Cape Peninsula University of Technology

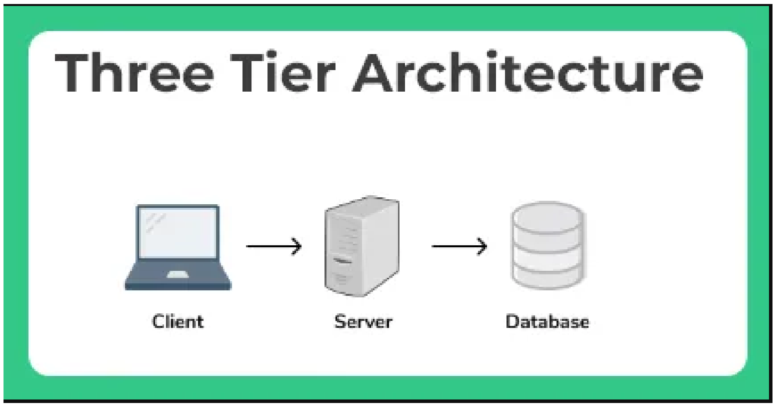
Applications Development Practice II (ADP262S)

Final Project **Due Date: 20th October 2024**

Lecturers: Radford Burger, Dr. Wilhelm Rothman, Dr. Richard Maliwatu

Question: Build a Simple Online Voting System

Description: Every year a panel of motoring journalists get together to select from a number of currently-produced vehicles, the one which represents excellence in its category. After 2 days of testing each journalist votes for the vehicle that impressed them the most. You are tasked with creating a basic online voting system using Java sockets for server-client communication. The system should allow a client to vote for predefined vehicle models (stored on a DB table) and display the current vote count. The client should be able to connect to the server, cast their votes, and see the updated vote counts for each option. The server should implement all database operations. Use Java Swing for the client and server interfaces.


Three Tier Architecture

Requirements:

1. **Server Side:**
   - Implement a server that listens on a specific port for an incoming client connection.
   - Maintain a database table to store the vehicles, and the vote counts for each vehicle.
   - There must be DB operations to add new vehicles to the table, and to update the table with the votes cast by the client.
   - There must be a DB operation to retrieve the complete list of vehicles and the number of votes cast for each.
   - A simple GUI is required here with a textarea in which the details of all the client-server communication is logged.

2. **Client Side:**
   - Create a Swing-based UI that displays the available vehicles and their current vote counts.
   - Implement client-side socket connection to the server.
   - Allow clients to select a vehicle and cast their vote.
   - Display the updated vote counts after voting.

Evaluation Criteria:

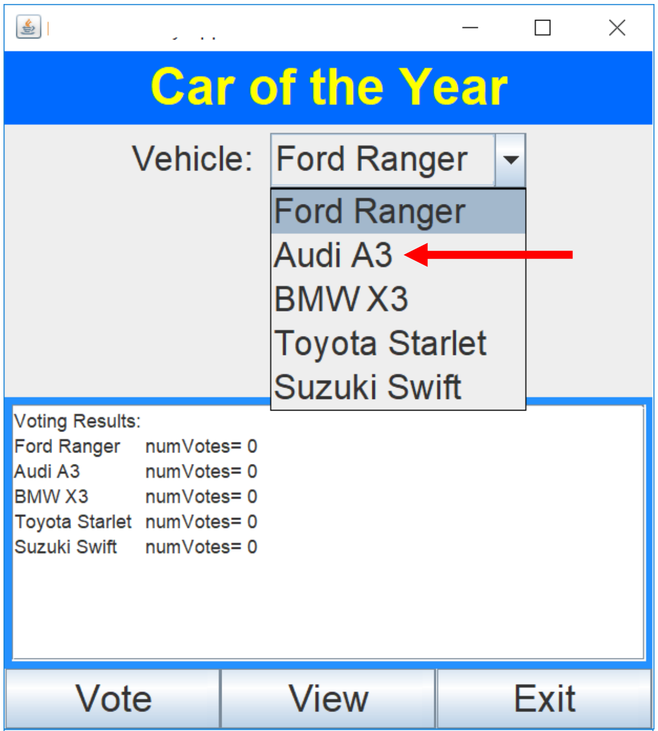
- Functionality of adding vehicles, voting, updating vote counts, and display of voting results
- Derby database setup
- Implementation of server-side logic
- Implementation of client-side UI and logic
- Proper client-server communication
- Implementation of database operations.
- Code organization, packaging, readability, and comments
- Group members' ability to answer questions based on their solution.

Rules:

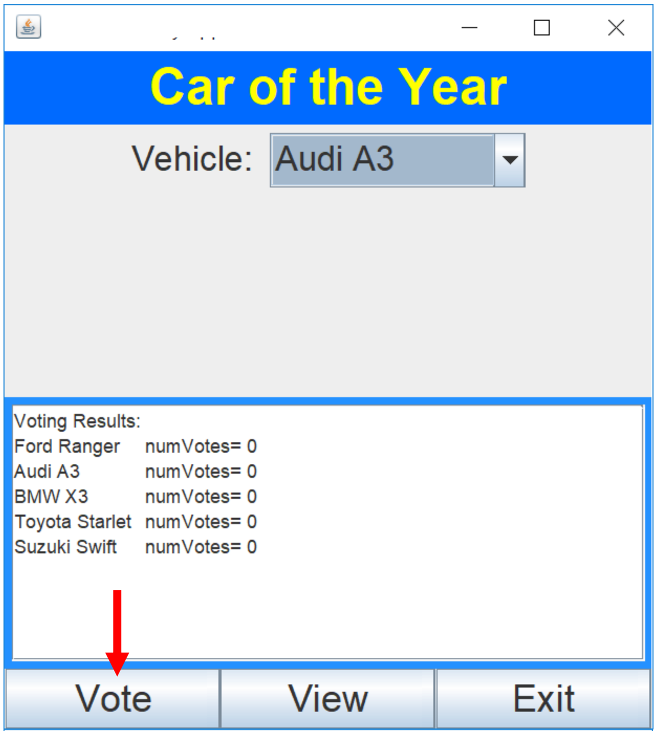
- You must complete this group project with **2 or 3 students per group**.
- This is a coding project and as such the members of the group must demonstrate that they have contributed equally to the coding of the application and that the work was distributed fairly. *Marks may be awarded differently for each member based on their contributions.*
- **The due date for this project is Sunday 20th October 2024 at midnight**. Upload your zipped solution (Netbeans Maven projects source code + database etc.) using the link provided on Blackboard. All group members should upload the same zipped solution.
- **Any plagiarism (copying) will be dealt with swiftly and severely.**

NO LATE SUBMISSIONS WILL BE ACCEPTED!! Dates for demonstrations will be from **21–25 October 2024**.

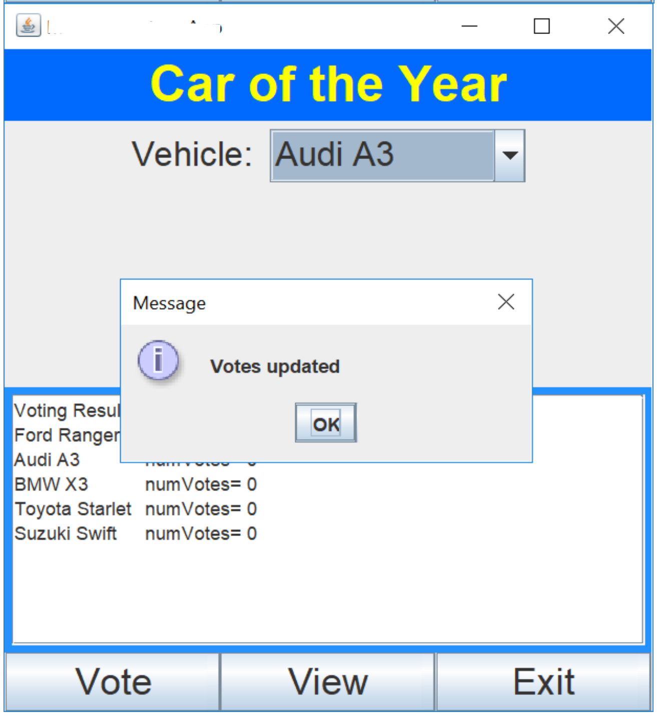
Sample Solution: This is just to give you an idea of what is expected. You will produce your own GUI design and include additional SWING components to add vehicles to the DB, etc.



```
Voting Results:
Ford Ranger    numVotes= 0
Audi A3        numVotes= 0
BMW X3         numVotes= 0
Toyota Starlet numVotes= 0
Suzuki Swift   numVotes= 0
```

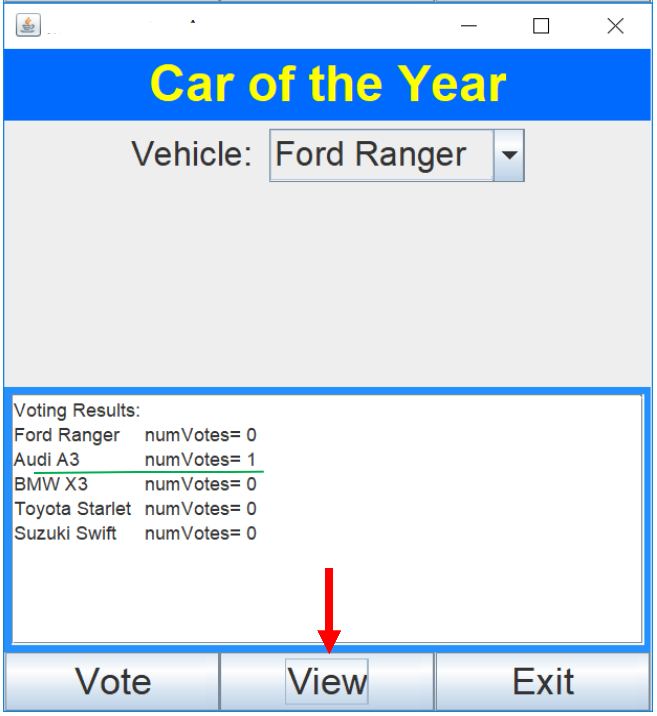


```
Voting Results:
Ford Ranger    numVotes= 0
Audi A3        numVotes= 0
BMW X3         numVotes= 0
Toyota Starlet numVotes= 0
Suzuki Swift   numVotes= 0
```



Votes updated

```
Voting Resul
Ford Ranger
Audi A3
BMW X3         numVotes= 0
Toyota Starlet numVotes= 0
Suzuki Swift   numVotes= 0
```



```
Voting Results:
Ford Ranger    numVotes= 0
Audi A3        numVotes= 1
BMW X3         numVotes= 0
Toyota Starlet numVotes= 0
Suzuki Swift   numVotes= 0
```