



FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE

# Fleet Management System

## Analysis

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

Authors: Berker Katipoglu  
Oleksii Tarbeiev  
Atsamaz Akopyan  
Sanan Pashayev



## Contents

1. Project Description .....	4
2. Business Model .....	5
2.1 ReservationManagementAsIs .....	5
3. Requirements .....	12
3.1 Requirements .....	12
3.1.1 Functional .....	12
3.1.1.1 F1: Keep record of car reservations .....	12
3.1.1.2 F1.1: Make a car reservation .....	13
3.1.1.3 F1.2: Close loan .....	13
3.1.1.4 F1.3: Cancel reservation .....	13
3.1.1.5 F2: Keep a record of cars .....	13
3.1.1.6 F2.1: Plan car maintenance .....	14
3.1.1.7 F2.2: Add a car .....	14
3.1.1.8 F2.3: Remove car .....	14
3.1.1.9 F3: Keep records of employees .....	15
3.1.1.10 F3.1: Add staff .....	15
3.1.1.11 F3.2: Remove staff .....	15
3.1.1.12 Requirements Satisfactory .....	15
3.1.2 Non-functional .....	17
3.1.2.1 NF1: A website .....	17
3.1.2.2 NF2: Site availability .....	17
3.1.2.3 NF3: Ease of access and comfort of use .....	17
3.1.2.4 NF4: Data backup .....	17
3.1.2.5 NF5: Website security .....	18
4. Use Case Model .....	19
4.1 Actors .....	19
4.1.1 CEO .....	19
4.1.2 Department Manager .....	19
4.1.3 Employee .....	19
4.1.4 Fleet Manager .....	20
4.2 Car Management .....	20
4.2.1 Add Car .....	21
4.2.2 Add/Remove Request .....	22
4.2.3 Confirm Requests .....	22
4.2.4 Find Car .....	22
4.2.5 Plan Maintenance .....	22
4.2.6 Remove Car .....	23
4.3 Reservation Management .....	23
4.3.1 Cancel Reservation .....	23
4.3.2 Close Loan .....	24
4.3.3 Confirm Trip Request .....	24
4.3.4 Find Reservation .....	25
4.3.5 Make Trip Request .....	25
4.3.6 Search Available Dates .....	26
4.4 Staff Management .....	26
4.4.1 Add Department .....	28
4.4.2 Add Staff .....	28
4.4.3 Delete Department .....	28



4.4.4 Delete Staff .....	28
4.4.5 Edit Department .....	29
4.4.6 Edit Staff .....	29
4.4.7 Find Staff .....	29
4.5 Requirements Satisfactory .....	29
5. Domain Model .....	31
5.1 Class Diagram .....	31
5.1.1 Car .....	32
5.1.1.1 CarStateMachine_Condition .....	32
5.1.1.2 CarStateMachine_Availability .....	33
5.1.2 Department .....	34
5.1.3 Employee .....	34
5.1.4 Maintenance .....	35
5.1.5 Reservation .....	35
5.1.5.1 ReservationStateMachine .....	35
5.1.6 Role .....	36
5.1.7 Trip .....	37
5.1.7.1 TripStateMachine .....	37



## 1. Project Description

The goal of the project is to create an application for managing a fleet of company cars and their reservations for business trips of the employees.

The system should keep record of all employees and all company cars. Information about car maintenance also should be stored in the system. Fleet Manager should be able to add a new car, remove or edit an existing car.

The system should also keep record of car reservations. Employees should be able to create reservation requests. System should automatically inform Employees about the state of the reservation request. Fleet Manager should be able to manage closing of reservations as Employees return the car from business trip. In case there is a damaged car which is reserved, system should renew the reservation with an alternative car automatically. If there is no alternative car, system should inform the employees about cancellation of the reservation.

The system should run on a server as a web application. Data from application should be backed up on a separate server periodically.

Web application should be available to all employees but access to different sections will be limited by user profiles.

## 2. Business Model

### 2.1 ReservationManagementAsIs

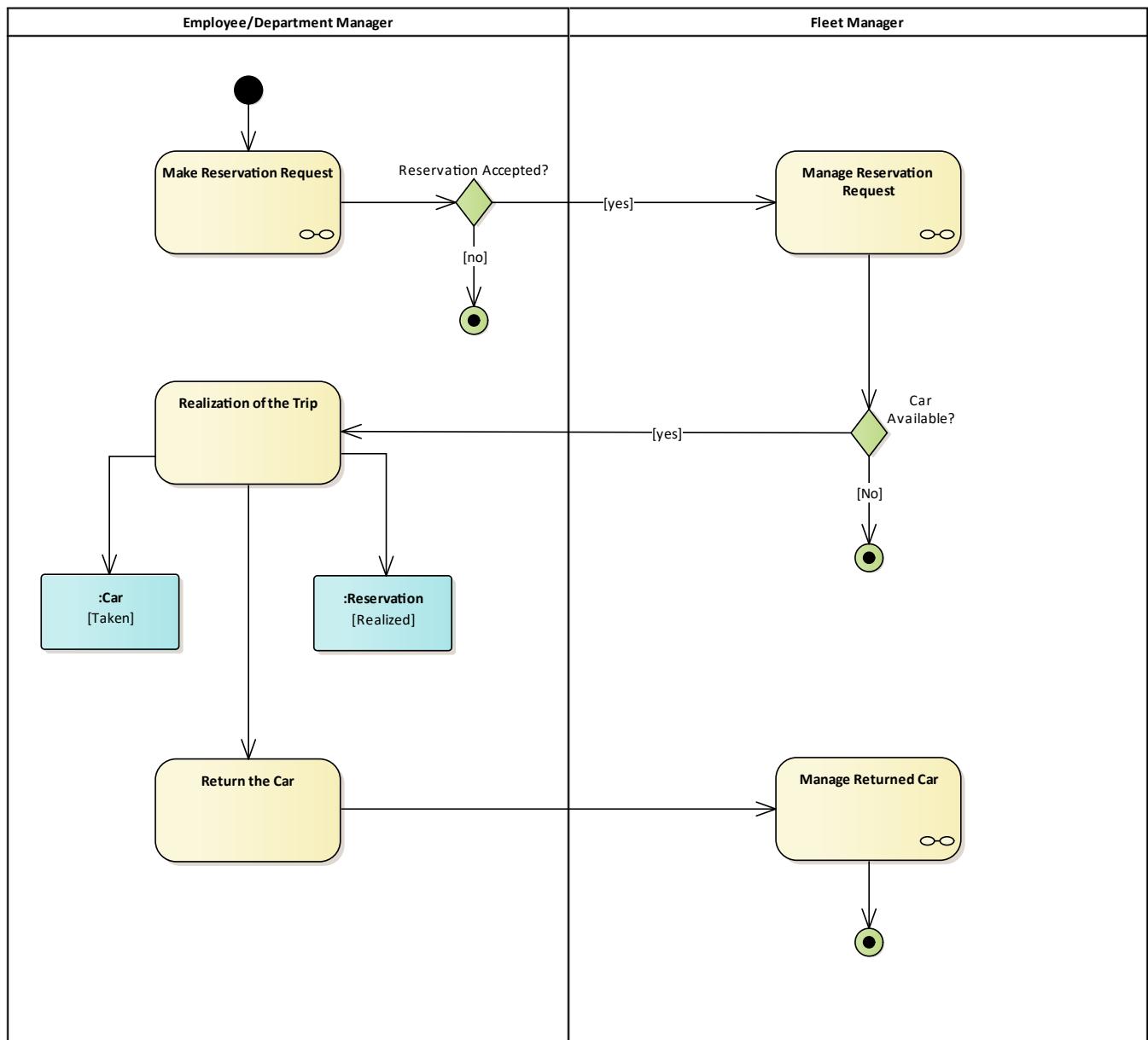


Figure 1 - ReservationProcess

This section shows the whole reservation process starting from an Employee creating business trip and car reservation

