

**Programming Module Code:*CP40061E***

***Course Name: Cyber Security***

**Student ID: 21363052**

**Module Leader: Fehmida Mohamedali**

**Element 2**

Guestbook v0.1®

**Element 2**

***Guestbook v0.1®***

**Programming Module Code:*CP40061E***

***Course Name: Cyber Security***

**Student ID: 21363052**

**Module Leader: Fehmida Mohamedali**

Table of contents

[2 Introduction 2](#_Toc533040865)

[3 The User Guide for the Application 3](#_Toc533040866)

[4 Class Diagram 5](#_Toc533040867)

[5 Examples 5](#_Toc533040868)

[6 Purpose of each class 5](#_Toc533040869)

[7 Further Discussion and Future Plans for the Application 5](#_Toc533040870)

[8 References 6](#_Toc533040871)

[9 Appendix 6](#_Toc533040872)

[10 6](#_Toc533040873)

# Introduction

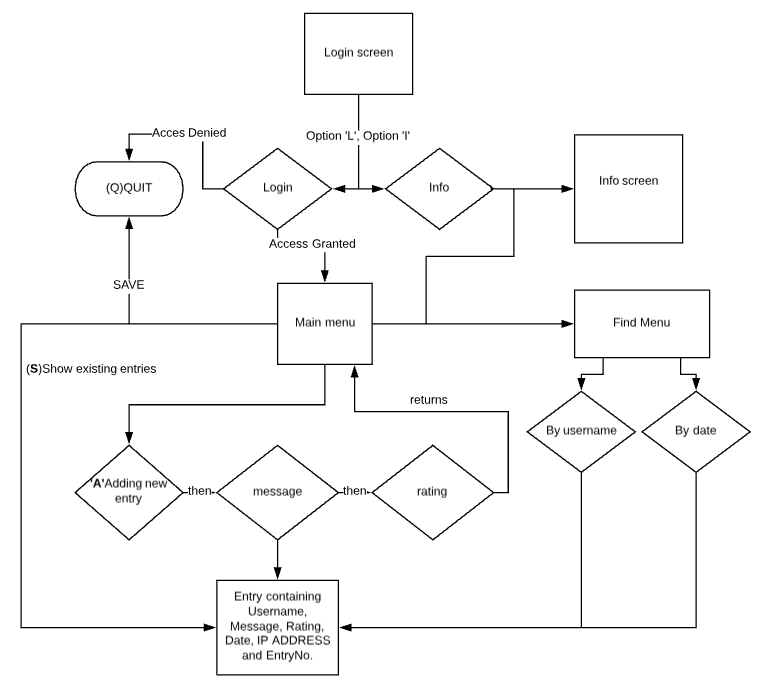
*The purpose of that application is, as its name says, to store entries from guest users in nice form of guestbook with easy to follow up menu driven application.*

*End user can store new entries, view old ones and find the entries throughout using the application, and have own login account.*

**There are 6 main options for this application (ordered by start of application)**

1. ***Login*** *– In order to get into main menu, user is prompted to enter a username and password, by default it is* ***test/test*** *(entering once wrong brings* ***Access denied error****)*
2. ***Info*** *– that option contains basic information about the version of the application.*
3. ***Add new entry –*** *its option to add new entry, by entering username, message(content), and rating guestbook option (from 1 to 10) then it goes back into main menu.*
4. ***Show existing entries -*** *Its printing all of the existing entries that are stored inside the application.*
5. ***Find an entry –*** *finds entry by Username , and Date(not finished yet)*
6. ***Quit –*** *quits and application and saves the file with current state*

This application features: User friendly graphical text menu, **Login** **feature**, **Adding**/**Viewing**/**Finding** entry. Entries contain **date**, **entry** **no**, also features like **IP** **Address** that was written from. **Open** and **Save** module is working that saves entries to **log.dat** file. There’s also rating feature that shows the overall rating of the Guestbook for our feedback.

