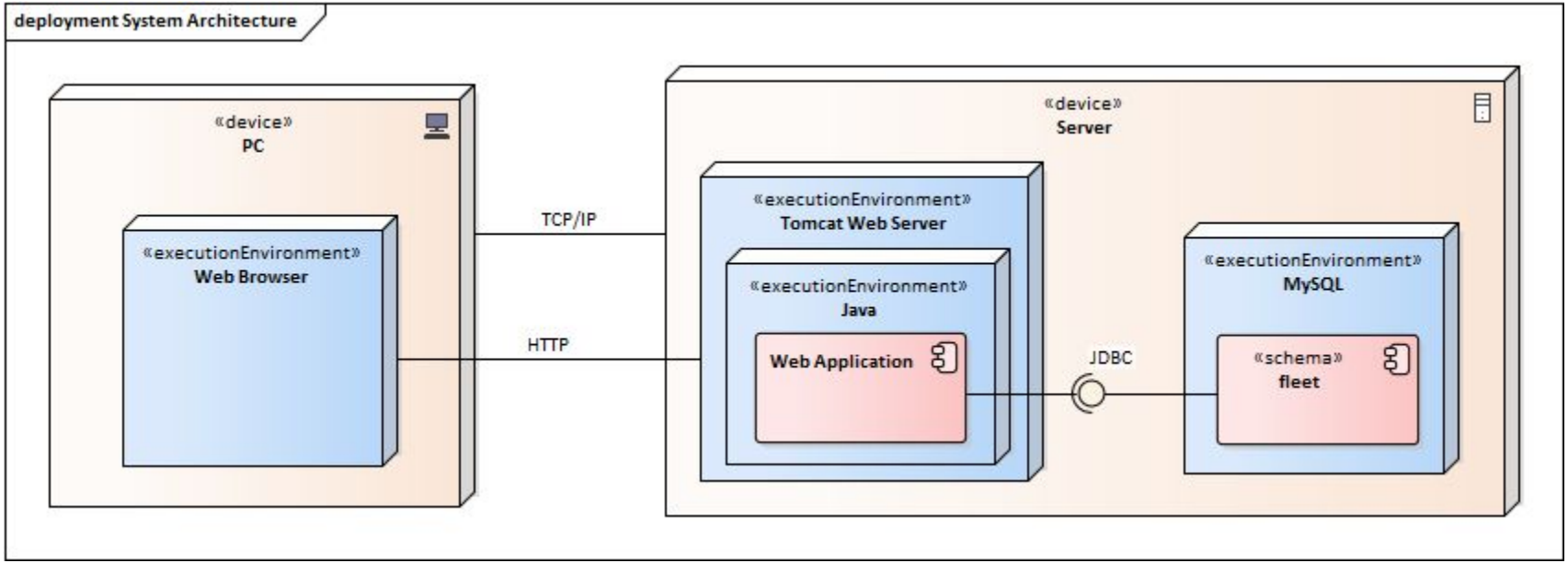
Company Fleet System

ITERATION 2

Agenda

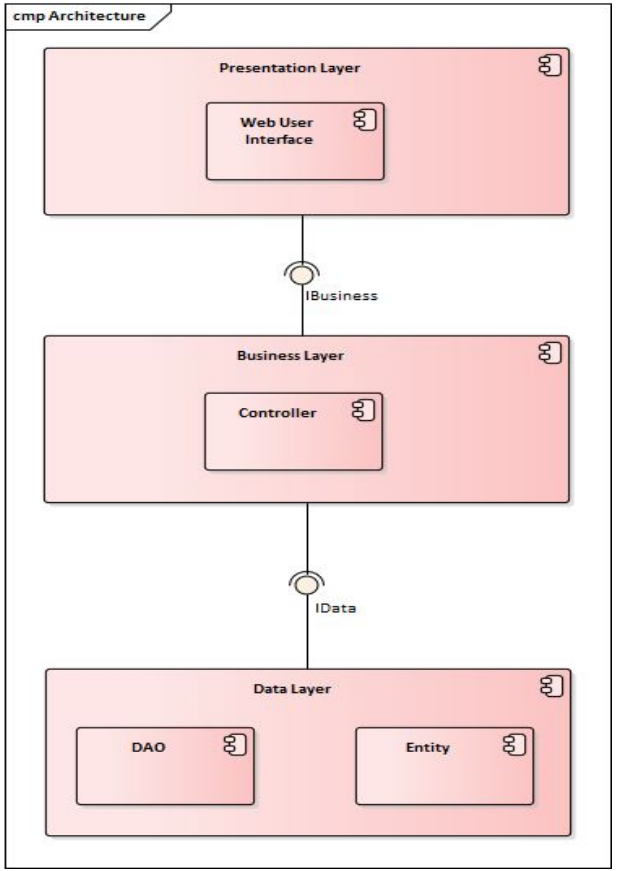
- System Architecture
- Logical Architecture
- Database Model
- Prototype
  - Presentation Layer
  - Controller Interface & Implementation
  - DAO Interface & Entity

