

PROGRAMMING ASSIGNMENT 4

Issue Date : 22.04.2022 - Friday

Recitation Date : 22.04.2022 - Friday (14:00) (will be held on Zoom)

Due Date : 13.05.2022 - Friday (23:59:59)

Advisors :

HUCS Cinema Reservation System



1 Introduction

In this assignment, you are expected to gain practice on developing a Graphical User Interface (GUI) application using Java programming language. As opposed to the command-line programs where the interaction between the user and the computer often relies on a string of text, a GUI program offers a much richer type of interface where the user uses a mouse and keyboard to interact with GUI components such as windows, menus, buttons, checkboxes, text input boxes, scroll bars, and so on. Because most people today interact with their computers exclusively through GUI, developing a GUI based application has become a must for the new developers.

There are many frameworks to develop a GUI application (such as Swing, SWT, AWT, and the like). In this assignment, you are to employ JavaFX framework to complete this assignment. JavaFX is a software platform for creating and delivering desktop applications and Rich Internet Applications (RIAs) that can run across a wide variety of devices. JavaFX supports desktop computers and web browsers on Microsoft Windows, Linux, and macOS.

In this assignment, a very simplified version of the Cinema Reservation System is expected from you to develop through JavaFX framework. All details you need while designing the system are explained in the following section; also, a demo video will be provided for you.

2 System Requirements

- The system must read “backup.dat” to initialize the database (e.g. Collections). Note that it must start with an empty database (just containing an admin that is club member namely “admin” with a password of “password”) for initialization if there is no “backup.dat” file.

The screenshot shows a terminal window with the title "backup.dat". The content of the file is as follows:

```

user gorkemakyildiz 0dQ6aC5+BxAt4xDaD0tWew== false false
user zeynepinkaya CJOtKN03T4Vw2Rk4FDWj== false false
user aliseydikekeli KFq1YRgo9y13xeecAGNDw== false true
user iliyascicekli ORozvuy6AfjnZ1pzy7PU== true true
user cemilzalluhoglu KFq1YRgo9y13xeecAGNDw== false true
user merveozdes c3PyLlmgJg4YV/KdQnftw== false false
user aydinkaya Pn1HBt+gRmx4THw2QszCA== true false
user sevilsen KFq1YRgo9y13xeecAGNDw== false true
user admin X03M01anZdYdyteui1PmQ== true true
user nebiyilmaz c3PyLlmgJg4YV/KdQnftw== false false
user berranurozturk CJOtKN03T4Vw2Rk4FDWj== false false
film Avengers: Endgame avengers_endgame.mp4 182
hall Avengers: Endgame S1 70 10 8
seat Avengers: Endgame S1 0 0 null -1
seat Avengers: Endgame S1 0 1 null 0
seat Avengers: Endgame S1 0 2 null 0
seat Avengers: Endgame S1 0 3 null -1
seat Avengers: Endgame S1 0 4 null -1
seat Avengers: Endgame S1 0 5 null 0
seat Avengers: Endgame S1 0 6 null 0
seat Avengers: Endgame S1 0 7 null 0
seat Avengers: Endgame S1 1 0 null 0
seat Avengers: Endgame S1 1 1 null 0
seat Avengers: Endgame S1 1 2 null 0
seat Avengers: Endgame S1 1 3 null 0
seat Avengers: Endgame S1 1 4 null 0
seat Avengers: Endgame S1 1 5 null 0
seat Avengers: Endgame S1 1 6 null 0
seat Avengers: Endgame S1 1 7 null 0
seat Avengers: Endgame S1 2 0 null 0

```

Figure 1: backup.dat

- Note that you are not responsible for any erroneous line, but you should skip empty lines even if they contain whitespace characters such as a newline character, a space character, tab character, etc.
- A sample backup file is provided for you, the structure of it as follows:

* **For user record:**

- user[tab]username[tab]hashed_password_in_base64_format[tab]true_if_club_member_false_if_vice_versa[tab]true_if_admin_false_if_vice_versa

* **For film record:**

- film[tab]filmname[tab]trailer_path(relative)[tab]duration

* **For hall record:**

- hall[tab]filmname[tab]hallname[tab]price_per_seat[tab]row[tab]column

* **For seat record:**

- seat[tab]filmname[tab]hallname[tab]row_of_seat[tab]column_of_seat[tab]owner_name(null_if_not_owned)[tab]price_that_it_has_bought(can_be_any_number_if_seat_is_not_bought_yet)

- Note that the user, film, hall and seat records can be provided in any line of the backup, but it is guaranteed that all the seats of a hall are provided after that hall's record, all the halls of a film are provided after that film's record, all the seats of a user are provided after that user's record.