**Menu Diagram**

# The User Guide for the Application

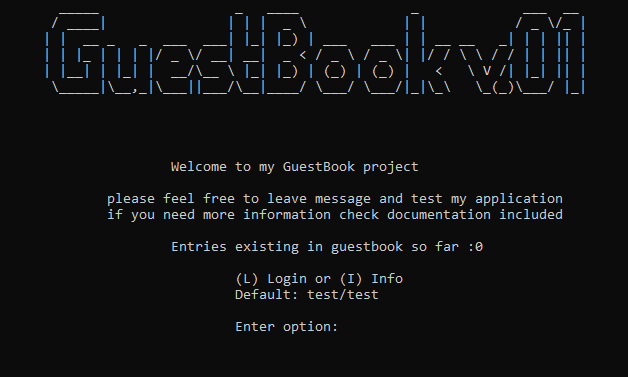
How to start an application

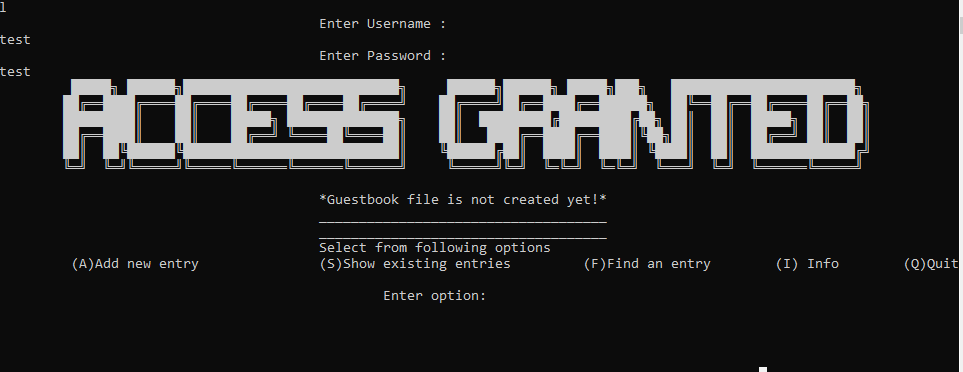
To start application you have to run it through the PowerShell or windows ***cmd***. Its source file is named **Guestbook.jar**.

*Type the following to execute your jar file:*

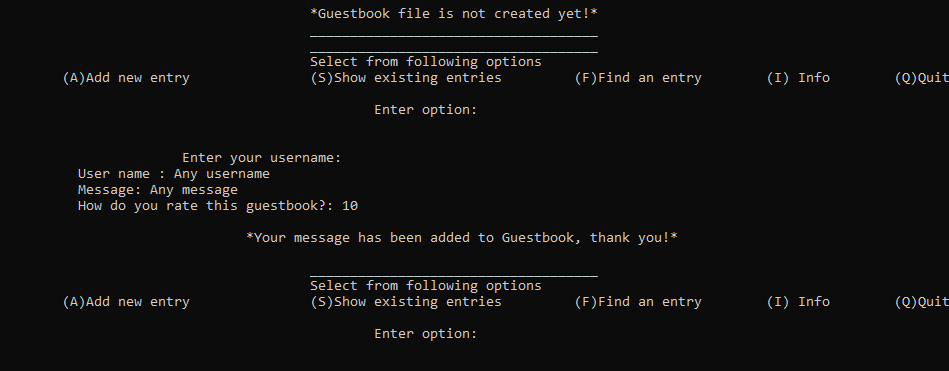
*java –jar Guestbook.jar*

*Note – to access jar file you will have to cd (change directory) to the folder where the jar file is*