System Architecture

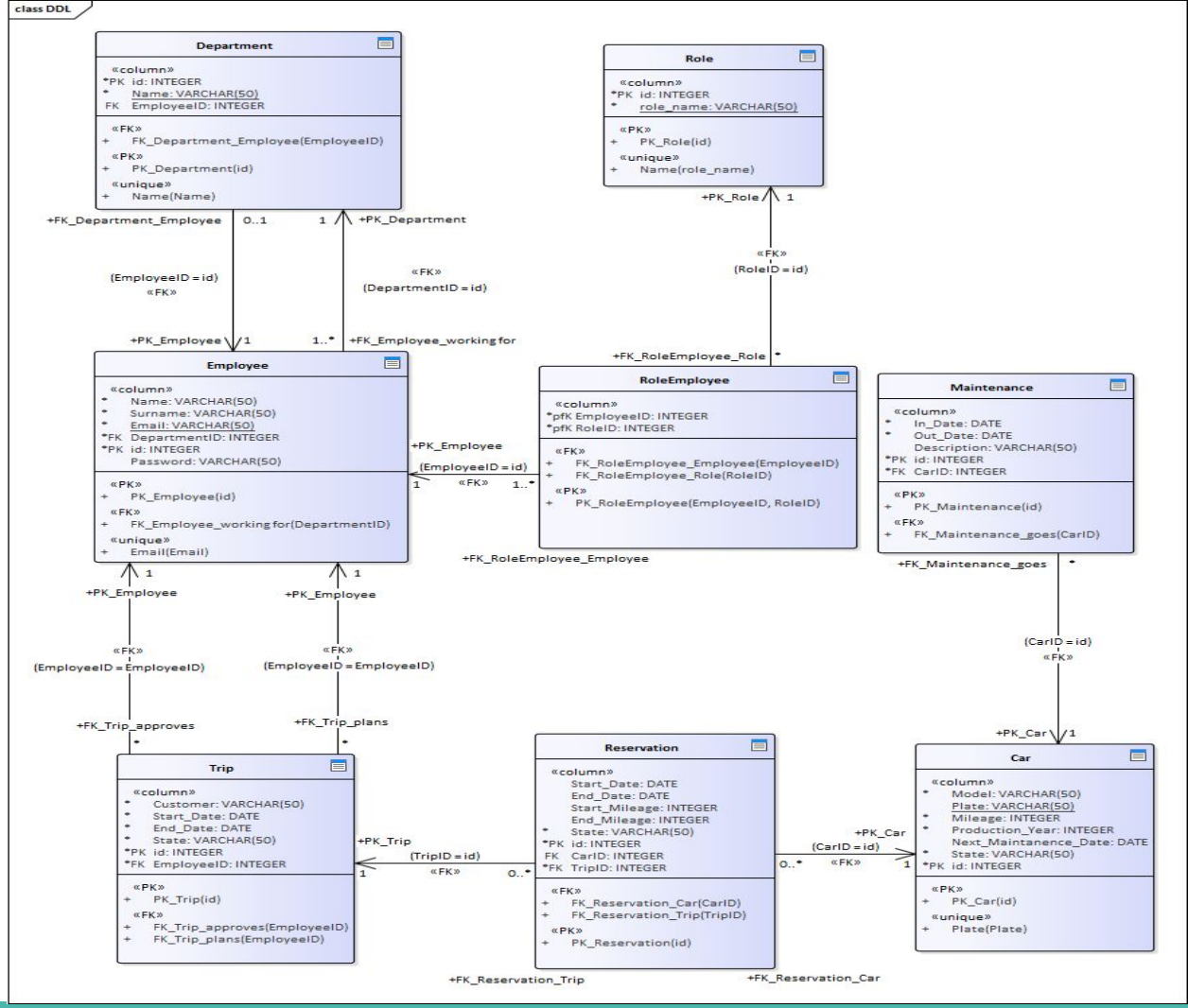


Logical Architecture

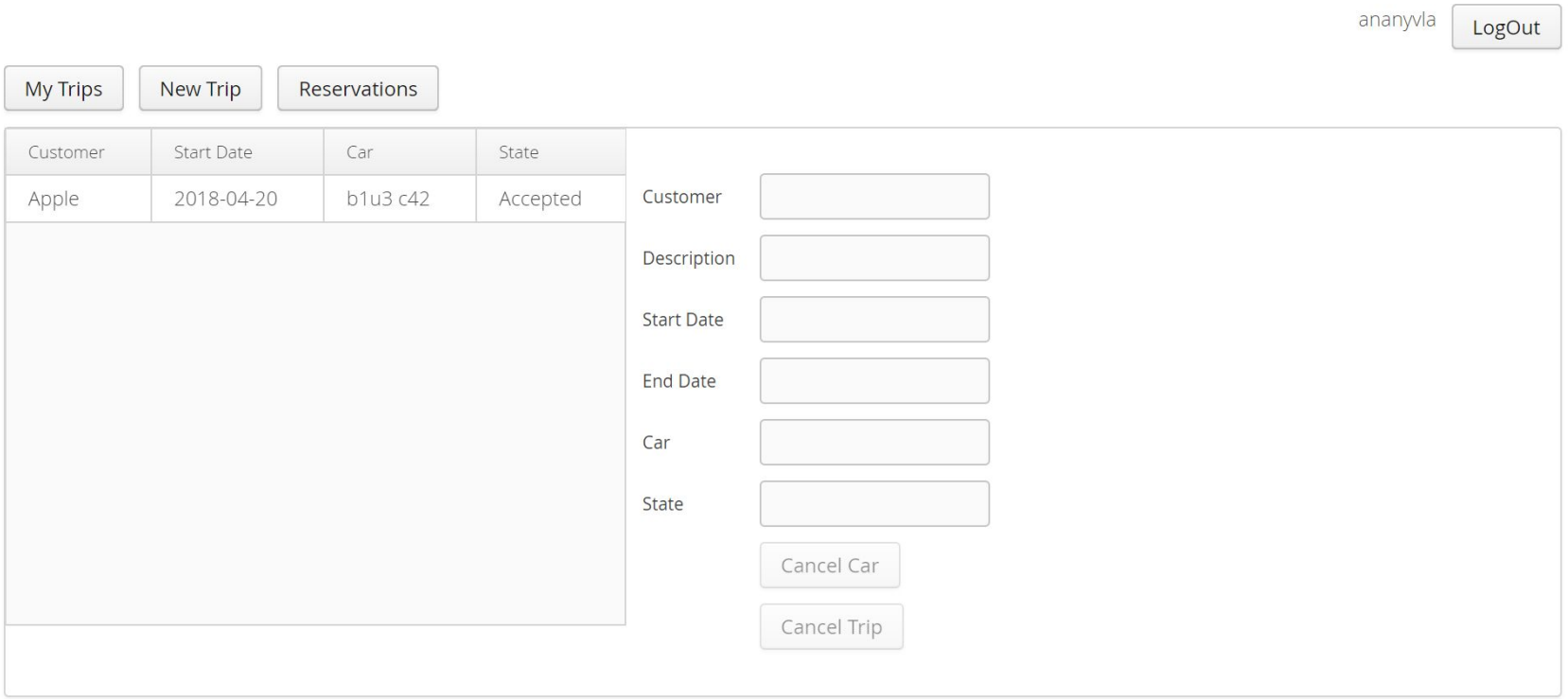
- 3 Layered Architecture
- Presentation layer - Vaadin framework
- Business layer - Controls workflow
- Data layer - Spring Data JPA



