A logo of a university

AI-generated content may be incorrect.

**School of Computing and Engineering Sciences**

**Master of Science in Information Technology**

**MIT8102:** Advanced Distributed Systems

**Assignment:** Learning about RPC/RMI

**GROUP MEMBERS:**

|  |  |
| --- | --- |
| TOKOYE SAMUEL SHADIVA | 222072 |
| CATHERINE ANYANGO OWINO | 223933 |

**1. Introduction**

This document outlines the development of a distributed system called **Fruit Service Engine** that allows users to add fruits, update fruit prices, delete fruits, calculate the cost of purchasing a given quantity, generate receipt and view all available fruits with their prices.

**2**. **Technologies Used**

1. Java Servlets: act as intermediaries between the client and the server
2. GSON: used for data persistence.
3. IntelliJ IDEA: used as the development environment.
4. Tomcat Server: used to deploy the application web interface
5. HTML, CSS, JavaScript: used to style the user interface and allow for dynamic interaction with the backend.

**3. Project Overview**

The project was developed using Java RMI server as the backend, the Servlets acting as intermediaries between the frontend and the server, HTML, CSS and JavaScript being used to develop the user interface which allows the user to submit actions and finally, the fruit data is written in a file called fruit\_data.json.

**4. Architecture and RMI Workflow**

1. The user is presented with a web interface.
2. The user selects an action from the drop-down menu he or she is presented with.
3. After performing the action and clicking on the submit button, the request is sent to a servlet.
4. The servlet constructs a task and sends this task to the RMI server.
5. The RMI server executes the task on receiving it, updates the fruit\_data.json file and sends back a response.
6. This result sent by the server is rendered on the user interface for the user to view.

**5. Tasks Implemented**

* AddFruitPrice: Adds a new fruit and its price to file.
* UpdateFruitPrice: Updates the price of an existing fruit in the file.
* DeleteFruitPrice: Deletes a fruit from file along with its price.
* CalFruitCost: Calculates the cost of a given quantity for a fruit.
* CalculateCost: Returns a receipt with total, amount and change.
* ListFruits: Displays a list of all the fruits with their prices written on file.
* ListFruitNames: Populates dropdowns dynamically with all fruit names.

**6. Features**

* **User form** which updates dynamically based on selected action
* **AJAX** dynamic dropdown population for existing fruits
* **JSON persistence** ensures fruit data is written to fruit\_data.json file.
* **Responsive UI** designed with HTML, CSS and JavaScript
* **Receipt generation** with total and change.
* **Separation of concerns** using RMI task classes where each task has its action.

**7. Challenges and Solutions**

1. **RMI registry binding error (Port 1099 in use):** Ensured rmiregistry runs first, before running FruitComputeEngine and finally the Tomcat server.
2. **Tomcat artifact deployment error:** Fixed by updating IntelliJ artifacts to reflect the right files, deleting stale untitled.war mappings and updating Tomcat Configuration.
3. **Empty dropdown lists:** Rewrote dropdown rendering logic in my JavaScript functions, updated the functions to correctly fetch fruit data.

**8. Setup and Execution Steps**

1. Import the project into your IDE
2. Open the terminal and navigate to the target/classes directory.
3. Start the ‘rmiregistry’.
4. Run the FruitComputeEngine file to start the RMI server.
5. Run the Tomcat Server.
6. Access the web UI on the browser at: http://localhost:8080/FruitServiceEngine\_war\_exploded/
7. Perform various actions and verify result rendering and updates.

**References**

Apache Software Foundation. (n.d.). *Apache Tomcat*. <https://tomcat.apache.org/>

Google. (n.d.). *Gson* [Computer software]. GitHub. <https://github.com/google/gson>

Jakarta EE Project. (n.d.). *Jakarta Servlet Specification.* <https://jakarta.ee/specifications/servlet/>

JetBrains. (n.d.). *Getting started*. IntelliJ IDEA Help. <https://www.jetbrains.com/help/idea/getting-started.html>

Oracle. (2014). *Java RMI* (Java SE 8 Documentation). <https://docs.oracle.com/javase/8/docs/technotes/guides/rmi/index.html>