- The system must save the current state to “backup.dat” if the user clicks the exit button (the native exit button of operating system, cross button at the top right for Windows OS, at the top left for MacOS, Linux, etc.).
 - Note that your system must reinitialize itself by that backup at next runs.
- The window’s header must be named as same as property namely “title” at properties file, and the logo must be the icon at the “.../assets/icons/logo.png”. You can read “properties.dat” by native reader of Properties class of Java.
- The system must check credentials of user at the **login window**.



Figure 2: Login Window

- The system must give appropriate error sound (“.../assets/effects/error.mp3”) and message, if there is an issue (such as no such a credential), and also the system, must ban that client for “block-time” seconds if he/she tries to login with not existing credentials for "maximum-error-without-getting-blocked" times that provided at properties.dat and give appropriate message and sound for that ban. User can try to log in at that phase, but the system must not let him/her log in and notify that he/she is banned for given seconds.
- The system must check if passwords are matching at **signup window**.



Figure 3: Signup Window

- The system must give appropriate error sound (“.../assets/effects/error.mp3”) and message if they are not matching.
- The system must check if username already exists at signup window.
 - The system must give appropriate error sound (“.../assets/effects/error.mp3”) and message if they are not matching.
- The newly created users must be non-admin and non-club-member.

- The system must open **welcome window** after login.
 - If logged in user is a **normal user**, then system must show following:
 - * A text that welcomes user with showing his/her nickname and stating that club member (if he/she is)
 - * A dropdown list that contains available films. Note that one of the options must be selected as default.
 - * OK button for confirming selection.
 - * Log Out button that logs out user from system and directs to login window.
 - If logged in user is an **admin**, then system must show following in addition to normal user:
 - * Welcome text must also state that user is admin.
 - * Buttons for adding/removing film, editing users.
- The system must open **film window** after selecting and confirming selection.

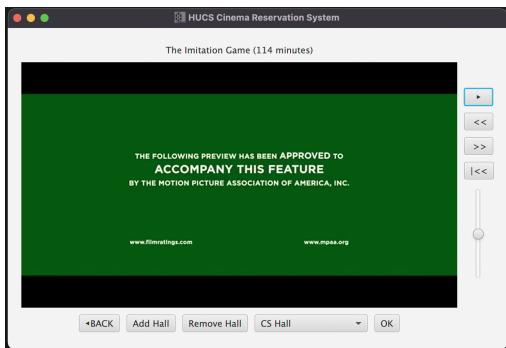


Figure 4: Film Window (Admin)

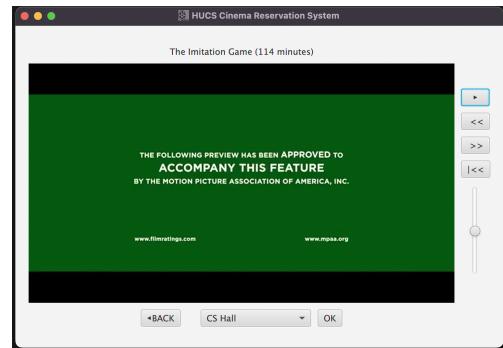


Figure 5: Film Window (Normal User)

- If current user is a **normal user**, the film window should show followings:
 - * Name of the film and duration of the film.
 - * Media buttons (play/pause, skip back for 5 seconds, skip forward for 5 seconds, rewind button, volume slider). Note that system should not skip back more than beginning of the trailer and should not skip forward more than ending of the trailer, in short, your system should not crash if user tries to skip forward for five seconds even if there is less than five seconds to end of the trailer.
 - * Back button that directs previous window.
 - * A dropdown list that contains available halls. Note that one of the options should be selected as default.
 - * OK button for confirming selection.

- If current user is an **admin**, the film window should show following in addition to normal user:
 - * Add Hall and Remove Hall buttons.
- The system must open **hall window** after selecting and confirming the hall, note that fragment must be stopped in case of changing screen (back or OK buttons).