Database Model



Prototype - Presentation Layer



Prototype - Presentation Layer

```
private void setLayout() {
  HorizontalLayout menu = new HorizontalLayout();

  List<String> roles = controller.getRole(user);

  Button trips = new Button("My Trips");
  trips.addClickListener(e -> showTrips());
  menu.addComponent(trips);

  Button newTrip = new Button("New Trip");
  newTrip.addClickListener(e -> addNewTrip());
  menu.addComponent(newTrip);

  if(user.getManaging() != null) {
    Button requests = new Button("Requests");
    requests.addClickListener(e -> showRequests());
    menu.addComponent(requests);
  }

  if(roles.contains("FM")) {
    Button reservation = new Button("Reservations");
    reservation.addClickListener(e -> showReservations());
    menu.addComponent(reservation);
  }

  HorizontalLayout userBox = new HorizontalLayout();

  Label userName = new Label(user.getEmail());
  userBox.addComponent(userName);

  Button logout = new Button("LogOut");
  logout.addClickListener(e -> logoutSession());
  userBox.addComponent(logout);

  addComponents(userBox, menu);
  this.setComponentAlignment(userBox, Alignment.TOP_RIGHT);

  panel = new Panel();
  addComponent(panel);

  showTrips();
}
```

```
private void setGrid() {
  List<Trip> trips = controller.getTrips(user);
  grid = new Grid<>();
  grid.setItems(trips);
  grid.addColumn(Trip::getCustomer).setCaption("Customer");
  grid.addColumn(Trip::getStart_date).setCaption("Start Date");
  grid.addColumn(Trip::getCarPlate).setCaption("Car");
  grid.addColumn(Trip::getState).setCaption("State");
  grid.setWidth("66%");
  grid.setSelectionMode(Grid.SelectionMode.SINGLE);
  grid.addSelectionListener(e -> updateInfo(e.getFirstSelectedItem()));
  grid.setWidth("100%");

  addComponent(grid);
}

private void setInfo() {
  FormLayout form = new FormLayout();

  customer = new TextField("Customer");
  customer.setReadOnly(true);
  description = new TextField("Description");
  description.setReadOnly(true);
  start = new TextField("Start Date");
  start.setReadOnly(true);
  end = new TextField("End Date");
  end.setReadOnly(true);
  car = new TextField("Car");
  car.setReadOnly(true);
  state = new TextField("State");
  state.setReadOnly(true);

  Button cancelCar = new Button("Cancel Car");
  Button cancelTrip = new Button("Cancel Trip");
  cancelCar.setEnabled(false);
  cancelTrip.setEnabled(false);

  form.addComponents(customer, description, start, end, car, state, cancelCar, cancelTrip);
  addComponent(form);
}
```

Prototype - Controller

```java
@Override
public void acceptTrip(Trip trip, Employee user) {
  trip.setManager(user);
  trip.setState("Accepted");
  tripDao.save(trip);

  if(trip.isReservation_request()) {
    Reservation reservation = new Reservation();
    reservation.setTrip(trip);
    reservation.setState("Open");
    makeReservation(reservation);
  }
}

@Override
public boolean endTrip(Reservation reservation, Boolean damaged) {
  if(reservation.getStart_date() == null)
    return false;
  reservation.setEnd_date(LocalDate.now());
  reservation.setState("Ended");
  reservationDao.save(reservation);
  if(!damaged) return true;

  Car car = reservation.getCar();
  car.setState("Damaged");
  carDao.save(car);

  List<Reservation> openRes = reservationDao.findByCarAndState(car, "Open");
  for(Reservation r : openRes)
    makeReservation(r);

  return true;
}
```

```java
public interface ControllerInterface {
  public boolean login(String email, String pass);

  public List<String> getRole(Employee user);

  public List<Trip> getTrips(Employee user);

  public void initDB();

  public List<Trip> getTripsByDepartment(Department managing);

  public List<Reservation> getReservations();

  public boolean createTripRequest(Trip trip);

  public void acceptTrip(Trip t, Employee user);

  public void rejectTrip(Trip t, Employee user);

  public boolean startTrip(Reservation r);

  public boolean endTrip(Reservation r, Boolean damaged);
}
```

Prototype - DAO Interface & Entity

```java
public interface ReservationDao extends CrudRepository<Reservation, Long> {
  public List<Reservation> findByCarModelNot(String model);

  public List<Reservation> findByCarAndState(Car car, String open);

  public List<Reservation> findByTripEnddateBeforeAndState(LocalDate now, String open);
}
```

```java
@Entity
public class Employee {
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  private Long     id;
  private String  first_name;
  private String  second_name;
  private String  email;
  private String  password;

  @ManyToMany
  private List<Role> role;
  @ManyToOne
  private Department department;
  @OneToOne
  private Department managing;
  @OneToMany
  @JoinColumn(name = "trip_id")
  private List<Trip> trip;
  @OneToOne
  private Trip       approved_trip;
```

Thank You!