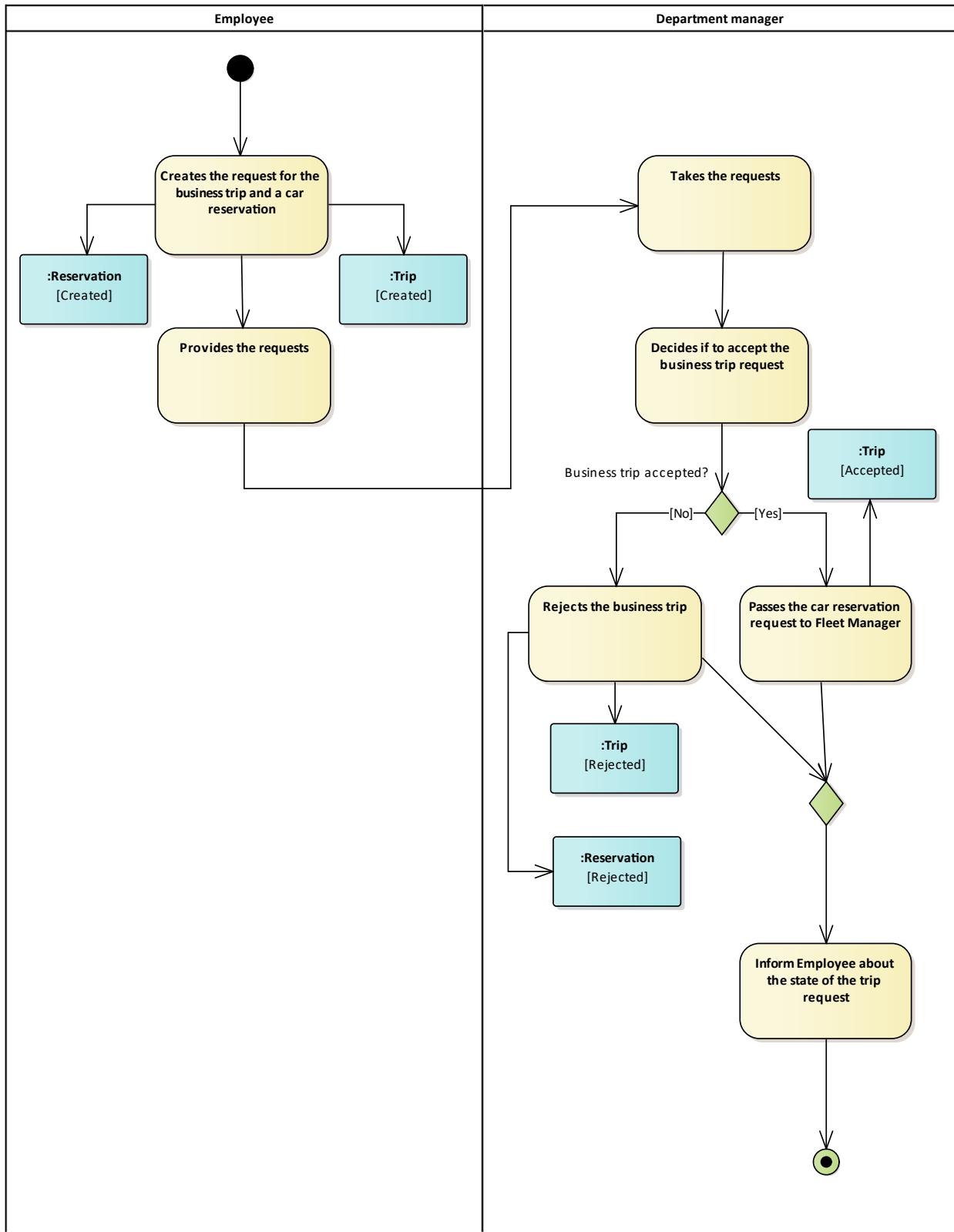


requests till returning the car and closing the reservation.

Process starts with Employee making business trip and reservation requests. Trip request is handled by the Department Manager and if approved, reservation request is passed to Fleet Manager. Employee will be informed about the trip request state.

Reservation request is handled by Fleet Manager, and Fleet Manager will inform the Employee about the state of reservation request.

If reservation request was approved, Employee realizes the trip, and returns the car. Then Fleet Manager handles the returning of the car and closes the reservation.





**Figure 2 - Make Reservation Request**

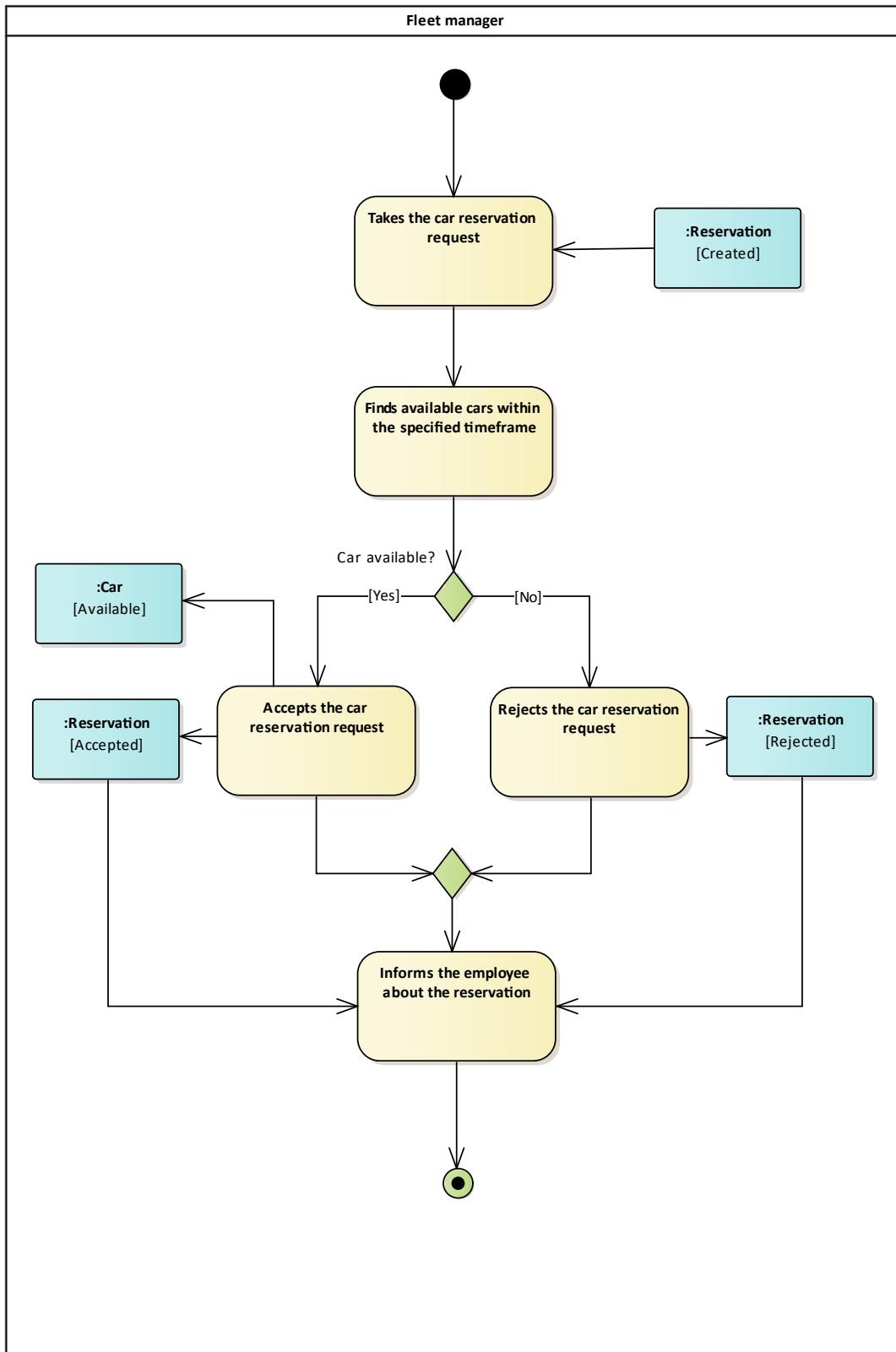
This process describes how employees make a business trip requests.

Process starts with employee creating a business trip request and a reservation request. Employee passes these requests to his department manager.

Then the Department Manager decides whether or not to accept the business trip.

If Department Manager accepts the business trip, he passes the reservation request to Fleet Manager.

Process ends informing employee about the state of trip request.



