



Java HandsOn Assessment

Sports Equipment Renting System

- Title of the Project - Sports Equipment Renting System-Set4
- Complexity - Medium
- Target Band - Band x / Band 1 / Band 2
- Downloadable Starter Code link: [Sports Equipment Renting System-Starter Code](#)
- Skills: Java, JUnit Mockito, PostgreSQL.
- Time taken to complete: 3 Hours
- IDE: IntelliJ, Eclipse

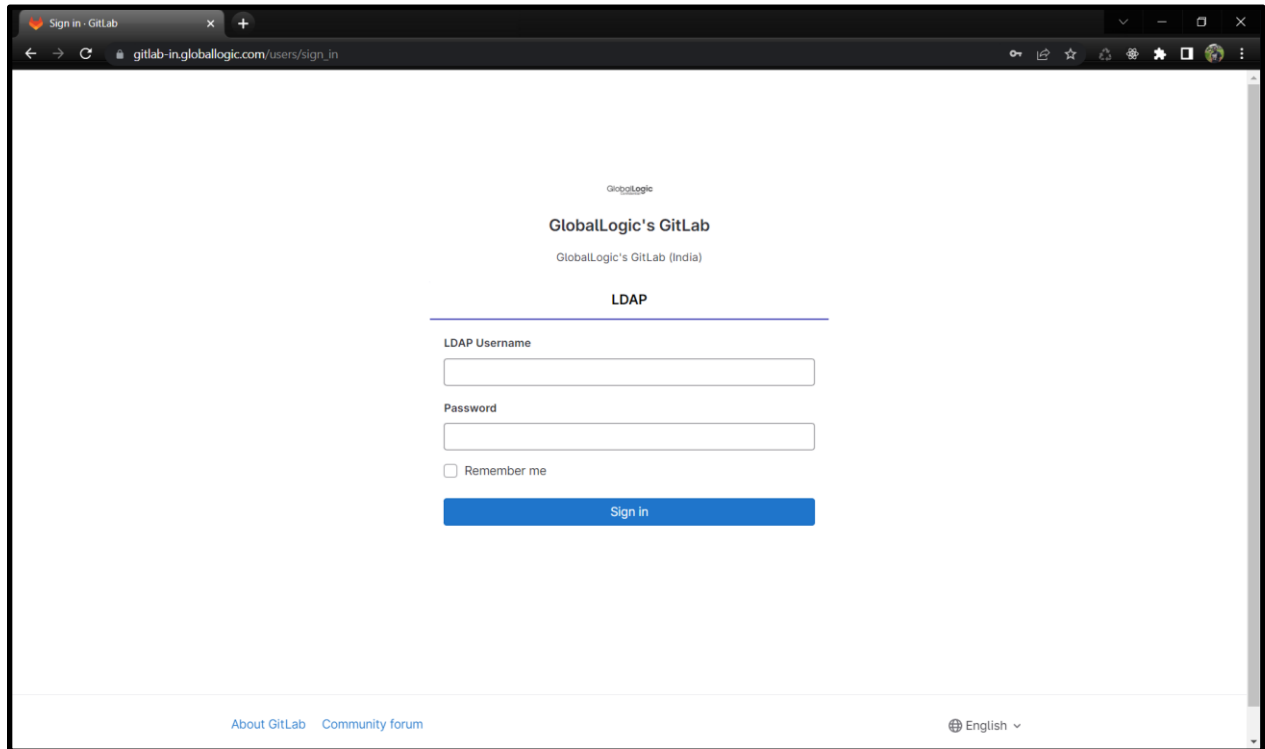
Technology	Topics
Java	OOPs, JDBC, Abstract Class and Interfaces, Exception Handling, Collections And Generics, Streams API, Lambda Expression, JUnit5 And Mockito, PostgreSQL.0

Table of Contents

Problem Statement	6
2. User Stories	7
3. DB Design	9
Equipment	9
4.Implementation	10
Class / Method Description	10
Output Screens	12
5.Testing	15
Instructions to Create and Execute the Project	15
Evaluation Metrics	16