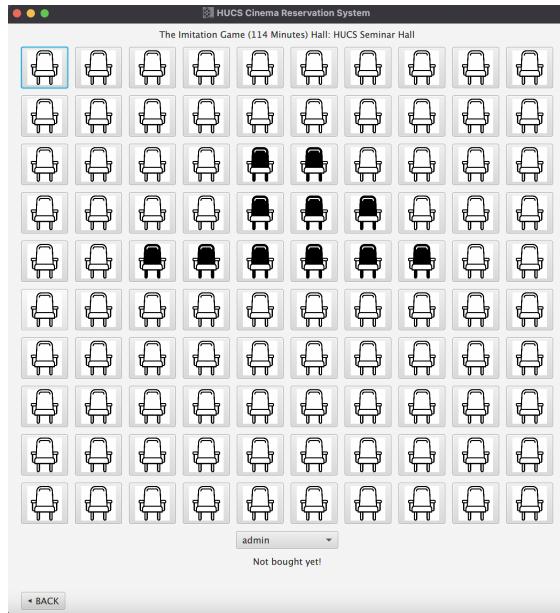


Figure 6: Hall Window (Admin)

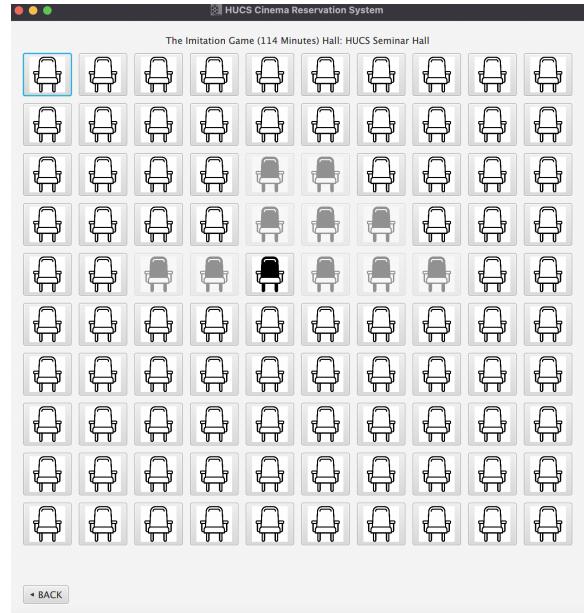


Figure 7: Hall Window (Normal User)

- If current user is a **normal user**, the hall window should show following:
 - * A text that tells the name of the movie and hall name.
 - * Available seats as selectable buttons with appropriate image (empty_seat.png)
 - * Bought seats
 - As selectable buttons with an appropriate image (reserved_seat.png) if those seats are reserved by current user.
 - As not selectable buttons with an appropriate image (reserved_seat.png) if vice versa.
 - * The seats must be in air view and the order that curtain is at bottom.
 - * User can buy empty seats or get refund for reserved seats simply by clicking on them, system show a message about the operation, containing seat number (in a-b format, where a is the row number and b is the column number – one indexed, indexing begins from one)
 - * Back button that directs to previous window.

- If current user is an **admin**, the hall window should show following in addition to normal user:
 - * All seats are as selectable buttons as admin can easily refund any reserved seat just by simply clicking on them.
 - * A dropdown list containing the users' name, admin can select the name of the user and then the seat so that he/she can reserve a seat for that user. Note that one of the options should be selected as default.
 - * A message about the seat that only appears if mouse is over that seat.
- The system must open **add film window** after clicking on it. This window is just for admins and contains following:

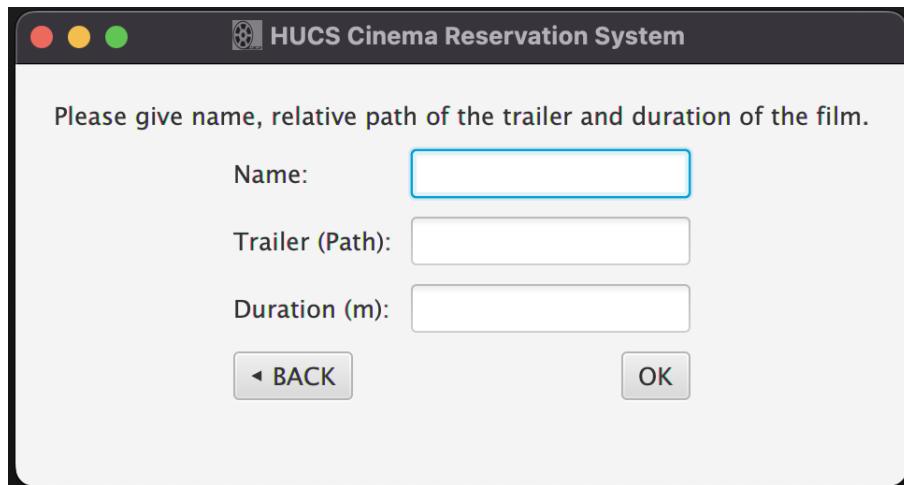


Figure 8: Add Film Window

- A message that tells what is going on (such as “Please give name, relative path of the trailer and duration of the film.”)
- Field for name of the film.
 - * System should check if name of the film is unique and not empty, if not then must give appropriate error message and error sound (“.../assets/effects/error.mp3”).
- Field for relative path of trailer (Relative to “.../assets/trailers/” path)
 - * System should check if there is such a file, if not then should give appropriate error message and error sound (“.../assets/effects/error.mp3”).
- Field for duration of film in minutes.
 - * System should check if given data is positive integer, if not then should give appropriate error message and error sound (“.../assets/effects/error.mp3”).
- Back button that directs to previous window.
- OK button for confirming input.

- The system must open **remove film window** after clicking on it. This window is just for admins and contains following:

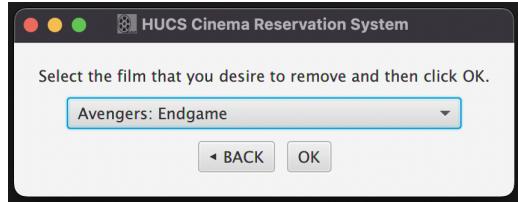


Figure 9: Remove Film Window

- A message that tells what is going on (such as “Select the film that you desire to remove and then click OK.”)
- A dropdown list that contains films. Note that one of the options should be selected as default.
- Back button that directs to previous window.
- OK button for confirming delete.
 - * System must refresh the list after deleting.
 - * System must delete corresponding halls and seats that belongs to that movie.

- The system must open **edit users window** after clicking on it. This window is just for admins and contains following:

Username	Club Member	Admin
gorkemakyildiz	false	false
zeynepinkaya	false	false
aliseydikeceli	false	true
ilyascicekli	true	true
cemilzalluhoglu	false	true
merveozdes	false	false
aydinkaya	true	false
sevilsen	false	true
nebiyilmaz	false	false
berranurozturk	false	false

Figure 10: Edit Users Window

- A table that has three columns namely username, club member, admin. Note that one of the options should be selected as default. The system should show the following message on table if there is no user at database: “No user available in the database!”

- Back button that directs to previous window.
- Promote/Demote Club Member button that promotes or demotes selected user (promotes if he/she is not, demotes if he/she is)
- Promote/Demote Admin button that promotes or demotes selected user (promotes if he/she is not, demotes if he/she is)
- **Note that the current user must not be in the table!**
- The system must open **add hall window** after clicking on it. This window is just for admins and contains following:

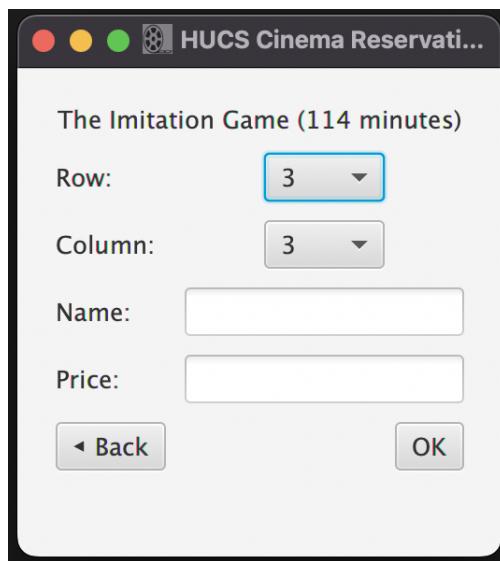


Figure 11: Add Hall Window

- A text that contains name and duration of the film.
- Two dropdown lists that is for selecting row and column count of the hall (from three to ten in number format, such as 3, not three) Note that the option three must be selected as default.
- Name of the hall.
 - * Name of the hall must be unique (not only for current film but also for all films) and cannot be empty, if not then should give appropriate error message and error sound (“.../assets/effects/error.mp3”).
- Price for single ticket.
 - * Price must be discounted for club users with discount rate as stated at properties file. Note that price should be discounted if and only if the user is a club member when buying a ticket.
 - * System must check if given data is positive integer, if not then should give appropriate error message and error sound (“.../assets/effects/error.mp3”).