**Figure 3 - Manage Reservation Request**

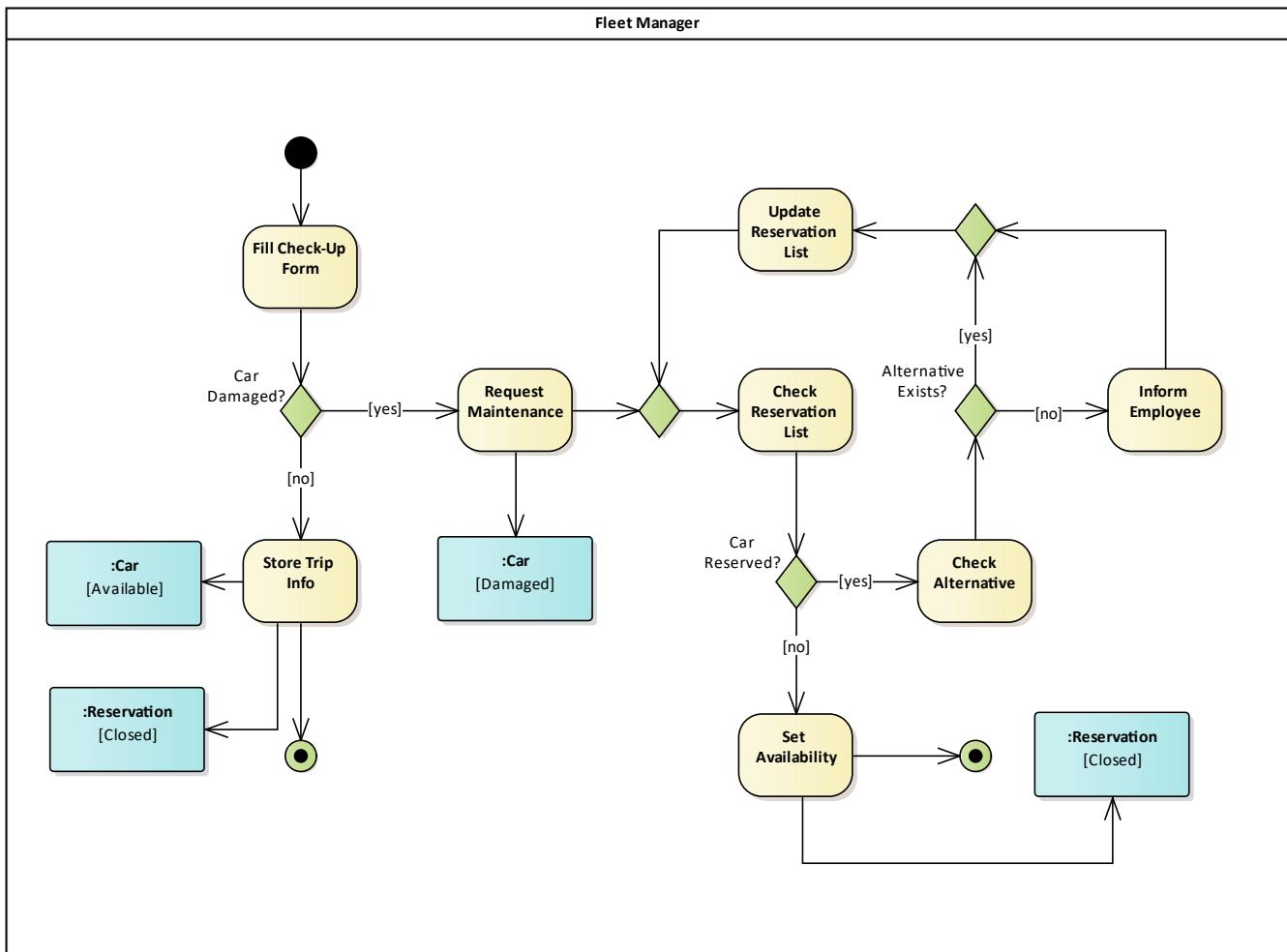
This process describes how Fleet Manager handles reservation requests.

Process starts with Fleet Manager receiving the car reservation request.

Fleet Manager checks if there is any available car during the requested dates.

If there is available car, Fleet Manager accepts the reservation request, else Fleet Manager rejects the reservation request.

Process ends with Fleet Manager informing the employee about the state of the reservation request.



**Figure 4 - Manage Returned Car**

This process describes how the reservations are closed after employee returns the car from a business trip.

The process starts when Fleet Manager fills the check-up form related to the trip information.

If the car is damaged after the trip Fleet Manager requests maintenance for the car.

If the damaged car was reserved by another employee, Fleet manager tries to reschedule the reservation with an alternative



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

car.

If there is no alternative car, Fleet Manager informs the employee about cancellation of the reservation.

The process ends with closing of the reservation.



## 3. Requirements

### 3.1 Requirements

Functional	Non-functional
<ul style="list-style-type: none"><li><input checked="" type="checkbox"/> + F1: Keep record of car reservations</li><li><input checked="" type="checkbox"/> + F1.1: Make a car reservation</li><li><input checked="" type="checkbox"/> + F1.2: Close loan</li><li><input checked="" type="checkbox"/> + F1.3: Cancel reservation</li></ul>	<ul style="list-style-type: none"><li><input checked="" type="checkbox"/> + NF1: A website</li><li><input checked="" type="checkbox"/> + NF2: Site availability</li><li><input checked="" type="checkbox"/> + NF3: Ease of access an</li><li><input checked="" type="checkbox"/> + NF4: Data backup</li></ul>

Figure 5 - Requirements

#### 3.1.1 Functional



Figure 6 - Functional

##### 3.1.1.1 F1: Keep record of car reservations

Priority: High

The system will keep record of reservations.

For each reservation the following information will be stored:

- Employee



- Car information
- Trip start date
- Car return date
- Fleet manager

Reservations will be managed by the system automatically.

System will keep record of following statistics for each employee:

- Total mileage
- Total damaged returns
- Total number of days of reservation

### **3.1.1.2 F1.1: Make a car reservation**

Priority: High

The system should provide the ability to make a car reservation by an employee.

- Employee should provide dates for the trip
- Employee can check available dates for a possible car reservation
- Department manager should approve or disapprove the trip request
- System will inform the employee if trip request is approved or not
- System will inform the employee if car is reserved for the trip or not

### **3.1.1.3 F1.2: Close loan**

Priority: High

The system should support the closing a car loan.

- If the returned car was damaged the system should update the reservations and check for alternative cars
- System should inform the employees whose reservations were affected

### **3.1.1.4 F1.3: Cancel reservation**

Priority: High

The system should support the cancellation of reservations.

- Employee can cancel a reservation
- System should update available dates for the car

### **3.1.1.5 F2: Keep a record of cars**

Priority: High

The following has to be stored for each car in the records.



- Car name
- Type of car and specifications
- Car warranty expiration date
- Car mileage and condition
- Date car was added and by whom

### **3.1.1.6 F2.1: Plan car maintenance**

Priority: Low

A car can be scheduled for maintenance by the fleet manager. To schedule a car for maintenance the following information will have to be filled.

- When is the car due for maintenance (date/time)
- To whom and where is the car going for maintenance(location)
- Why is the car going for maintenance(fault)
- Expected down time

### **3.1.1.7 F2.2: Add a car**

Priority: High

Fleet manager has the option to add a car.

When the fleet manager is choosing to add a car he/she is greeted by a window which requires the following details to be filled in:

- Car name
- Type of car and specifications
- Car warranty expiration date
- Car mileage and condition

### **3.1.1.8 F2.3: Remove car**

Priority: High

Fleet manager has the option to remove a car.

When the fleet manager chooses to remove a car he/she has is greeted by a window which requires the question "Why is the car being removed?" to be answered in textual format.

After removal of the car, car's state changed to 'Discarded' and it is not displayed in regular car searches.



### **3.1.1.9    F3: Keep records of employees.**

Priority: High

The system will keep record of employees.

For each employee, the following information needs to be stored: Name, Surname, Current department, Salary, Contact and Password.

The employees will be managed by a HR using the web application, where it will be possible to add new employees, edit information about existing employees and remove employees.

### **3.1.1.10    F3.1: Add staff**

Priority: High

HR manager has the option to add new staff.

When the HR Manager is choosing to add employee he/she is greeted by a window which requires the following details to be filled in:

- User name
- User email
- User password
- User department
- User salary

Each user will be identified with his/her unique email address.

