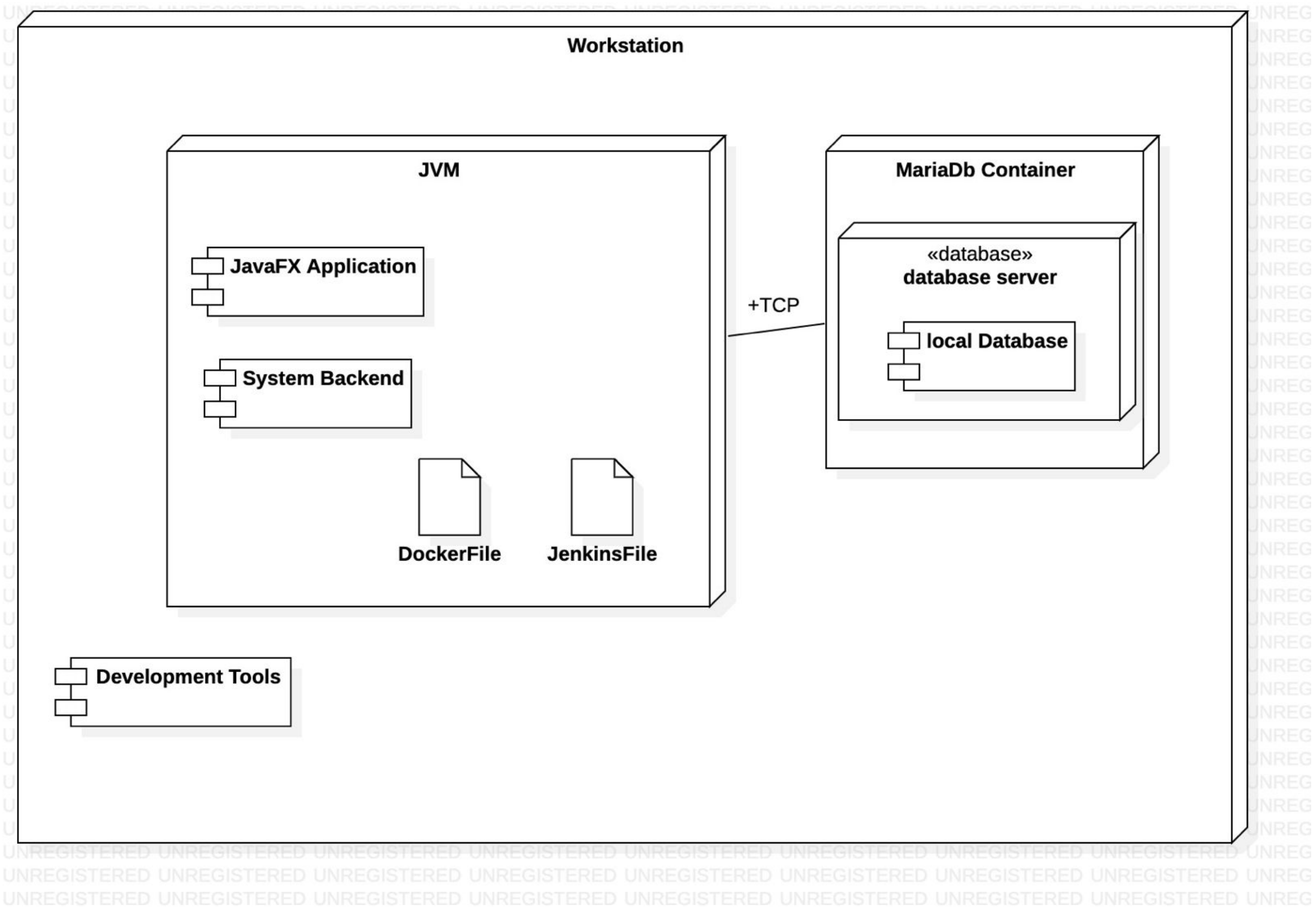
Group 5:
Mahnoor Fatima
Sergei Vilka
Trung Vu

Development Environment Diagram



Explanation:

Workstation Node - The Workstation represents the computer where the application is run. It hosts all the necessary components for development and execution.

JVM Node - The Java Virtual Machine (JVM) is responsible for running the application. It is the running environment.

JavaFX Application component - represents the front-end of your system, built with JavaFX. It provides the Graphical User Interface (GUI) for users to interact with application.

System Backend component - handles the back-end logic, business processes, and communication with external services, including the database.

