



BRACE YOURSELF

HOMEWORK IS COMING

Software Design Assignment 1

The semester just started and we already need a vacation :(But we can't decide on a destination for that. So, let's design an application that would make this task as easy as possible :)

Now, the application needs to accommodate only **2 types** of users (lucky you):

- *Regular User/Vacay seeker*
- *Travelling Agency*

The **Travelling Agency** should be able to:

1. add vacation destination
2. add vacation packages for a specific destination
 - a. should contain information: name, price, period, extra details, number of people that can book the vacation
3. edit an existing vacation package
4. delete an existing vacation package
5. view all its listed vacation packages (with status: BOOKED, NOT_BOOKED, IN_PROGRESS)
6. delete vacation destination

The **Regular User** should be able to:

1. register on the platform using some credentials (username/email - *unique* & password)
2. login on the platform
3. view all **available** vacation packages
4. filter vacation packages by destination/price/period
5. book a vacation
6. view all its booked vacations

Note: Each vacation package has a limit of people that can book the trip. When someone books the trip, you should keep track of the number of people who booked a vacation, so that if the maximum number of people who can book that vacation is reached, another user will not be able to see that vacation anymore in the platform.

Note2: When displaying a vacation to the Travelling Agency, a status will be shown, based on the number of people who booked that vacation, such as:

7. **NOT_BOOKED:** nobody booked that vacation yet
8. **IN_PROGRESS:** somebody booked that vacation, but limit not reached
9. **BOOKED:** limit of people is reached

Technical constraints:

- Desktop application, written in **Java + Java Swing/Java FX**
- Connectivity to **relational database** + storage of data
 - each table in the database should contain at least 5 entries
- Implement Data Access Layer using **ORM** (eg. Hibernate)
- Implement the app using **Layered Architecture**
- Documentation
- Validation of the inputs on specific flows (valid dates, non-negative values, unique username etc.)

Deliverables:

Create a **github repository** with the followings: **documentation, source code, SQL script**. Provide the link of the repository.

Grading:

Grade	Requirement
5	Documentation + Travelling Agency CRUD operations (1, 2, 3, 4, 5 - without status)
6	Travelling Agency delete destination (6)
7	User register & login (1, 2)
8	User view & filter (3, 4)
9	User: book vacation, view booked vacation (5, 6) Travelling Agency: view vacation with status (5 - with status)
10	Clean code, clean architecture, validations

Example of a valid list of vacations:

Maldiva:

Maldiva Package 1:

Maldiva Package 2:

Alaska:

Alaska Package 1:

Alaska Package 2:

Documentation should contain:

- Database diagram
- Use case diagram
- Class diagram (show how the classes are placed inside each layer and the relationship between them - to denote Layered Architecture)