When the program is executed, you should see following menu

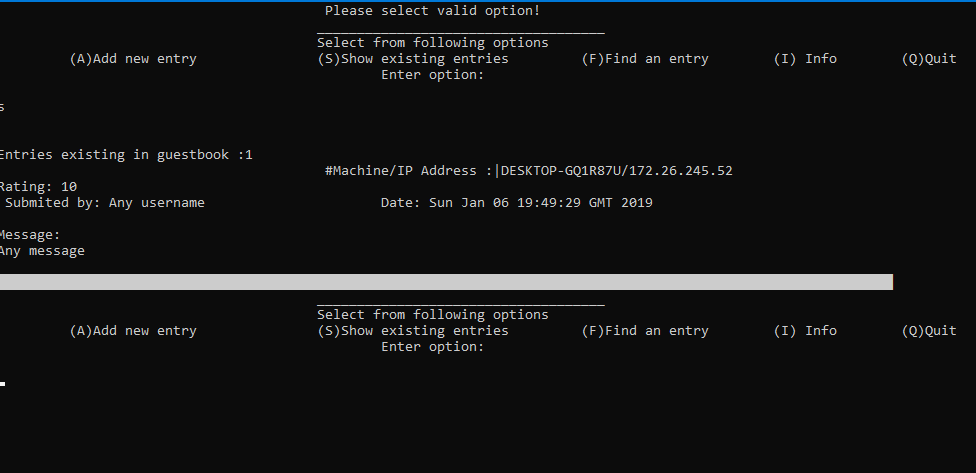
By pressing L button and typing in as a Username: **test** and Password: **test** ,it will brought us to a main menu which looks like this

Available features

1. Add new entry

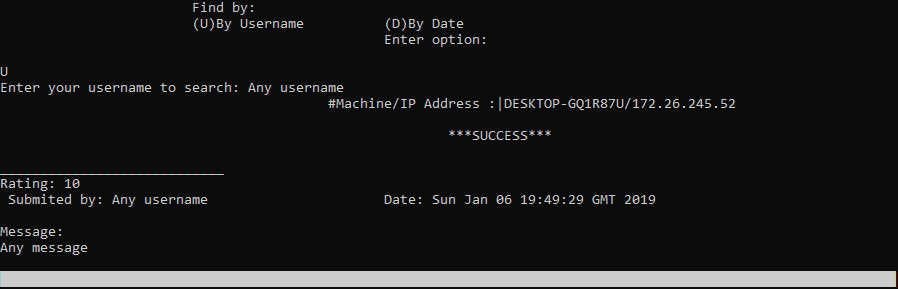
By choosing this option, you will be prompted to enter your **Username, Message,** and **How do you rate this guestbook?** (number from 1 to 10)\*

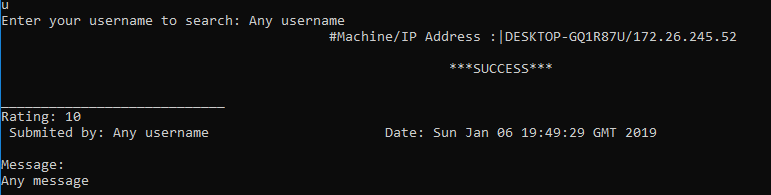
**(S)** Show existing entries

That will allow you to see every single entry that is stored inside Guestbook, you could distinct following information’s

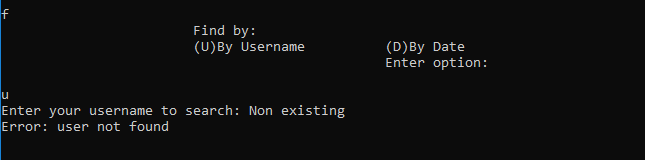
**Entries existing in guestbook, Machine, IP Address, Rating, Submitted by, Date, Message**

**(F)** Find an entry

In that option user have two selections, to either Find by *username* or by *Date* , in current Guestbook version unfortunately the Username option is working, but we are planning to add *Date* option very soon! There’s example of using that option.