Dockefile artifact - defines how the application is packaged and deployed inside a Docker container. Specifies the environment setup, including the JVM, dependencies, and application files.

Jenkinsfile artifact - defines the CI/CD pipeline for automated building, testing, and deployment.

MariaDb Container Node - provides the database environment that hosts database servers.

Database server Node - represents the actual database server instance that manages data storage and retrieval.

Local Database component - database instance within the server that stores project-related data.

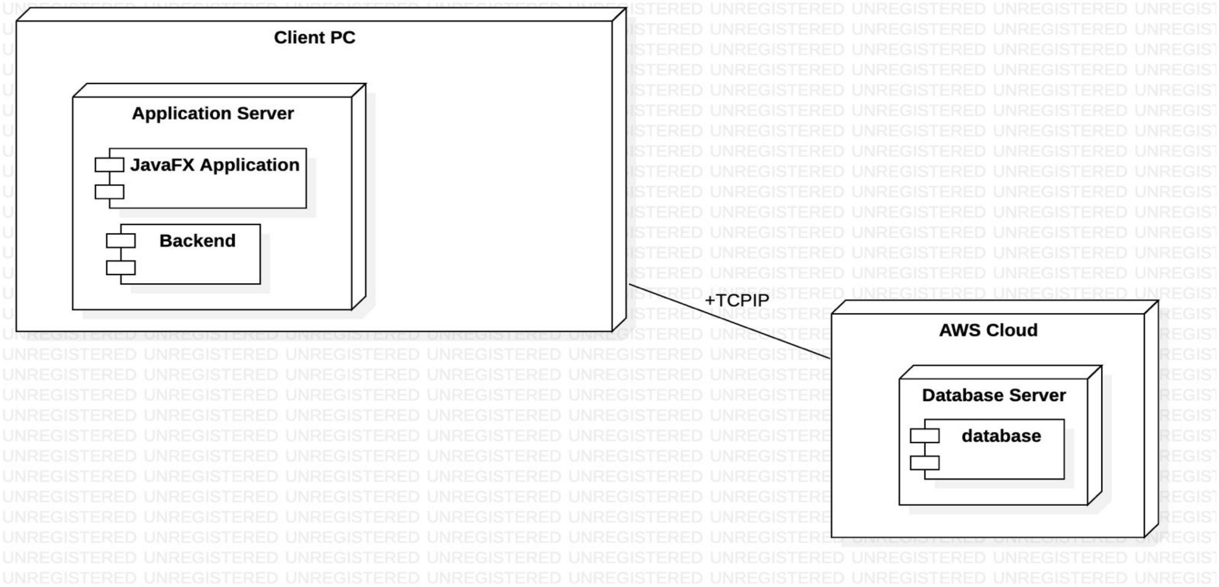
TCP connection - transmission Control Protocol (TCP) connection links the JVM to the Database Server, enabling communication between the backend system and the database.

Development Tools Component

It is the execution or development environment. It includes:

1. **IDE** – IntelliJ IDEA, Eclipse, or VS Code (for Java development)
2. **Build Tool** – Maven (for dependency management and build automation)
3. **Version Control** – Git (GitHub, GitLab, or Bitbucket for collaboration)
4. **Database Management** – MariaDB client tools (MySQL Workbench, DBeaver, etc.)
5. **Testing Frameworks** – JUnit 5 (unit testing), Mockito (mocking dependencies), Testcontainers (integration testing with MariaDB in Docker)
6. **Containerization** – Docker (for containerized environments)
7. **CI/CD** – Jenkins, GitHub Actions, GitLab CI/CD (for automated builds and tests)
8. **Logging & Monitoring** – Logback

Production Environment Diagram



Explanation:

Client PC Node - represents the computer where the application is run. It contains all the necessary components for development and execution.

Application Server Node - hosted locally on the client's machine and runs the application.

JavaFX Application component - represents the front-end of your system, built with JavaFX. It provides the Graphical User Interface (GUI) for users to interact with application.

System Backend component - handles the back-end logic, business processes, and communication with external services, including the database.

AWS Cloud Node - represents the Amazon Web Services environment that hosts the database server.

Database server Node - represents the actual database server instance that manages data storage and retrieval.

Database component - database instance within the server that stores project-related data.

TCP/IP connection - Transmission Control Protocol/Internet Protocol connection enables secure communication between the Application Server and the Database Server in the AWS Cloud, ensuring data exchange between the backend system and the database.