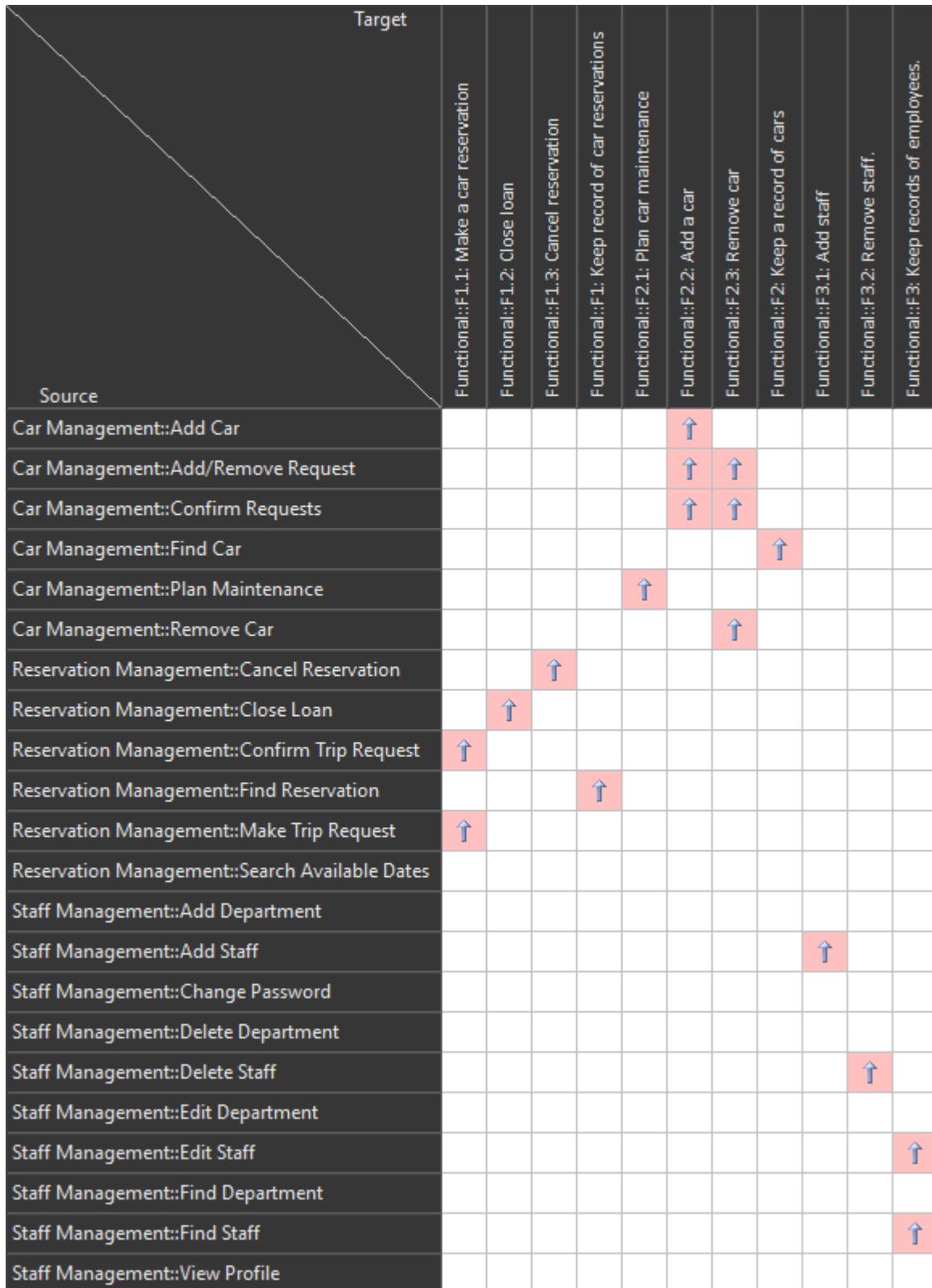
### **3.1.1.11    F3.2: Remove staff.**

Priority: High

HR Manager has the option to remove an employee.

After removal of the employee, his/her data still should be kept in the system but should not be displayed in regular searches.

### **3.1.1.12    Requirements Satisfactory**





### 3.1.2 Non-functional

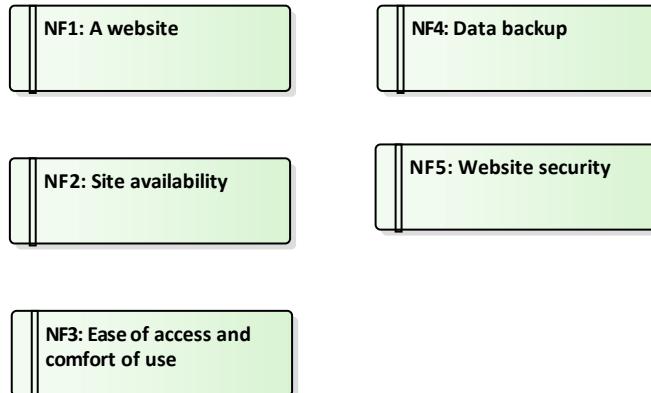


Figure 7 - Non-functional

#### 3.1.2.1 NF1: A website

Priority: High

The user will interact with a website where he/she is able to use all the features that were implemented.

#### 3.1.2.2 NF2: Site availability

Priority: High

The website must be available at all times except when under maintenance which should a rare occurrence.

#### 3.1.2.3 NF3: Ease of access and comfort of use

Priority: High.

The website must be easily accessible across a range of different devices.

Therefore, the website must be responsive without an abundance of data which must be loaded by the user such as high definition images.

Furthermore, the website must be intuitive to navigate and its color theme must not distract from whatever the user is trying to accomplish.

#### 3.1.2.4 NF4: Data backup

Priority: Low



All databases and all changes must be backed up by the server on a regular basis.

### **3.1.2.5 NF5: Website security**

Priority: High

Administrator privileges must be username and password protect so that unauthorized users are unable to edit sensitive data or damage the integrity of databases.

## 4. Use Case Model

### 4.1 Actors

This package describes the actors of the system.

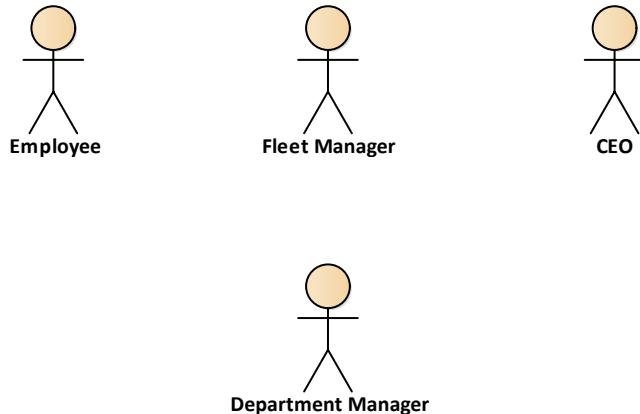


Figure 8 - Actors

#### 4.1.1 CEO

CEO represents the person who runs the company.

CEO use the system to confirm/decline car purchase and selling.

CEO has also access to all reservation history from the web application.

#### 4.1.2 Department Manager

A person who is responsible of a department in the company.

Using the web application, DM decides on the trip requests of the employees working in his department.

DM also can use the web application and see status of all reservations made by his department.

#### 4.1.3 Employee



Employee represents a person who is working in the company.

Employee can make trip requests and reserve cars using the web application.

Employee can also cancel the reservation he made and see his reservation history in the system.

#### **4.1.4 Fleet Manager**

Fleet Manager is an employee who is responsible for fleet management.

Using the system FM can:

- \* make car purchase request.
- \* make car sell request.
- \* manage car maintenance.
- \* close car loans.
- \* find information about all reservations.

#### **4.2 Car Management**

This package defines use cases regarding the management and maintenance of cars.

