Fleet Management System

Architecture and Database

Documentation of a project for the purpose of the course BIE-SI1.

Authors: Vladimir Ananyev
Berker Katipoglu
Oleksii Tarbeiev
Atsamaz Akopyan
Muntaser Abu Zaid
Sanan Pashayev

Contents
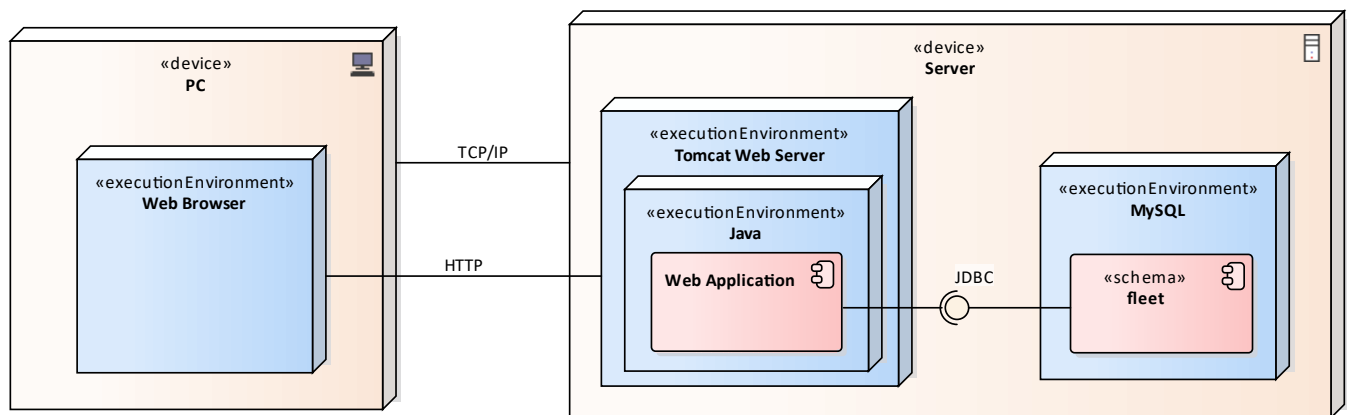
1. System Architecture

Figure 1 - System Architecture

The Company Fleet System consists of a central database located on central server. This server will be located in the company and will also run a Java web application. Employees will access to the web application using the internet browser on their own PCs.

2. Architecture

This chapter describes the architecture of the web application of the Fleet Management System.

The web application will be implemented as Java using following technologies:
• Java 1.8
• Spring Boot Framework
• Vaadin Framework

The architecture is divided into three independent layers:
• Presentation layer - layer responsible for presentation of application data.
• Business layer- layer responsible for all business logic of the application.
• Data layer - layer responsible for data persistence.

The layers are isolated using interfaces:
• IBusiness - interface between the presentation and business layers. It consists of one main IBusiness to handle any request from presentation layer.
• IDao - interface between the business and data layer. It consists of several IDao interfaces defining the persistence operations for individual entities in the system.
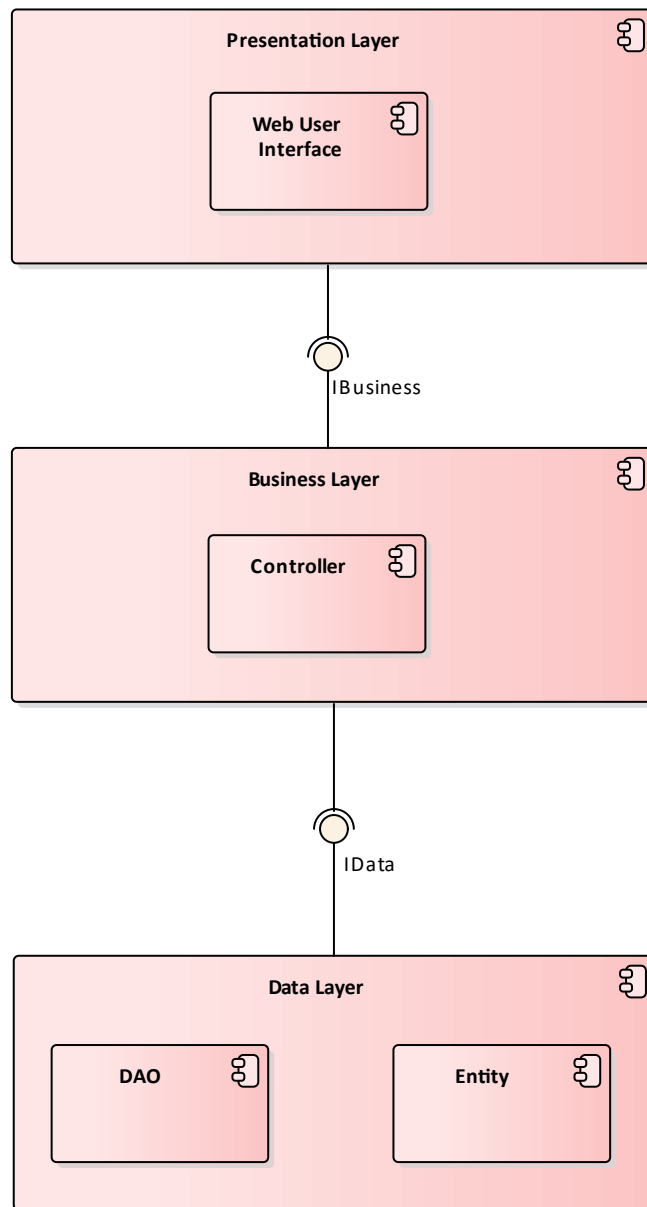
Figure 2 - Architecture

