|  |  |
| --- | --- |
|  | INFO-6120  Project – Part 2  Due Date: December 9, 2023, 11:59pm |

**Goal/Objective**

Now that we have designed a database for Yoober, we need an easy way to view and manipulate the data in the database. To do this, you are being asked to develop a console application to communicate with your database.

The requirements for the console application are as follows:

* When starting the application, a welcome message is shown, including the names of both members of your group
* Immediately following the welcome message, a menu of options supported by the application should be shown. The supported options should be:
  1. View all account details
  2. Calculate the average rating for a specific driver
  3. Calculate the total money spent by a specific passenger
  4. Create a new account
  5. Submit a ride request
  6. Complete a ride
* Then, the application should accept user input to select a menu option. The user will enter the number of the menu option they wish to perform. For example, if they want to view all account details, they would enter the number **1**. Your application should perform basic validation on the user’s input. (i.e., if the user enters something other than a number between 1 and 6, they should be instructed to enter a valid choice).
* Based on the option entered by the user, the application should handle the request appropriately. The required behaviour for each option is as follows:
  1. The following data for each account should be displayed to the user:
     + First and last name
     + Full address (i.e. street, city, province, postal code)
     + Phone number
     + Email address
     + Whether account is used by a passenger, driver, or both
  2. The user should be instructed to enter the email address of the driver in which they are interested. Then, the average of all available ratings given by passengers for all trips the specified driver has provided should be calculated and displayed to the user.
  3. The user should be instructed to enter the email address of the passenger in which they are interested. Then, the total amount of money charged for all trips taken by the specified passenger should be calculated and displayed to the user.
  4. With the data provided by the user specified below, insert new records into the database as appropriate. The user should be prompted to enter values for:
     + - First and last name
       - Birthdate
       - Full address (i.e. street, city, province, postal code)
       - Phone number
       - Email address
     + Then, the user should be asked if the new account will be used by a passenger, driver, or both.
       - If passenger, prompt user to enter:
         * Credit card number
       - If driver, prompt user to enter
         * Driver’s license number
         * Driver’s license expiry date
       - If both, prompt the user to enter the information listed above for passengers and drivers
  5. With the data provided by the user specified below, insert new records as appropriate into the database. The user should be instructed to enter:
     + - The email address of the passenger making the request
     + Then, ask the user whether they want to choose their destination from the specified passenger’s list of favourite destinations
       - If yes:
         * Display a list containing the ID and full address details of all the passenger’s favourite destinations
         * Prompt the user to enter the ID corresponding to their choice of destination from their list of favourites
       - If no:
         * Prompt the user to enter the full address details of their destination
         * Ask the user if they want to make this destination a new favourite

If they do, prompt the user to provide a name for the location, and add the destination as a new favourite location for the passenger

* + - Then, prompt the user to enter:
      * Desired pick-up date and time
      * Total number of riders
    - For simplicity, we will assume the pick-up location is always the address tied to the passenger’s account
  1. Display a list containing the following details of all **uncompleted** rides:
     + - ID
       - First and last name of passenger who requested the trip
       - Street and city of both the pick-up and destination addresses
       - Desired pick-up date and time
     + Then, prompt the user to enter the ID corresponding to the ride they want to complete
     + Complete the ride in the database with the following information provided by the user:
       - Driver’s email address
         * Use this to help specify the driver for the trip
       - End date and time
       - Distance travelled
       - Cost
       - Driver’s rating
       - Passenger’s rating
* Every time the user selects a menu option and finishes the option’s workflow, they should be prompted again to select a new option. They may exit the application by entering the word “exit” instead of a menu option.
* Remember, all addresses should only be stored in the database once. So, whenever address details are provided by the user, a new address should only be created in the database if it doesn’t already exist.
* For options 2, 3, 5, and 6, if no passenger/driver can be found with the email address entered by the user, indicate this to the user and ask for a new email address until a valid one is provided

The exact visual design of how you display data to the user is up to you. Since this is a database course, the focus of this project is on your application interacting with a database, and not how beautiful your output is. However, your output should still be neat enough to understand, and contain all the information required.

**Technical Requirements**

Your application should be a Java console application using the Sqlite JDBC to communicate with the provided *yoober\_project.db* Sqlite database to retrieve, insert, and update the appropriate data. **You can also use the database you designed in project part 1.** The *yoober\_project.db* file is available on FOL. Do not modify the schema of the provided database. All requirements for this project can be met with the database provided, as is. Failing to follow these requirements will result in a grade of 0 for the project.

**Submission Details**

You will be working in groups of two to complete the project. You will submit a zipped file containing your Java project. The *yoober \_project.db* file should be contained within the *src* folder of your Java project, and the url defining the location of your database should be pointing to your project’s *src* folder as demonstrated in class. Also, please include the Sqlite JAR file directly in your project’s *lib* folder. This will allow your project to automatically link to it without me having to manually reference the JAR when I open it on my computer. **Be sure to follow these instructions to avoid mark deductions.**

When submitting, please name your file in the format ***<initials\_last\_name1>-<initials\_last\_name2>-project2.zip***. So if Mitchell Dooreleyers and Santa Claus were partners for the project, their submitted file should be called ***m\_dooreleyers-s\_claus-project2.zip***. Only one member of your group needs to submit the project.

The project is due on December 9, 2023, by 11:59pm.

**Grading**

The project will be marked using the following rubric:

|  |  |
| --- | --- |
| **Requirement** | **Mark** |
| Application displays welcome message on launch | 1 |
| Application displays menu options at least once | 1 |
| Menu option 1 behaviour meets requirements | 2 |
| Menu option 2 behaviour meets requirements | 2 |
| Menu option 3 behaviour meets requirements | 2 |
| Menu option 4 behaviour meets requirements | 4 |
| Menu option 5 behaviour meets requirements | 5 |
| Menu option 6 behaviour meets requirements | 4 |
| User can continue selecting menu options indefinitely | 1 |
| Basic input validation exists for menu options | 1 |
| All displayed output is reasonably humanly readable | 2 |
| User can exit application as described | 1 |
| **Total** | **26** |
| **Possible deductions** |  |
| Database file not included with project | -100% |
| Database schema modified from original | -100% |
| Database connection url not correctly pointing to database file in *src* folder | -30% |
| Sqlite JAR not in *lib* folder | -10% |

If you have any questions related to the project, please don’t hesitate to contact your professor as soon as possible.