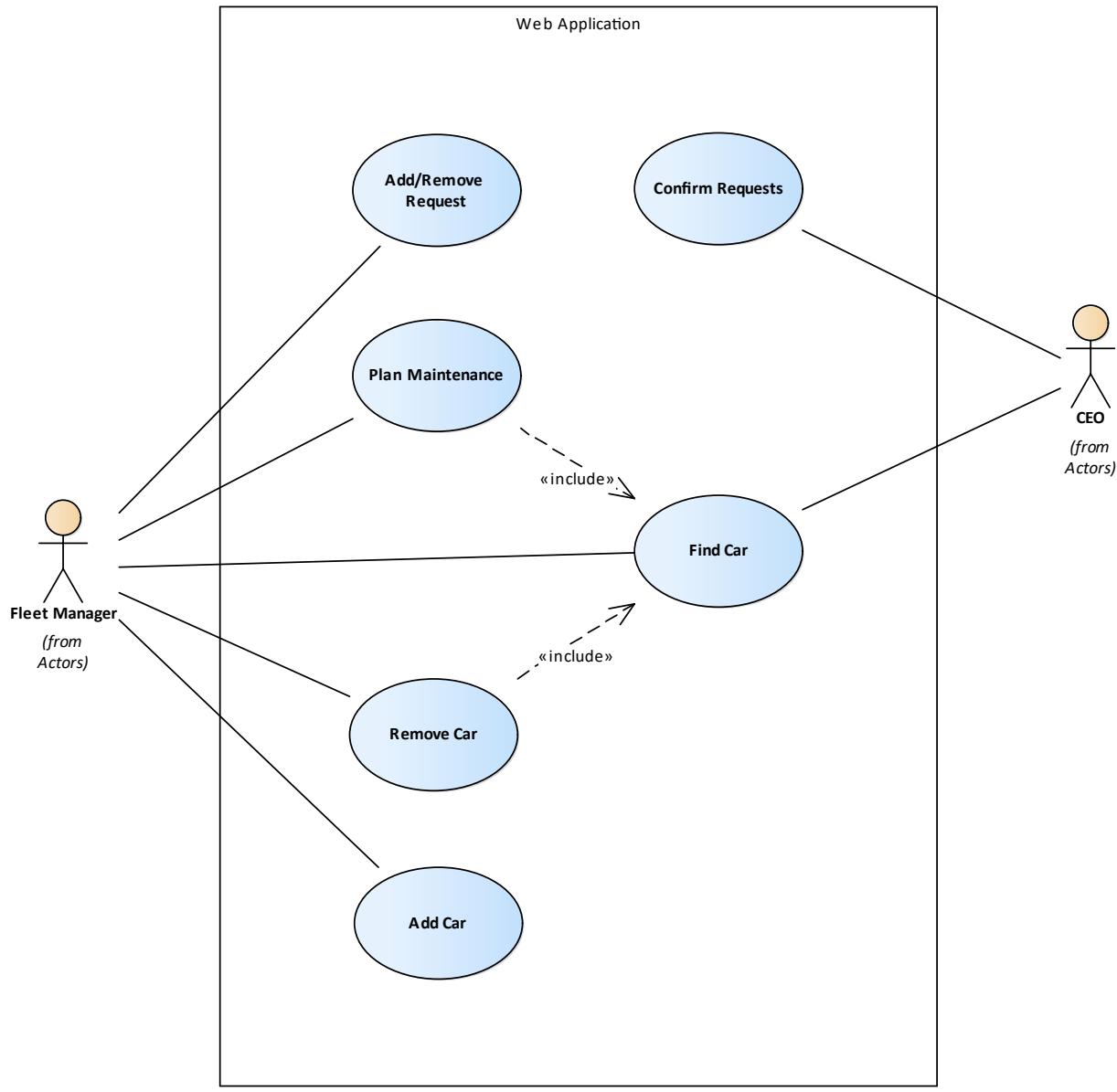


Figure 9 - CarManagement

#### 4.2.1 Add Car

Fleet Manager can add a new car to the system.

##### Basic Path: Basic Path

1. User clicks on the "Add Car" button.
2. System shows a form containing information about the car: plate number, brand, production year, description.
3. User fills all fields and confirms the action by pressing "Save" button.



4. System confirms the action and closes the window.

**Alternate: Invalid plate number**

Scenario happens if plate number is invalid.

1. System shows error message.

#### 4.2.2 Add/Remove Request

Fleet Manager can make requests regarding to purchase or removal of cars.

#### 4.2.3 Confirm Requests

Director can decide on purchase and removal requests.

#### 4.2.4 Find Car

User can find details of the car.

#### 4.2.5 Plan Maintenance

Fleet Manager can plan maintenance of the cars.

##### Basic Path: Basic Path

1. Use case starts when Fleet Manager needs to send a car to maintenance.
2. Fleet Manager finds a car which needs maintenance using find car use case.
3. System shows car's details on the screen.
4. Fleet Manager presses 'send to maintenance' button on the screen.
5. System opens another window and requires dates for the maintenance.
6. Fleet Manager fills the dates and confirms.
7. System confirms maintenance dates and closes the window.
8. System blocks the given dates for reservation and sets car's state to 'in maintenance'.
9. System checks if there are existing reservations for the given dates.
10. System replaces the existing reservations.

**Alternate: No Possible Replacement**

In this use case, it is possible that Fleet Manager needs to send a car to maintenance which is already reserved. This scenario happens if the system cannot find alternative car for existing reservations.

1. System cannot find replacement for existing reservations.
2. System cancels the reservations and informs the employees.

#### 4.2.6 Remove Car

Fleet Manager can remove car from the system.

### 4.3 Reservation Management

This package defines use cases regarding the management of loans and reservations.

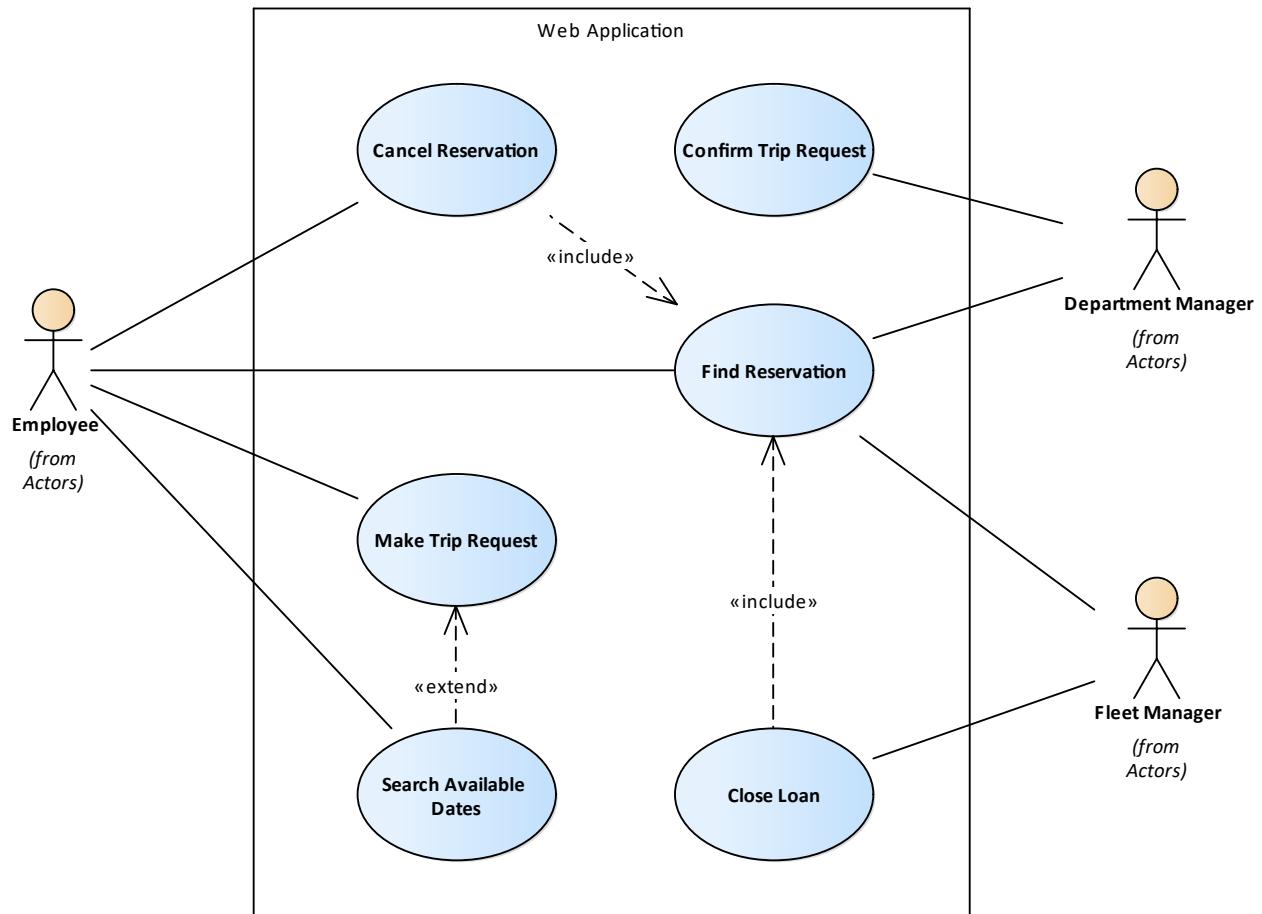


Figure 10 - ReservationManagement

#### 4.3.1 Cancel Reservation

The employee can cancel a reservation which he made.

**Basic Path: Cancel reservation**



1. The use case starts when the employee decides to cancel one of the reservations. The employee opens the web application.
2. The system presents the list of employee's current reservations.
3. The employee selects a specific reservation and presses the "view details" button.
4. The system presents a detailed view of a reservation with several action buttons.
5. The user presses the "cancel reservation" button.
6. The system presents a confirmation dialog to the user.
7. The user presses the "confirm" button.
8. The system removes the reservation from the employee's current reservations list.
9. The system updates the availability of the car.

#### 4.3.2 Close Loan

Using the system FM records returning of a loaned car by using reservation details like car id or employee name.

##### Basic Path: Closing Reservation

1. Use case starts when the user receives a car from a business trip.
2. User uses the action to close reservation.
3. <include> Find Reservation.
4. User fills the following fields in the reservation form: return date, return milage, condition.
5. User confirms closing reservation.
6. System validates closing.
7. System saves the reservation data.

##### Alternate: Late Return

Scenario happens if the Employee returns the car later than he supposed to be.