- The system must open **remove hall window** after clicking on it. This window is just for admins and contains following:

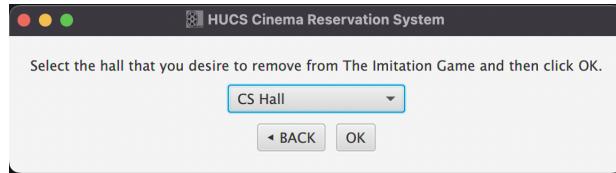


Figure 12: Remove Hall Window

- A message that tells what is going on (such as “Select the hall that you desire to remove from NAME_OF_THE_CURRENT_MOVIE and then click OK.”)
- A dropdown list that contains halls. Note that one of the options should be selected as default.
- Back button that directs to previous window.
- OK button for confirming delete.
 - * System should refresh the list after deleting.
- You are expected to implement an extra feature that is not stated at PDF, and you must state it at your report (benefits of it, how to use etc.) if you did any, it will be considered as bonus, note that the points that you will get as bonus will not only evaluated by completing it but also its benefit and your creativity will be part of grading too. Note that your extra feature must be unique, so that please do not share your ideas about your extra feature with your friends, etc., because the same extra features will be considered cheating and you will not get any bonus points.
- Moreover, you are free to implement any extra features or any more good-looking GUI than provided but remember that your points will be off if your extra feature ruins one of the main requirements of the system that stated at PDF.
- You must prepare a report, the report must contain following parts:
 - A cover
 - An index
 - The part that you define the problem
 - The part that you explain your solution approach
 - The part that you explain the problems you faced and the solutions you found to solve these problems
 - Benefits of this system
 - Benefits of GUI
 - Resources (if you used any)
 - Appendix (if you have any)

- Moreover, you can add image from your GUI if it is **necessary**, the text at your report cannot be more than three pages and less than one page when it copied to an empty Microsoft Word document with settings of font type Calibri, font size 11, so do not concern about images, cover and index pages as the limitation is only about the text that you write during your report. Your report must be in passive voice and it must be in justify alignment. Note that justify alignment is not needed to be obeyed for the headers or sections. Do not inject any code fragment to your report. Your reports must also be unique as your code, so do not share your code and report with anybody until objections are finalized.

3 Restrictions

- Your code must be clean, do not forget that main and start methods are just driver methods that means they are just for making your code fragments to run, not for using them as main container, create classes and methods in necessary situations but use them as required. Moreover, use the four pillars of Object-Oriented Programming (Abstraction, Encapsulation, Inheritance, Polymorphism) if there is such a need, remember that your code must satisfy Object-Oriented Programming Principles, also you can benefit from exceptions and even if create your own exception class if you need any.
- You are encouraged to use lambda expressions which are introduced with **Java 8**.
- You must append a ReadMe file ("**README.md**") that contains compilation details of your project.
- You must append a PDF file ("**Checklist.pdf**") in addition to your report that contains the requirements that you did not satisfy, please state your situation such as "This functionality does not work at all.", "There is a tiny issue such as ..." so that your assignment will be graded by taking them into consideration, please note that if you did not stated a requirement, that is assumed that that functionality is fully working on your system and that will be graded as zero even if a tiny issue occurs that stated as has to be checked (not for the issues that they are not stated at requirements, but remember that there may be some extra conditions that you should check, but they are not going to be graded as zero if you do not satisfy them). Moreover, if you did not provide such a file, it assumed that your code is working well without issue and that it will be graded as zero even if a tiny issue occurs that stated as must be checked.
- You must provide a **demo video** (in English with system sounds) that shows requirements. This demo must not exceed 8 minutes (even for one second, details of the demo can be reached at "**Checklist.docx**"). Remember that limit of submit system for each file is 8192 KB, so that you must upload your videos to your YouTube account as **unlisted** and you must keep them as unlisted until the current term (Spring 2022) ends. Place its link to your "**Checklist.pdf**". You must store all the data about your homework (code, report, video etc.) until the current term ends.
- **You must store users' password in MD5 hashed format** and encoded as Base64 (code is given at Appendix Section), remember that you must read from backup as hashed and Base64 encoded (not as plaintext as it is vulnerable to attacks).