**

*When the username exists, it should look like this*

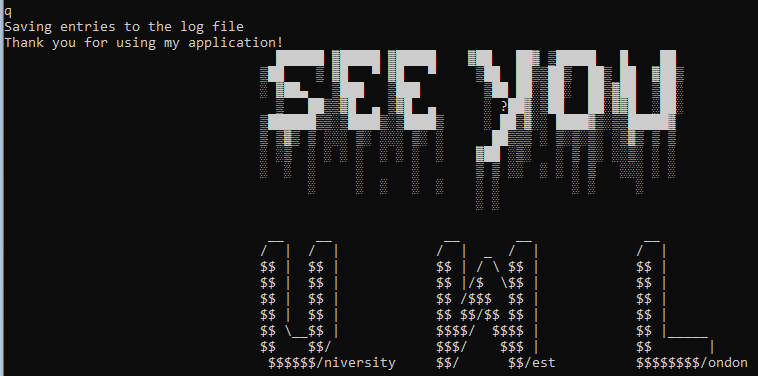
**

*When the username doesn’t exist, it should return an error*



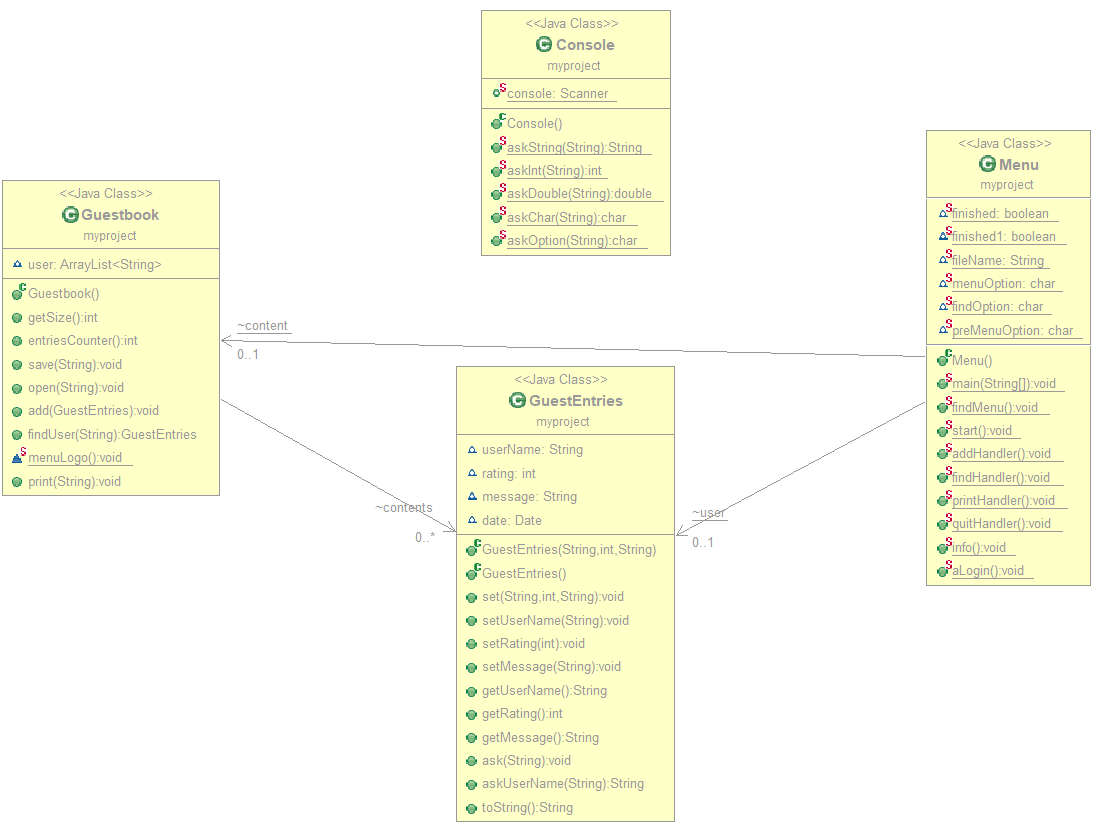
**(I)** Info

Its contains basic info screen to shows the current version of the program, and little bit showoff text and author, including mail.

**(Q)** Quit

Saves entries into logfile, shows following screen and quits the program.