GITLAB GENERAL INSTRUCTIONS

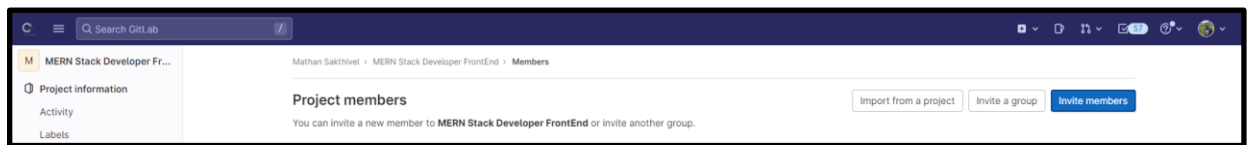
1. Connect to VPN to access GlobalLogic GITLAB
2. Open the URL in the browser - https://gitlab-in.globallogic.com/users/sign_in



3. Login using GlobalLogic credentials.
4. Find a project to fork: Navigate to the project that you want to fork on GitLab. You can find projects by browsing the public repositories or by searching for a specific project.
5. Fork the project: Once you have found the project you want to fork, click the "Fork" button in the top right corner of the project page. This will create a copy of the project in your GitLab account.
6. Clone the forked project: Once you have forked the project, clone the project to your local machine using Git. To do this, open a terminal window and navigate to the directory where you want to clone the project. Then, run the following command:
`git clone [URL of the forked project]`
7. Make changes to the project: Once you have cloned the project to your local machine, make any necessary changes to the project files using your preferred code editor.
8. Commit your changes: Once you have made changes to the project files, stage the changes and commit them to your local repository using the following commands:
`git add.`
`git commit -m "Your commit message here"`

9. Push your changes to GitLab: Once you have committed your changes to your local repository, push the changes to GitLab using the following command:
`git push origin [name of your branch]`
10. The forked project will have the Private Access so explicitly add the users in the project members.

Click on Invite Members



Provide access to the Educators:

Username: Ex: minakshi.sharma@globallogic.com and vipin.n@globallogic.com

Role: Owner

Invite members

×

You're inviting members to the **Flight** project.

Username or email address

Vipin N X

Select members or type email addresses

Select a role

Owner

[Read more](#) about role permissions

Access expiration date (optional)

YYYY-MM-DD

Cancel

Invite

SYSTEM REQUIREMENTS

1. Eclipse/STS/IntelliJ IDEA
2. PostgreSQL
3. Java 17 or above

Sport Equipment Renting System

Problem Statement

The Sport Equipment Renting System is a Java-based project designed to manage and book various types of sports equipment.

Features

The system should have the following features:

Equipment Management:

- The system should facilitate the inclusion of equipment details, including type, brand, model, rental cost, and dates of availability. This process will necessitate the creation of a new equipment record within the system. Additionally, the system should offer features for the modification and removal of equipment records.
- The system should permit the adjustment of dates when the equipment is available for rent. This will require updating the availability dates for a specific piece of equipment.

2. User Stories

User Story	Description	Acceptance Criteria
US01	As an admin, I want to add equipment to the system.	<ul style="list-style-type: none">• The system should provide a feature for adding equipment.• The system should validate the input to ensure it is not empty.• The system should confirm the successful addition of new equipment.
US02	As an admin, I want to update availability dates for equipment so that users can book equipment accurately.	<ul style="list-style-type: none">• The system should provide a feature for updating availability dates.• The system should validate the input to ensure it is not empty.• The system should confirm successful updates of availability dates.
US03	As an admin, I want to be able to get the list of all equipments along with their available date	<ul style="list-style-type: none">• The system should provide a list of all the equipments along with their available dates.
US04	As an admin, I want to retrieve equipment details using an equipment ID so that I can view specific equipment information. Acceptance Criteria.	<ul style="list-style-type: none">• The system should provide a function <code>getEquipmentById(int equipmentId)</code> to retrieve equipment details.• The system should validate the input to ensure it is a valid equipment ID.• The system should return the equipment details if the equipment ID exists.• The system should return an appropriate error message if the equipment ID does not exist.