1. System detects late return of the car.
2. System sends email to reservation holder's manager with reservation details.

##### Alternate: Damaged Car

Scenario happens if the returned car is damaged.

1. System shows maintenance form with duration field.
2. User fills maintenanceform.
3. User confirms maintenance form.
4. System replaces reservations within maintanence duration.
5. System sends email to non-replaces reservation holders.
6. System blocks possible reservations of the car for maintenance duration.
7. System sends email to reservation holder's manger with reservation details.

#### 4.3.3 Confirm Trip Request

Department manager uses the system to approve business trip requests.

##### Basic Path: Manage business trip



1. Use case starts when DM decides to check whether there are new trip requests. DM opens the web application.
2. The system provides the list of incoming trip requests.
3. DM will select one of the trip requests from list and click "Detailed view button" to see description.
4. The system will show detailed informations related to trip request.
5. DM approves the requests by clicking "Approve" button.
6. System confirms the decision and closes the window.
7. System changes the status of trip request as 'Accepted'.
8. The system checks if there is a reservation request for the business trip.
9. System creates a new reservation object.
10. The system checks if there is a car available for the requested dates.
11. System fills the reservation object with reservation details and changes its status to 'Active'.
12. The system informs the employee about the status of trip and reservation status.

#### **Alternate: No reservation request**

Scenario happens if there is no reservation request related to a trip request.

1. System finds no reservation requests related to the trip request.

#### **Alternate: Rejected business trip**

Scenario happens if the trip request is rejected

1. System changes the status of business trip object to 'Declined'.

#### **Alternate: No available car**

Scenario happens if there is no available car for the reservation.

1. System changes the status of reservation request to 'Declined'.

### **4.3.4 Find Reservation**

User can search for all open and closed reservations in detail.

#### **Basic Path: Find Reservation**

1. Use case starts when user wants to find a reservation.
2. System shows a list of reservations with following fields: reservation holder, department, car plate number, reservation date, return date, return condition. System also shows filters for filtering reservations with any of the displayed fields.
3. User selects one of the reservations.
4. System shows the reservation form.

#### **Alternate: Filtering**

Scenario happens if user wants to use filters.

1. User fills in values in the filters.
2. User confirms the filtering.
3. System filters out the displayed reservations according to the provided values.

### **4.3.5 Make Trip Request**



Employee can create a business trip request using the system. During this process employee can make a reservation for a car. Employee needs to specify the details of the business trip to the system and if he wants a car reservation or not.

#### **Basic Path: Make reservation request**

1. The use case starts when the employee needs to reserve a car for a business trip. The employee opens the web application and starts making the request.
2. The system presents the following mandatory fields: business trip description, the start date and the end date.
3. The employee fills in the required fields and confirms the reservation by pressing the "send request" button.
4. The system sends the request to the department manager.
5. The system saves the reservation with the status "created" and adds it to the list of employee's current reservations.
6. The system prompts the employee about the successfull reservation request creation.

#### **Exception: Mandatory fields not filled**

1. The system detects that one of the mandatory fields was not filled.
2. The system prompts the user about the missing fields.
3. The use case continues to the step 3 of the basic path.

### **4.3.6 Search Available Dates**

Employee can search available reservation dates in the system.

#### **Basic Path: Basic Path**

1. Use case starts when an employee wants to see available dates for reservation via web application.
2. System shows list of free dates in a form of two mandatory fields for each: start date and end date.
3. Employee chooses appropriate date.

#### **Alternate: Filtering**

Scenario happens if user wants to use filters.

1. User fills in such values as day, month, year.
2. User confirms the filter.
3. System sorts all available dates in order to satisfy specified values.

## **4.4 Staff Management**

This package defines use cases regarding the management staff.

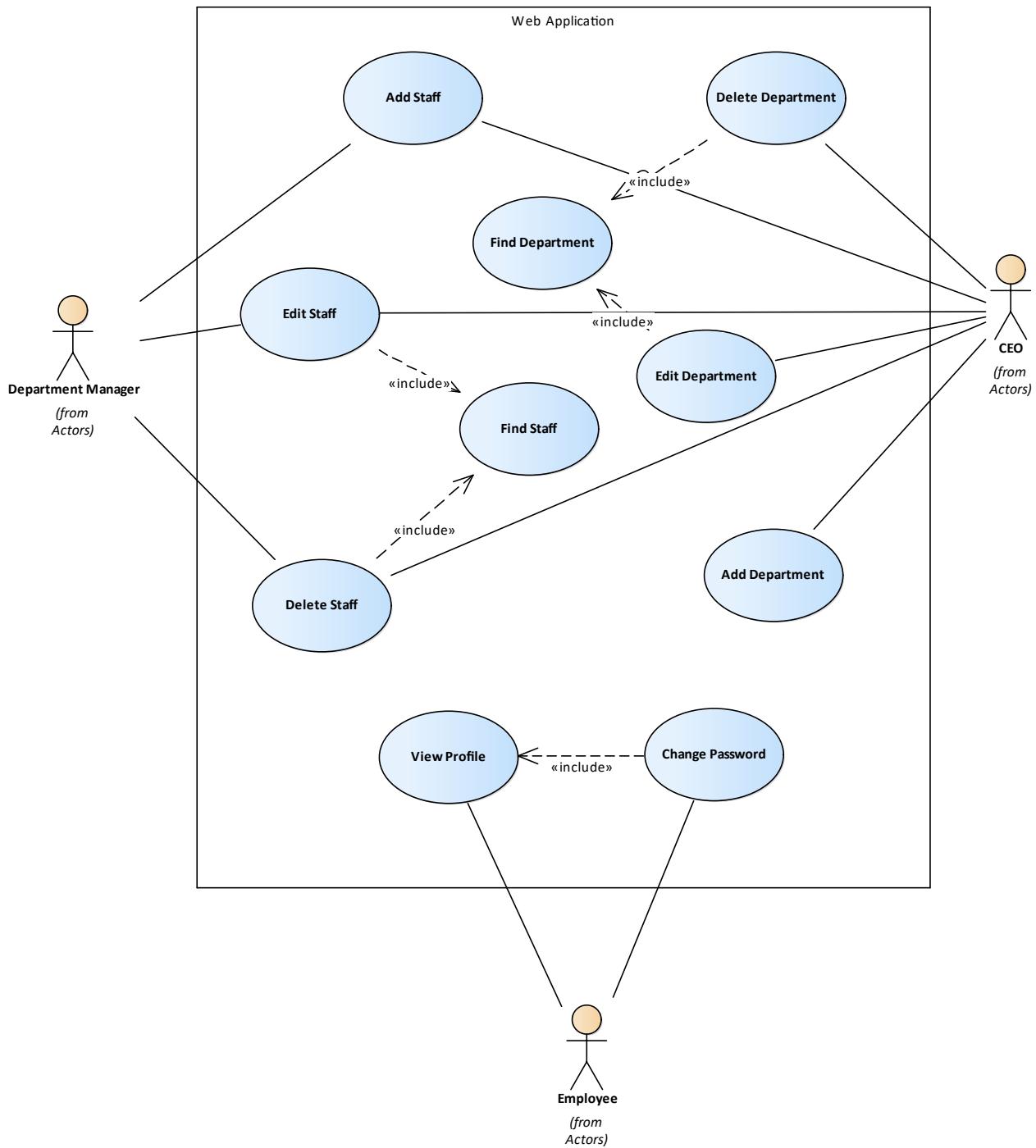


Figure 11 - StaffManagement



#### 4.4.1 Add Department

CEO can add new department to the system.

**Basic Path:** Basic Path

#### 4.4.2 Add Staff

CEO and Department Manager can add new staff members to the system.

**Basic Path:** Basic Path

1. The use case starts, when a user wants to add new employee into the system by pressing the 'Add New' button under Employee Management > Employees menu.
2. System pops a new window with a form containing following fields: Name, Surname, Email, Department, Fleet Manager.
3. The user fills all the necessary forms provided by the system.
4. The user presses the "Add Employee" button.
5. The system validates the provided information.
6. The system saves the new employee and shows a notification.

**Alternate:** Missing Input

1. The system marks the missing fields.
2. The scenario returns back to step 3 of the Basic Path.

**Exception:** Email Collision

1. System shows a warning notification.
2. The scenario returns back to step 3 of the Basic Path.

#### 4.4.3 Delete Department

CEO can delete an existing department in the system.

#### 4.4.4 Delete Staff

CEO and Department Manager can remove staff from the system.



#### **4.4.5 Edit Department**

CEO can edit an existing department in the system.

