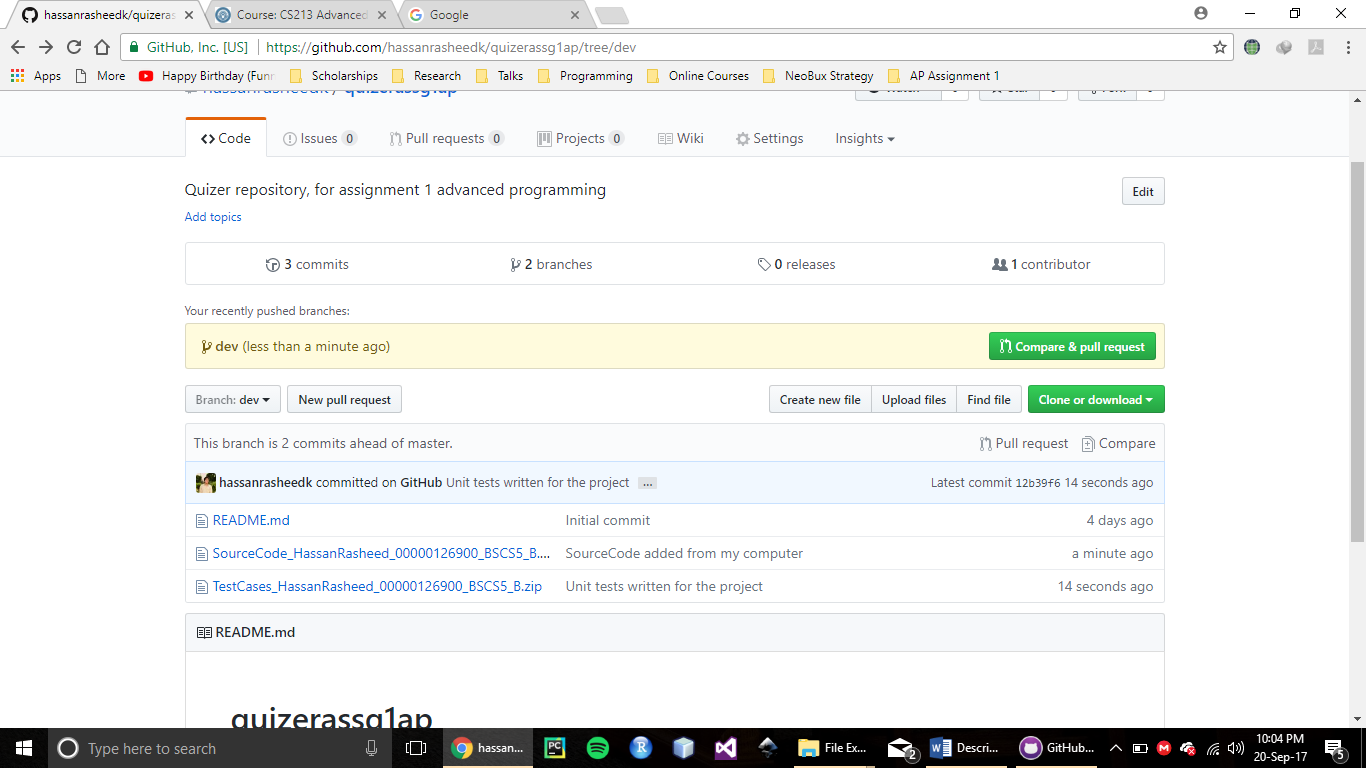
**Github**



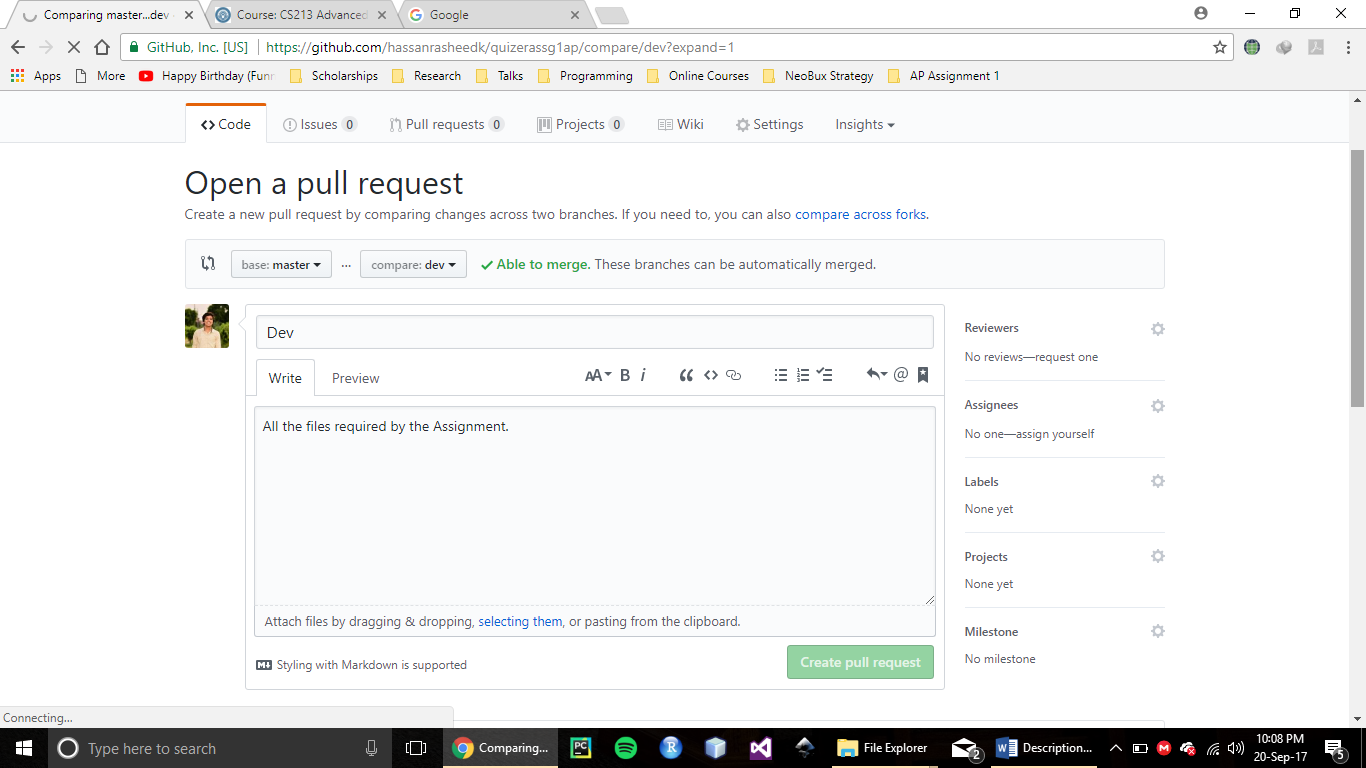
A repository is created for the assignment.