US05	As an admin, I want to have the ability to delete equipment records from the system using the equipment's unique ID so that I can manage the inventory effectively.	<ul style="list-style-type: none">• Given that I am an admin and I am on the equipment deletion interface, when I input a valid equipment ID, then the system should validate the ID and delete the corresponding equipment record.• Given that I am an admin and I am on the equipment deletion interface, when I input an invalid or non-existent equipment ID, then the system should display an error message.

3. DB Design

Equipment

Column Name	Description	Data Type	Not Null
equipmentId	This is a unique identifier for each equipment	INT	NOT NULL
type	This field represents the type of equipment(e.g "Tennis Racket", "FootBall")	VARCHAR(100)	NOT NULL
brand	Brand of the equipment	VARCHAR(100)	NOT NULL
model	Model of the equipment	VARCHAR(100)	NOT NULL
rentAmount	Rental price for the equipment per day	DECIMAL(10,2)	NOT NULL
availabledate	Identifier for the availability date	DATE	NOT NULL
Primary Key - equipmentId			

○

4.Implementation

Class / Method Description

This is a standalone Java application where a menu-driven interface needs to be developed to execute the following functionalities.

Note:- The above UserStories are converted into equivalent methods and the descriptions of the service methods provided below need to be implemented.

Entity Class	Service Methods	Description
Equipment(equipmentId,type,brand ,model,rentAmount,dateAvailability)	EquipmentService	
	addEquipment(Equipment equipment)	This method takes an Equipment object as a parameter. The Equipment object should contain all the necessary details about the equipment, such as its name, type, model,rent_amount , etc and add it to the database table
	updateAvailability(int equipment_id equipment, Date oldDate, Date newDate)	The `updateAvailability(int equipment_id, Date oldDate, Date newDate)method is designed to update the availability of a specific piece of equipment, identified by `equipment_id`. It changes the availability from the `oldDate` to the `newDate`. If the

		<p>`equipment_id` does not correspond to any existing equipment in the system, the method should ideally throw an `EquipmentNotFoundException`, ensuring that availability can only be updated for valid, existing equipment..</p>
	deleteAvailability(int equipment_id)	<p>The `deleteAvailability(int equipment_id)` method deletes the equipment based on `equipment_id`.</p> <p>If the `equipment_id` is not found in the system, an `EquipmentNotFoundException` should be thrown.</p>
	getEquipmentById(int equipment_Id) Use Stream API filter method to implement the above functionality	<p>The getEquipmentById(int equipmentId) method is a function that retrieves and returns the details of a specific piece of equipment in the system, identified by its unique equipmentId. The equipmentId is passed as an integer parameter to the function. If the equipmentId exists in the system, the function returns an Equipment object containing the details of the equipment. If the equipmentId does not exist, the function may return null or throw an exception, depending on the implementation.</p>
	public List<Equipment> getAllEquipments();	<p>The `getAllEquipments()` method is a public method that returns a list of `Equipment` objects.</p>

		This method does not take any parameters. When called, it is expected to retrieve all `Equipment` records from a database table.
--	--	--

Output Screens

1) Adding New Equipment Details in the database table

```
1. Add Equipment
2. Update Equipment Availability
3. Delete Equipment
4. Get EquipmentBy Id
5. List All Equipments
6. Exit
Enter your choice: 1
Enter equipment type: Badminton
Enter equipment brand: SportsWear
Enter equipment model: B010
Enter equipment rent amount: 200
Enter equipment available date: 2024-02-02
Equipment Added Successfully
1. Add Equipment
2. Update Equipment Availability
3. Delete Equipment
4. Get EquipmentBy Id
5. List All Equipments
6. Exit
Enter your choice:
```