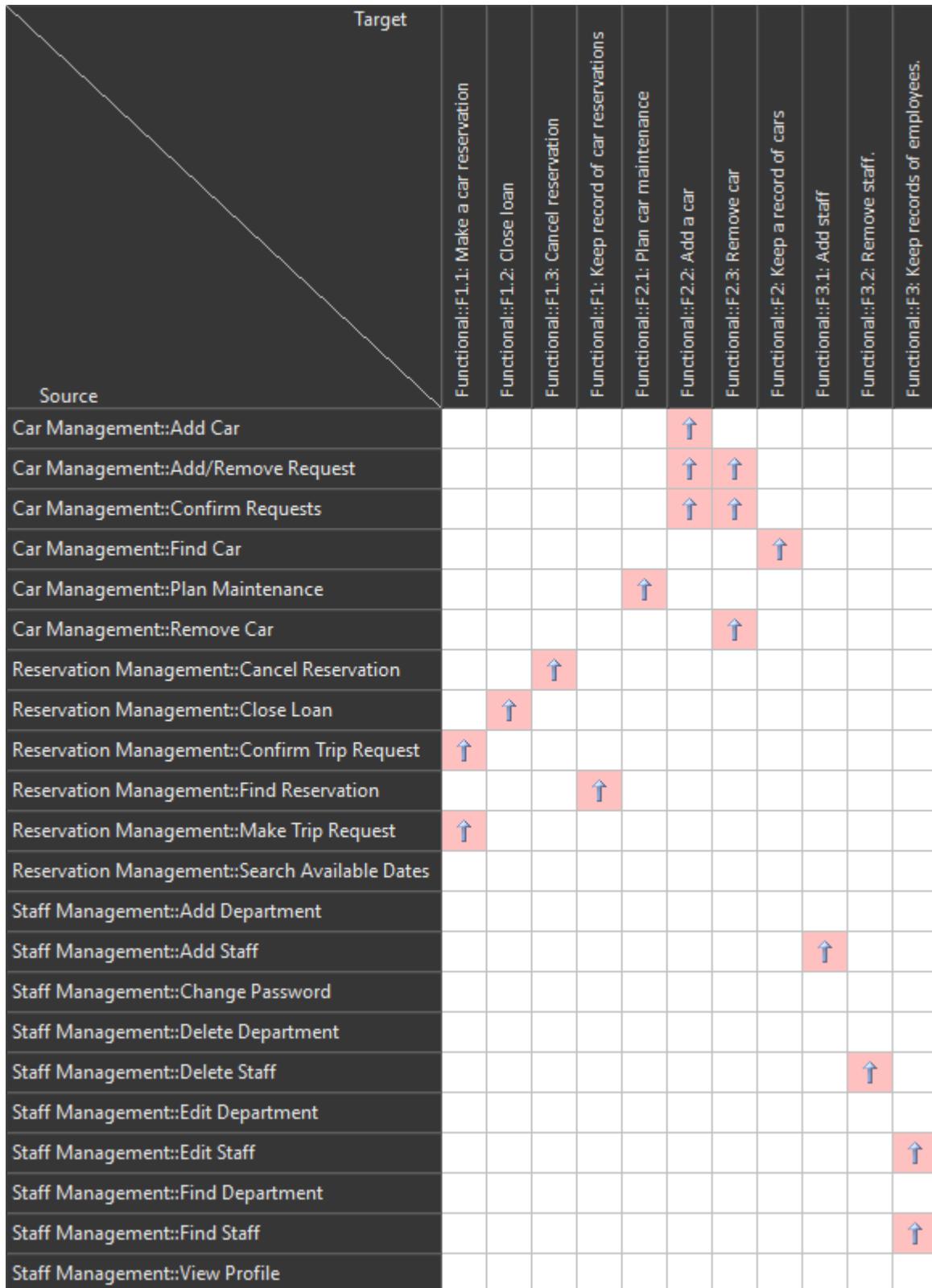
#### **4.4.6 Edit Staff**

CEO and Department Manager can edit the information of specific staff members.

#### **4.4.7 Find Staff**

CEO and Department Manager can view the list of all staff members.

### **4.5 Requirements Satisfactory**



## 5. Domain Model

### 5.1 Class Diagram

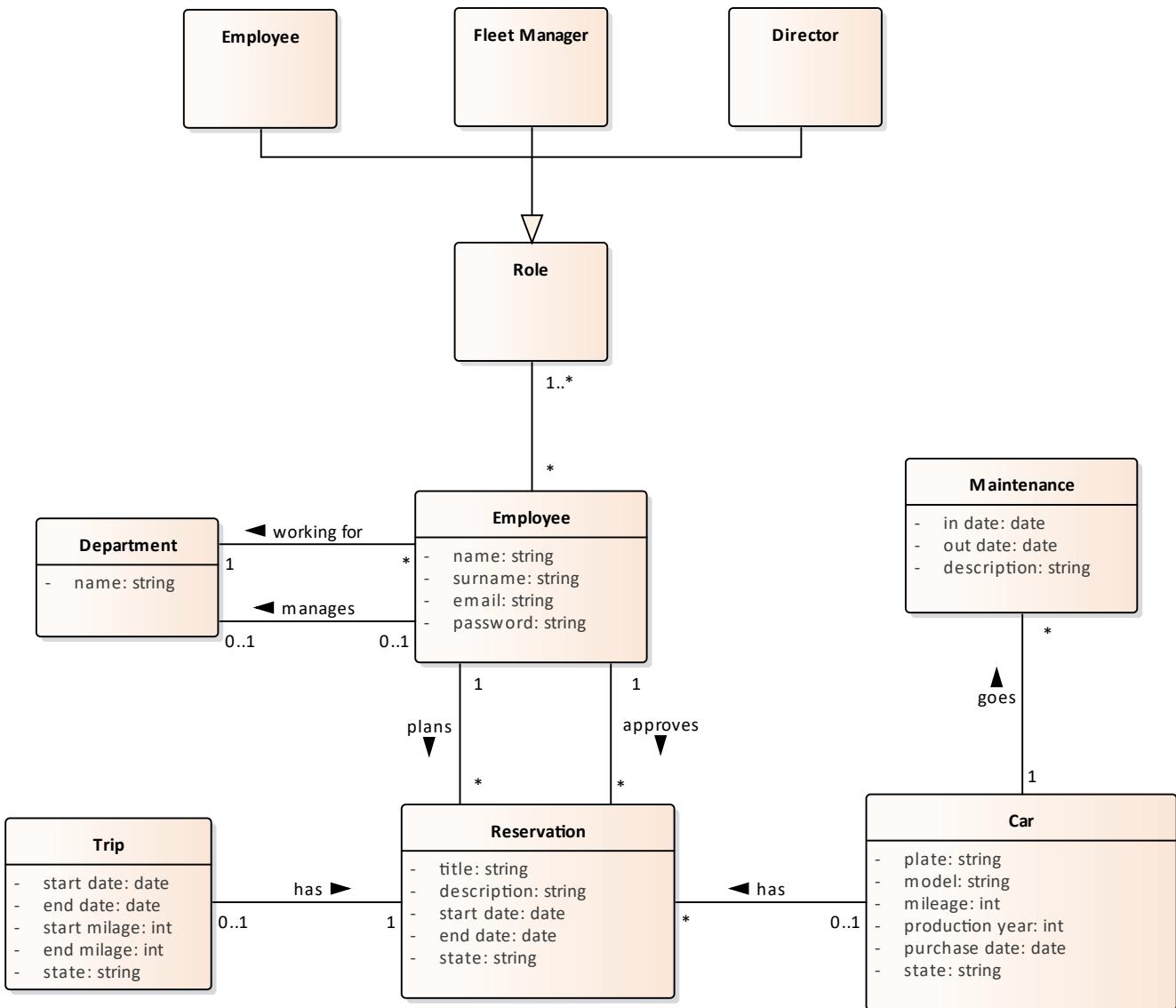


Figure 12 - ClassDiagram



### 5.1.1 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.

Attribute title	Description
plate	Plate number of the car stored as string.
model	Model of the car stored as string.
mileage	Current mileage of the car stored as integer.
production year	Production year of the car stored as integer.
purchase date	Next planned maintenance date for the car stored as date.
state	State of the car(eg. available, damaged, in maintenance etc.) stored as string.

#### 5.1.1.1 CarStateMachine\_Condition

The following diagram shows all possible condition state transitions for the car object.