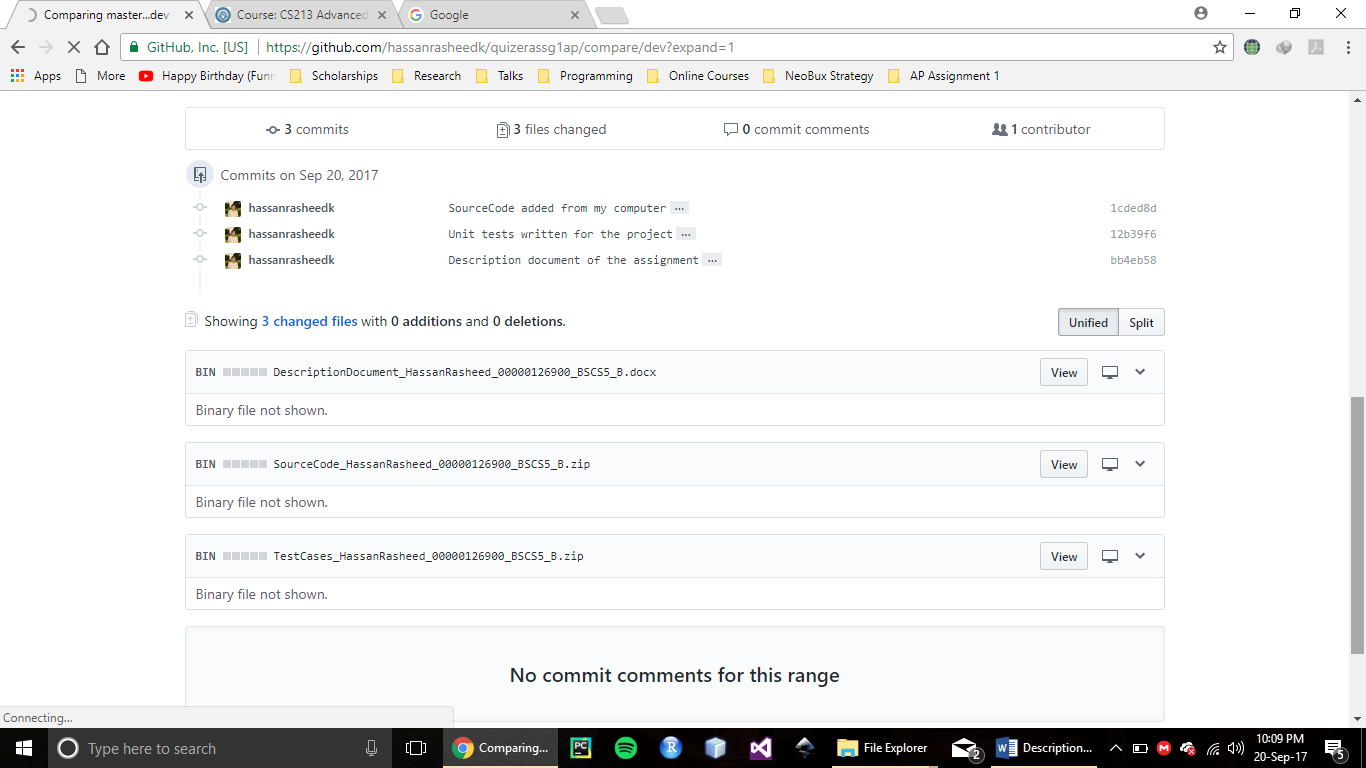


A dev branch is created as required by the assignment.

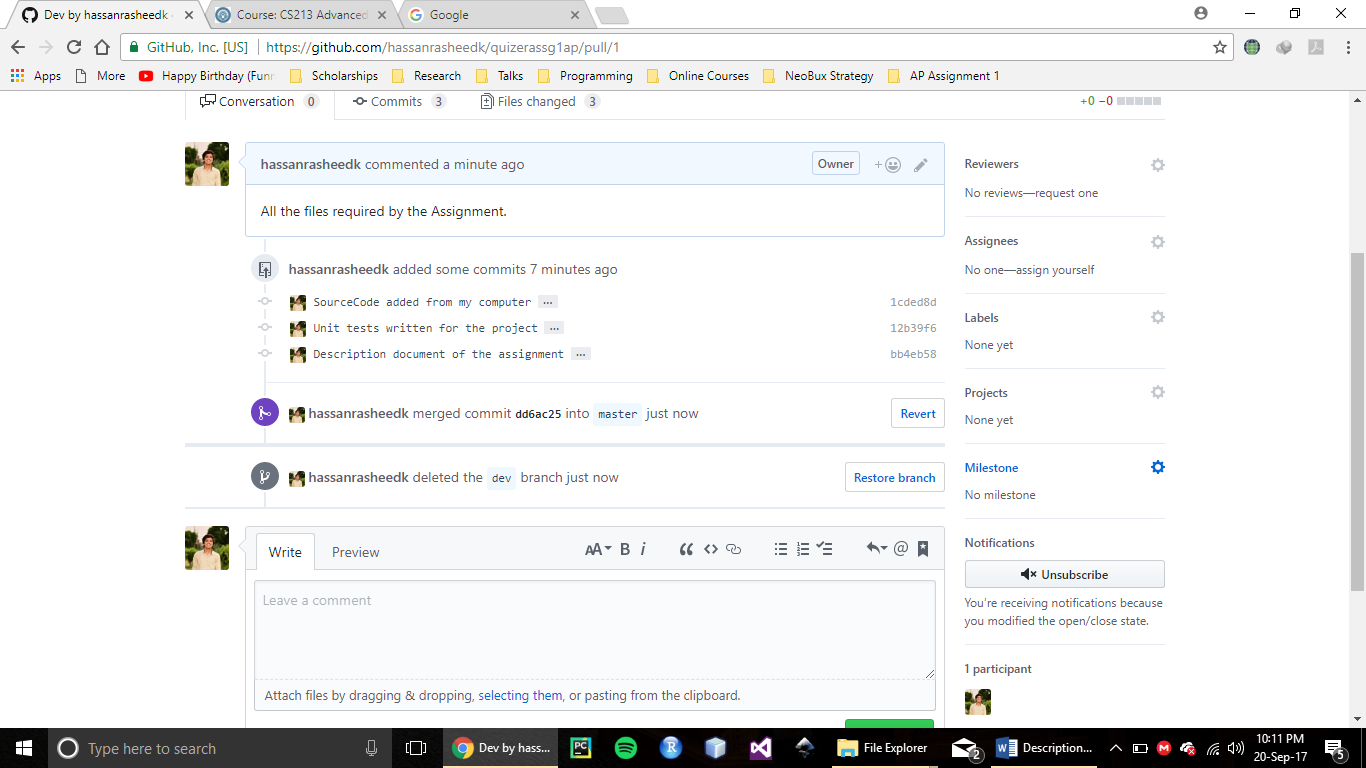


Source code and Test cases are uploaded on the dev branch.





Pull request has been generated by uploading the source code, test cases and description document.



The dev branch is merged with the master branch!

GitHub Link of the Project:

<https://github.com/hassanrasheedk/quizerassg1ap/>