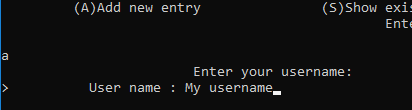
# Class Diagram

Diagram for each class to show Class name, Data properties and Method signatures

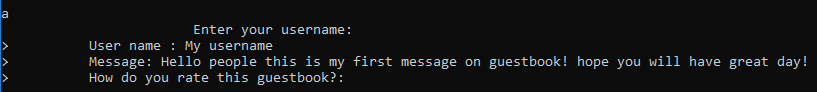
# Examples

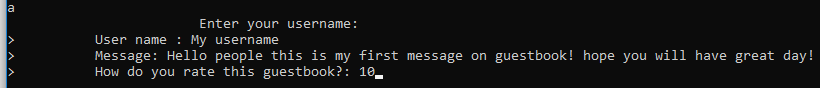
Examples

Adding new entry

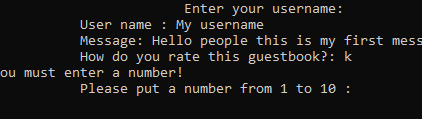


First it will prompt to enter your username

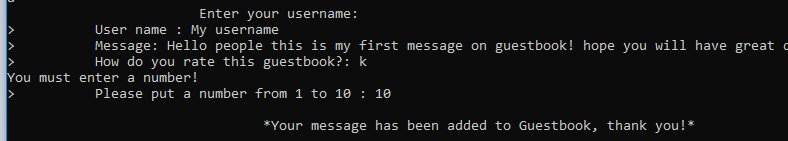
After that program will ask for message



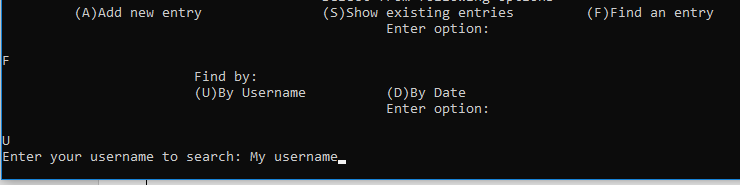
In the end it will ask user for rating



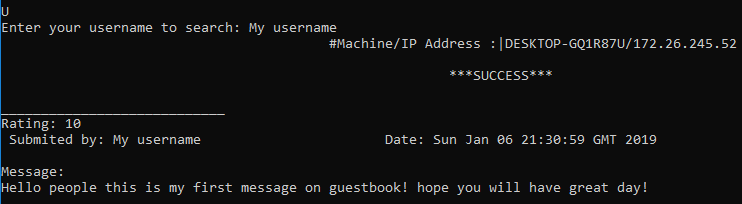
If user types in invalid number or character, it returns with following error



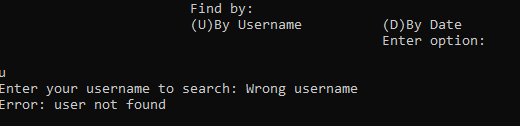
The end result should look like this

Viewing your entry

To view your entry you must search it by username, choosing option (U)By Username then typing username



Successful search should produce following screen



If the find by username fails, it should appear following

# Purpose of each class

That application contains 4 main classes



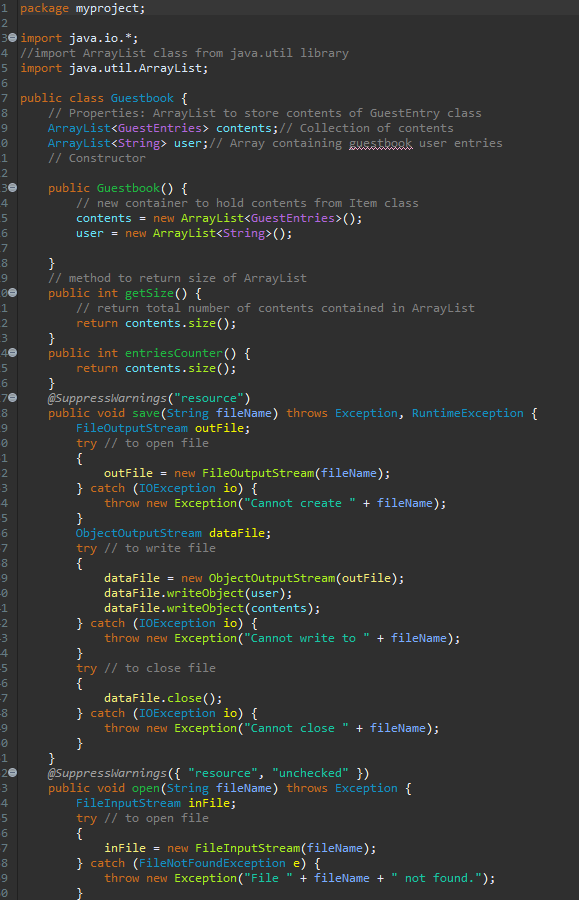
Console.java

Console class contains methods and parameters for input scanner



Guestbook.java

Guestbook class contains Array list to store contents, String array list to store guestbook user entries, log file save and opener as well as finding method