2.1 Presentation Layer

The presentation layer is responsible for sharing the information with a user or other systems. It consists of a web user interface and connected to business layer through business interface.

2.1.1 Web User Interface

Web User Interface is implemented in Java using Swing Boot and Vaadin frameworks.

2.2 Business Layer

The business layer contains implementation of the business logic and It consists of controllers implementing the system behavior. It is connected to data later through data interface.

2.2.1 Controller

Controller is implemented in Java using Spring Boot framework.

2.3 Data Layer

The data layer is responsible for data persistence. It consists of DAO classes implementing the persistence operations and entities representing the persistent data.

The implementation of the data layer is based on the Spring Boot Framework and its native support for database.

3. Database Model

This chapter describes the structure of the database schema where all the information about the employees, departments, trips, reservations and cars are stored.

3.1 MySQL

This package describes the structure of the MySQL database used by the Fleet Management system.
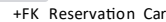
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Department
«column»
*PK id: INTEGER
* Name: VARCHAR(15)
*FK Employee_ID: INTEGER

«FK»
+ FK_Department_Manager(Employee_ID)
«PK»
+ PK_Department(id)
«unique»
+ Name(Name)

Role
«column»
*PK id: INTEGER
* role_name: VARCHAR(15)

«PK»
+ PK_Role(id)
«unique»
+ Name(role_name)

+FK_Department_Manager 0..1   1 +PK_Department

+PK_Role 1

(Employee_ID = id)
«FK»

«FK»
(Department_ID = id)

«FK»
(Role_ID = id)

+PK_Employee 1   1..* +FK_Employee_working for

+FK_RoleEmployee_Role *

Employee
«column»
* Name: VARCHAR(20)
* Surname: VARCHAR(20)
* Email: VARCHAR(20)
*FK Department_ID: INTEGER
*PK id: INTEGER
* Password: VARCHAR(20)

«PK»
+ PK_Employee(id)
«FK»
+ FK_Employee_working for(Department_ID)
«unique»
+ Email(Email)

RoleEmployee
«column»
*pfK Employee_ID: INTEGER
*pfK Role_ID: INTEGER

«FK»
+ FK_RoleEmployee_Employee(Employee_ID)
+ FK_RoleEmployee_Role(Role_ID)
«PK»
+ PK_RoleEmployee(Employee_ID, Role_ID)

Maintenance
«column»
* In_Date: DATE
Out_Date: DATE
Description: TEXT
*PK id: INTEGER
*FK Car_ID: INTEGER

«PK»
+ PK_Maintenance(id)
«FK»
+ FK_Maintenance_goes(Car_ID)

+PK_Employee
(Employee_ID = id)
1 «FK» 1..*

+FK_RoleEmployee_Employee

+FK_Maintenance_goes *

+PK_Employee 1   +PK_Employee 1

«FK»
(Employee_ID = id)

«FK»
(Employee_ID = id)

(Car_ID = id)
«FK»

+FK_Trip_approves *   +FK_Trip_plans *

+PK_Car 1

Trip
«column»
* Customer: VARCHAR(20)
* Start_Date: DATE
* End_Date: DATE
* State: VARCHAR(15)
*PK id: INTEGER
Description: TEXT
*FK Employee_ID: INTEGER

«PK»
+ PK_Trip(id)
«FK»
+ FK_Trip_approves(Employee_ID)
+ FK_Trip_plans(Employee_ID)

Reservation
«column»
Start_Date: DATE
End_Date: DATE
Start_Mileage: INTEGER
End_Mileage: INTEGER
* State: VARCHAR(15)
*PK id: INTEGER
FK Car_ID: INTEGER
*FK Trip_ID: INTEGER

«FK»
+ FK_Reservation_Car(Car_ID)
+ FK_Reservation_Trip(Trip_ID)
«PK»
+ PK_Reservation(id)