2) Update Equipment Availability

```

SportsEquipmentRentApplicationSystem [Java Application] C:\sts-4.18.1.RELEASE\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (Apr 21, 2024, 5:02:06 PM) [pid: 3444]
Enter equipment model: B010
Enter equipment rent amount: 200
Enter equipment available date: 2024-02-02
Equipment Added Successfully
1. Add Equipment
2. Update Equipment Availability
3. Delete Equipment
4. Get EquipmentBy Id
5. List All Equipments
6. Exit
Enter your choice: 2
Enter equipment id: 101
Enter old available date (yyyy-mm-dd): 2024-02-02
Enter new available date (yyyy-mm-dd): 2024-02-07
Equipment availability updated successfully
1. Add Equipment
2. Update Equipment Availability
3. Delete Equipment
4. Get EquipmentBy Id
5. List All Equipments
6. Exit
Enter your choice:

```

3) Delete Equipment (By Id)

```

SportsEquipmentRentApplicationSystem [Java Application] C:\sts-4.18.1.RELEASE\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (Apr 21, 2024, 5:02:06 PM) [pid: 3444]
6. Exit
Enter your choice: 2
Enter equipment id: 101
Enter old available date (yyyy-mm-dd): 2024-02-02
Enter new available date (yyyy-mm-dd): 2024-02-07
Equipment availability updated successfully
1. Add Equipment
2. Update Equipment Availability
3. Delete Equipment
4. Get EquipmentBy Id
5. List All Equipments
6. Exit
Enter your choice: 3
Enter equipment id: 101
Equipment deleted successfully
1. Add Equipment
2. Update Equipment Availability
3. Delete Equipment
4. Get EquipmentBy Id
5. List All Equipments
6. Exit
Enter your choice:

```

4) Get Equipment By Id

```
1. Add Equipment
2. Update Equipment Availability
3. Delete Equipment
4. Get EquipmentBy Id
5. List All Equipments
6. Exit
Enter your choice: 4
Enter equipment id: 100
Equipment details:
Equipment(equipmentId=100, type=amc, brand=mmm, model=xy, rentAmount=900, availability=2024-02-03)
1. Add Equipment
2. Update Equipment Availability
3. Delete Equipment
4. Get EquipmentBy Id
5. List All Equipments
6. Exit
Enter your choice:
```

5) List All Equipments

```
1. Add Equipment
2. Update Equipment Availability
3. Delete Equipment
4. Get EquipmentBy Id
5. List All Equipments
6. Exit
Enter your choice: 5
List of all equipments:
Equipment(equipmentId=100, type=amc, brand=mmm, model=xy, rentAmount=900, availability=2024-02-03)
Equipment(equipmentId=101, type=Ball, brand=Super, model=S019, rentAmount=300, availability=2024-03-02)
1. Add Equipment
2. Update Equipment Availability
3. Delete Equipment
4. Get EquipmentBy Id
5. List All Equipments
6. Exit
Enter your choice:
```

6. Exit

```
1. Add Equipment
2. Update Equipment Availability
3. Delete Equipment
4. Get EquipmentBy Id
5. List All Equipments
6. Exit
Enter your choice: 6
Exiting...
```

5. Testing

- Create a new Java class named `EquipmentServiceTest` in your test directory.
 - Create a test method as `testGetEquipmentById_ValidId()`
This test will pass a valid `equipmentId` and expect a non-null `Equipment` object.

Instructions to Create and Execute the Project

- Please download the code structure from the provided GitLab link. The project structure can be found under the '`com.gl.app`' package and its subpackages.
- Please design the necessary database schema to store equipment details.
- Create a Menu driven Java Project which can add equipment details.
 - Use PostgreSQL as the database management system.
 - In addition to implementing the functionalities mentioned above.
 - Read Input from the keyboard using `Scanner` class.
 - You should include code for establishing a database connection, executing queries, and handling exceptions.
 - Use Java 8 Lambda Expressions.
 - Field Validations(Acceptance Criteria)
 - `equipment_id` should have a unique value in the equipment table.
 - Fields should not be null.

Evaluation Metrics

Sl. No.	Assessing Parameter	Marks
1	User Stories 5* 7 marks	35 marks
2	Usage of Lamdba	5 marks
3	Testing Module	5
4	Exception Handling and following best practices	5
	Total Marks	50 Marks