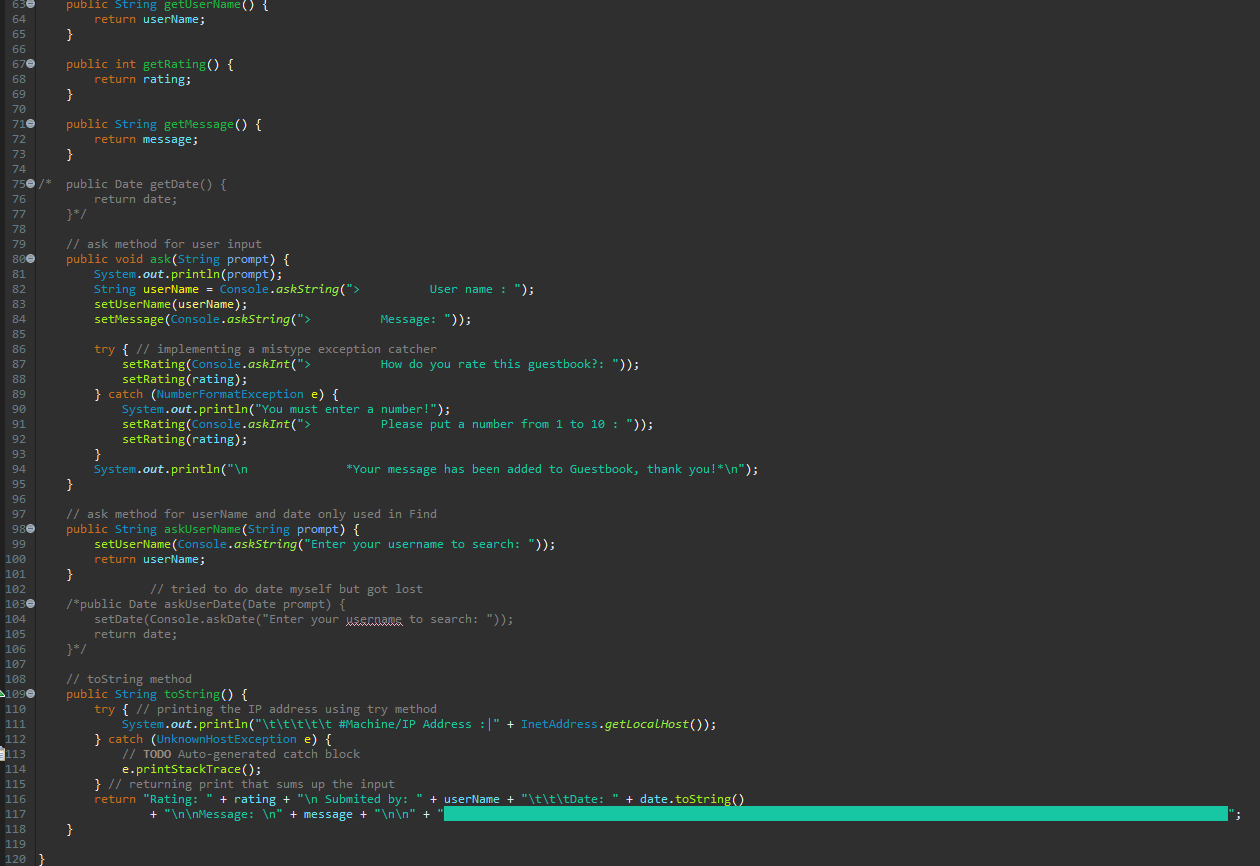




GuestEntries.java

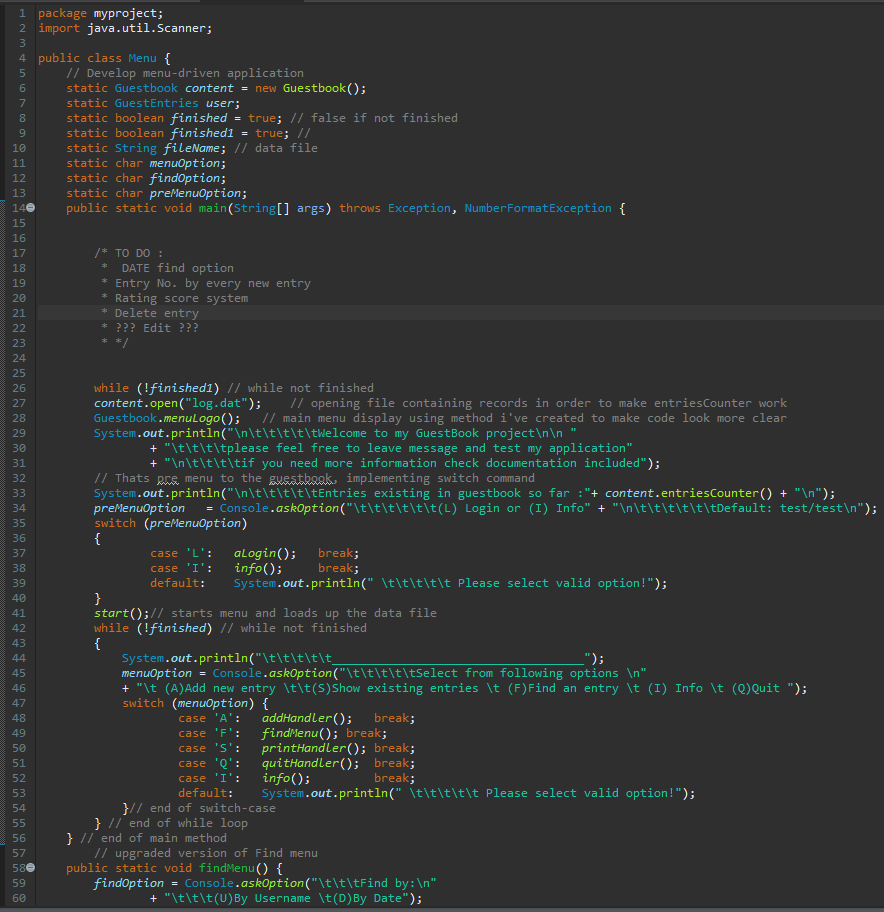
GuestEntries class contains methods and constructor its mainly a methods for contents inside the program. It defines the way they are and uses methods from other classes. There is