- You can benefit from "extra.dat", which is created by you, for storing data related to your extra feature. Moreover, you can add effects and icons to your program. Note that you must also add to your submit the things that you added to assets folder.
- You can benefit from Internet sources for inspiration but do not use any code that does not belong to you.
- You can discuss high-level (design) problems with your friends but do not share any code or implementation with anybody.
- You must use Javadoc commenting style for this project, and you must give brief information about the challenging parts of your code, do not over comment as it is against clean code approach. Design your comments so that some wants to read your code can easily understand what is going on.
- Do not miss the submission deadline.
- Source code readability is a great of importance. Thus, write READABLE SOURCE CODE, comments, and clear MAIN function. This expectation will be graded as "clean code".
- Use UNDERSTANDABLE names to your variables, classes, and functions regardless of the length. The names of classes, attributes and methods should obey Java naming convention. This expectation will be graded as "coding standards".
- You can ask your questions through course's piazza group, and you are supposed to be aware of everything discussed in the piazza group. General discussion of the problem is allowed, but **DO NOT SHARE** answers, algorithms, source codes and reports.
- All assignments must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating.
- Submit system for this homework will be opened a few days before deadline, so please be patient.

4 Execution and Test

Your code must be compiled and executed under **Java 8** (preferably the one provided on Piazza) on Windows Operating System (preferably Windows 10 or newer) as you will not have any chance to compile and run your program on dev server since it does not contain any graphical user interface. **It is forbidden to use any external libraries** such as JSON Library as you have used at Assignment 2, and **it is also forbidden to use any scene builders, FXML stuff etc.** You are encouraged to build your own GUI just with Java Code as stated at JavaFX slides. It is also forbidden to use any extra CSS stuff, you can only use something like following:

```
lb1.setStyle("-fx-border-color: green; -fx-border-width: 2");
```

Note that the contents of assets file may changed but it is guaranteed that file names are provided as provided at the assets folder that shared with you.

5 Grading

Task	Point
Reading the backup and properties	12
Storing the backup	8
Graphical Design	10
Login/Signup Window	5
Welcome Window	5
Add/Remove Film Windows	5
Edit Users Window	5
Film Window	5
Add/Remove Hall Windows	5
Hall Window	15
Error Handling (with sound and message)	15
Report	20*
Extra Feature	10 (Bonus)*
Total	120 (Including Bonus)

* The score of your report and extra feature will be multiplied by your overall score excluding report and extra feature divided by the maximum score taken from these parts. Say that you got 72 from all parts excluding report and extra feature, and 15 from report, 5 from extra feature, your score for report is going to be $15 * (72/90)$ which is 12 and your score for extra feature is going to be $5 * (72/90)$, which is 4, and your overall score will be 88.

6 Submit Format

File hierarchy must be zipped before submitted (Not .rar, only not compressed the system supports .zip files) Note that you will not upload assets folder, it will be placed as the same directory of your Main class.

- b<studentid>.zip
 - Report.pdf
 - README.md
 - Checklist.pdf
 - <src>
 - Main.java
 - *.java (optional)
 - <assets> (optional)
 - <data> (optional)
 - extra.dat (optional)
 - <effects> (optional)
 - *.* (any extra effects - optional)
 - <icons> (optional)
 - *.* (any extra icons - optional)

7 Late Policy

You have three days for late submission. You will lose 10 points from maximum evaluation score for each day (your submitted study will be evaluated over 110, 100 and 90 for each late submission day). You must submit your solution in at the most three days later than submission date, otherwise it will not be evaluated.

8 Appendix

```
/**  
 * Returns Base64 encoded version of MD5 hashed version of the given password.  
 *  
 * @param password Password to be hashed.  
 * @return Base64 encoded version of MD5 hashed version of password.  
 */  
private static String hashPassword(String password) {  
    byte[] bytesOfPassword = password.getBytes(StandardCharsets.UTF_8);  
    byte[] md5Digest = new byte[0];  
    try {  
        md5Digest = MessageDigest.getInstance("MD5").digest(bytesOfPassword);  
    } catch (NoSuchAlgorithmException e) {  
        return null;  
    }  
    return Base64.getEncoder().encodeToString(md5Digest);  
}
```

Listing 1: Code 1: Method that hashes given password.