Car
«column»
* Model: VARCHAR(20)
* Plate: VARCHAR(15)
* Mileage: INTEGER
* Production_Year: INTEGER
Next_Maintanence_Date: DATE
* State: VARCHAR(15)
*PK id: INTEGER

«PK»
+ PK_Car(id)
«unique»
+ Plate(Plate)

+PK_Trip
(Trip_ID = id)
1 «FK» 0..*

(Car_ID = id)
0..* «FK» 1

+PK_Car

+FK_Reservation_Trip   +FK_Reservation_Car

Figure 3 - DDL

3.1.1 «table» Car

Car represents the cars in the company.
Each car may have zero or many maintenance that it was sent.
Also a car may be involved in zero or many reservations.

Column name	Data type	Not null	Description
Model	VARCHAR(20)	True	Model of the car stored as string.
Plate	VARCHAR(15)	True	Plate number of the car.
Mileage	INTEGER	True	Current mileage of the car stored as integer.
Production_Year	INTEGER	True	Production year of the car stored as integer.
Next_Maintanence_Date	DATE	False	Next planned maintenance date for the car stored as date.
State	VARCHAR(15)	True	State of the car(eg. available, damaged, in maintenance etc.) stored as string.
id	INTEGER	True	ID of the Entity.

3.1.2 «table» Department

Department represents different departments in the company.
Each department is managed by exactly one employee.
Each department may have one or more employees.

Column name	Data type	Not null	Description
id	INTEGER	True	ID of the Entity.
Name	VARCHAR(15)	True	Name of the department.
Employee_ID	INTEGER	True	FK from Employee entity.

3.1.3 «table» Employee

Employee represents a person working in the company.
According to responsibilities each employee has one or many roles in the company.
Each employee has a department that he is working for and also some employees manages some departments.
Employee may have only one department that he is working for.
It is possible that employee does not manage any departments.
Also each employee may have zero or many business trips.

Column name	Data type	Not null	Description
Name	VARCHAR(20)	True	Name of employee stored as string.
Surname	VARCHAR(20)	True	Surname of employee stored as string.
Email	VARCHAR(20)	True	Email of employee stored as string.
Department_ID	INTEGER	True	FK from Department entity.
id	INTEGER	True	ID of the Entity.
Password	VARCHAR(20)	True	Password of the user to access the system.

3.1.4 «table» Maintenance

Maintenance represents the maintenance history of each car in the company.
Maintenance must have exactly one car.

Column name	Data type	Not null	Description
In_Date	DATE	True	The date in which car was sent to maintenance stored as date.
Out_Date	DATE	False	The date in which car was taken from maintenance stored as date.
Description	TEXT	False	General description about why car was sent to maintenance and what was done during maintenance stored as string.
id	INTEGER	True	ID of the Entity.
Car_ID	INTEGER	True	FK from Car entity.

3.1.5 «table» Reservation

Reservation represents a car reservation made during the planning of a business trip.
It contains information about the trip and the car which was reserved for the given dates.
Each reservation must have trip object.
Each reservation may have a car, which represents a successful reservation.
Each reservation may lead to a loan, unless it was canceled.

Column name	Data type	Not null	Description
Start_Date	DATE	False	Start date of realized reservation stored as date.
End_Date	DATE	False	End date of realized reservation stored as date.
Start_Mileage	INTEGER	False	Mileage of the car when employee is taking the car stored as integer.
End_Mileage	INTEGER	False	Mileage of the car when employee is returning the car stored as integer.
State	VARCHAR(15)	True	State of the reservation(eg. accepted, rejected, canceled, realized) stored as string.
id	INTEGER	True	ID of the Entity.
Car_ID	INTEGER	False	FK from Car entity.
Trip_ID	INTEGER	True	FK from Trip entity.

3.1.6 «table» Role

Role represents the role of an employee in the company.
Each role may have zero or many employees.

Column name	Data type	Not null	Description
id	INTEGER	True	Name of the role, stating the role of the employee.
role_name	VARCHAR(15)	True	

3.1.7 «table» Trip

Trip represents a business trip that was planned by an employee.
Each trip needs to be approved by exactly one DM, who is managing the department who made the trip request.
Each trip also must have an employee who planned it.

Column name	Data type	Not null	Description
Customer	VARCHAR(20)	True	Name of the customer to be visited during business trip stored as string.
Start_Date	DATE	True	Trip start date stored as date.
End_Date	DATE	True	Trip end date stored as date.
State	VARCHAR(15)	True	State of the trip (eg. accepted, rejected, cancelled) stored

Column name	Data type	Not null	Description
			as string.
id	INTEGER	True	ID of the Entity.
Description	TEXT	False	
Employee_ID	INTEGER	True	FK from Employee entity.