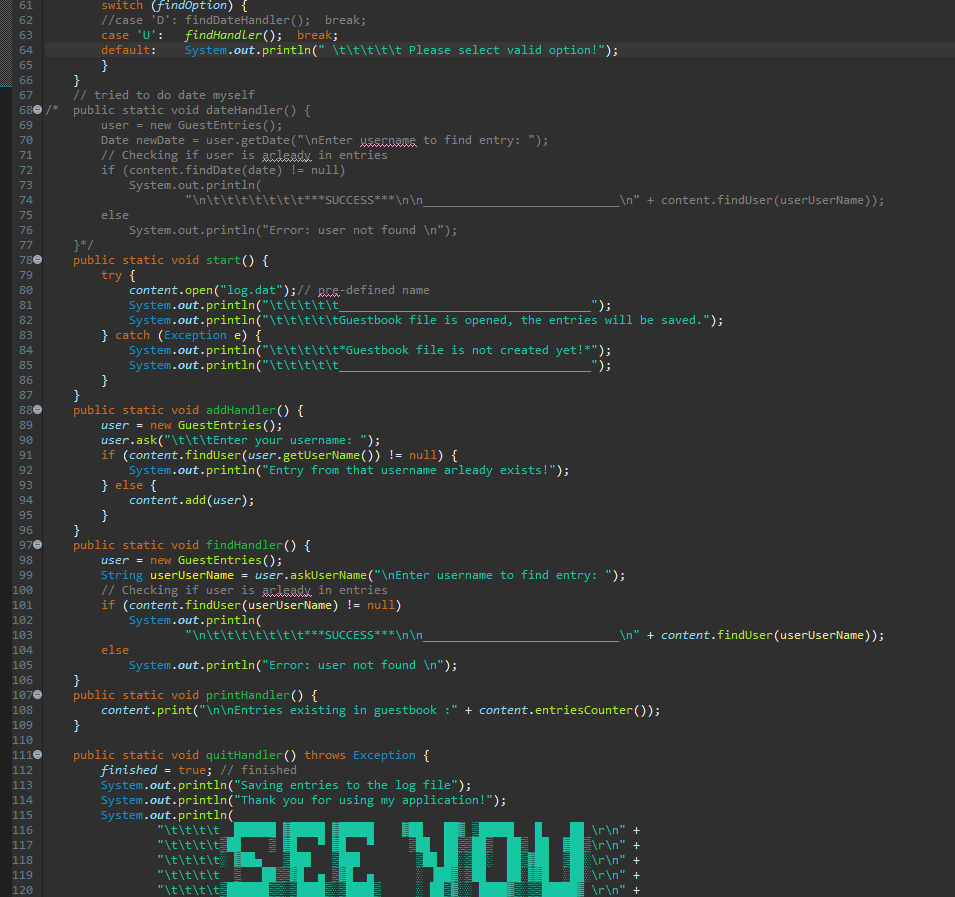




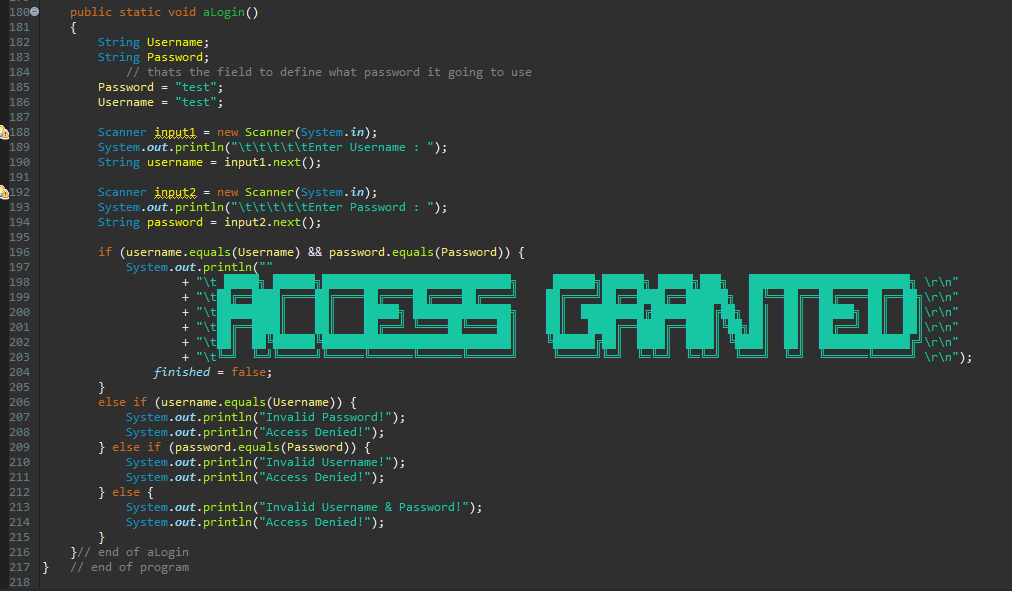
Menu.java

Menu class is bounding everything together and uses everything defined as it has main declared.



****



****

# Further Discussion and Future Plans for the Application

**The application is my first ever project created and its mainly my tool to learn more of JAVA programming, I would like to be able to do it completely myself without feedback however I am mostly limited by my knowledge. I would like to expand it more with the time and try new scripts and codes. For now I would like to add those features –**

* Find by date
* Have entry no. inside every entry, (entry counter)
* Rating score system, that takes every rate inside entry and AVG them
* Deleting an entry
* Two menus for two users, ADMIN/USER
* Admin view to find,remove entry and USER to ADD + others

# References

**References that was used for this project**

UWL Blackboards

Github

Stackoverflow

Java Tutorials

# Appendix



# 