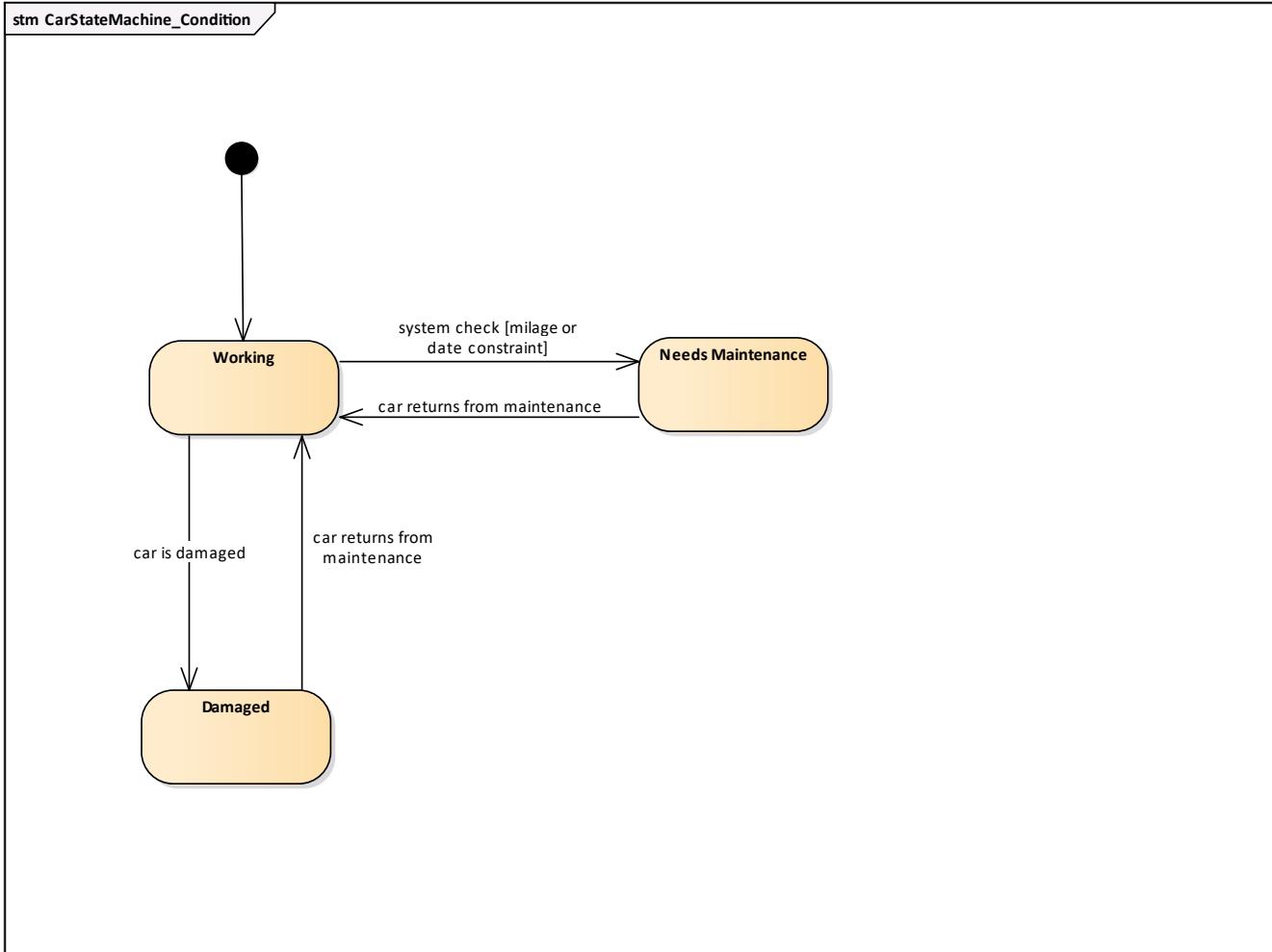


Figure 13 - CarStateMachine\_Condition

#### 5.1.1.2 CarStateMachine\_Availability

The following diagram shows all possible availability state transitions for the car object.

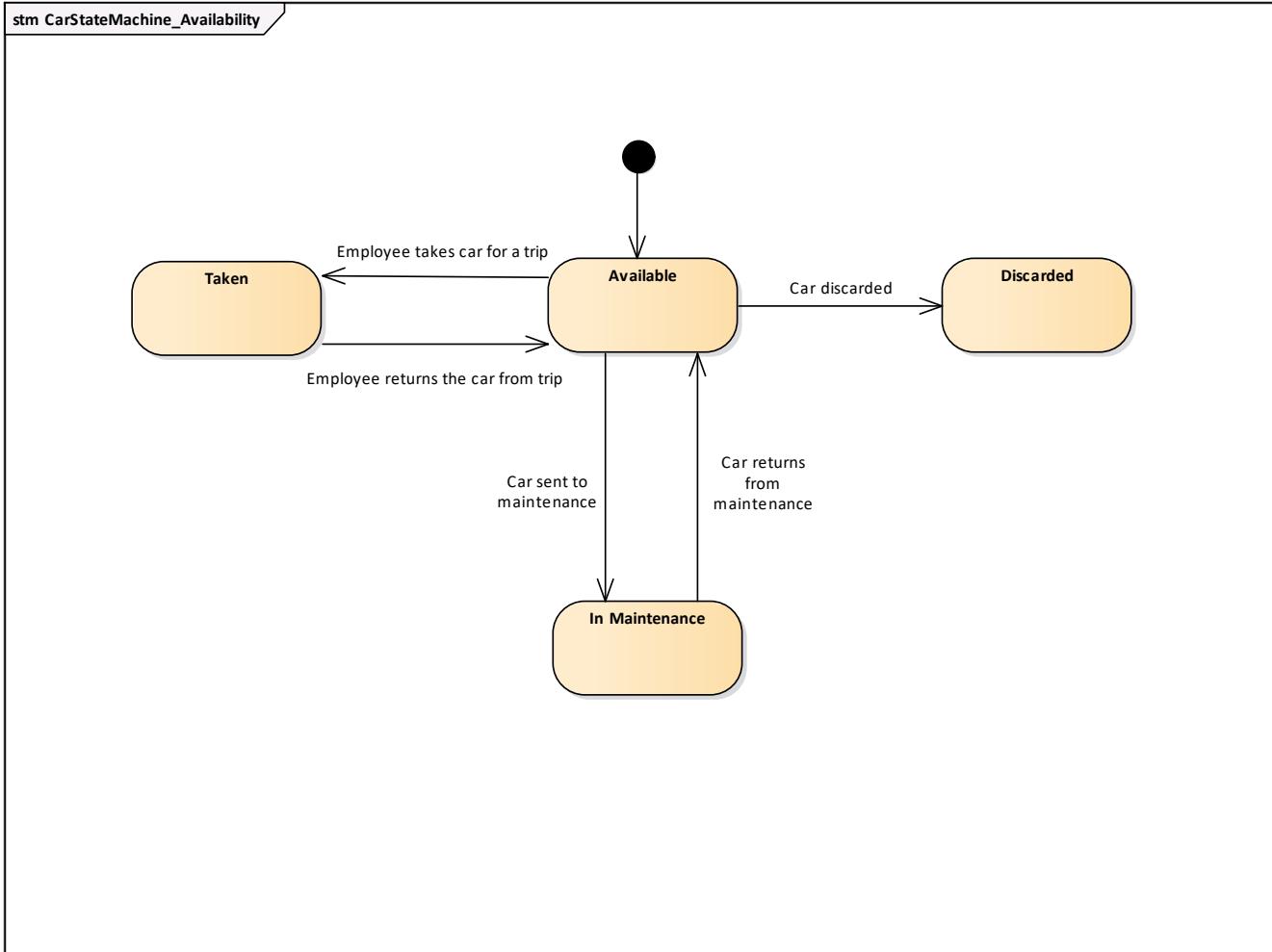


Figure 14 - CarStateMachine\_Availability

### 5.1.2 Department

Department represents different departments in the company.

Each department is managed by exactly one employee.

Each department may have one or more employees.

Attribute title	Description
name	Name of the department stored as a string.

### 5.1.3 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 manage 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.

Attribute title	Description
name	Name of employee stored as string.
surname	Surname of employee stored as string.
email	Email of employee stored as string.
password	

### 5.1.4 Maintenance

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

Attribute title	Description
in date	The date in which car was sent to maintenance stored as date.
out date	The date in which car was taken from maintenance stored as date.
description	General description about why car was sent to maintenance and what was done during maintenance stored as string.

### 5.1.5 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.

Attribute title	Description
title	Mileage of the car when employee is taking the car stored as integer.
description	Mileage of the car when employee is returning the car stored as integer.
start date	Start date of realized reservation stored as date.
end date	End date of realized reservation stored as date.
state	State of the reservation(eg. accepted, rejected, canceled, realized) stored as string.

#### 5.1.5.1 ReservationStateMachine

This diagram shows all possible state transitions for the reservation object.

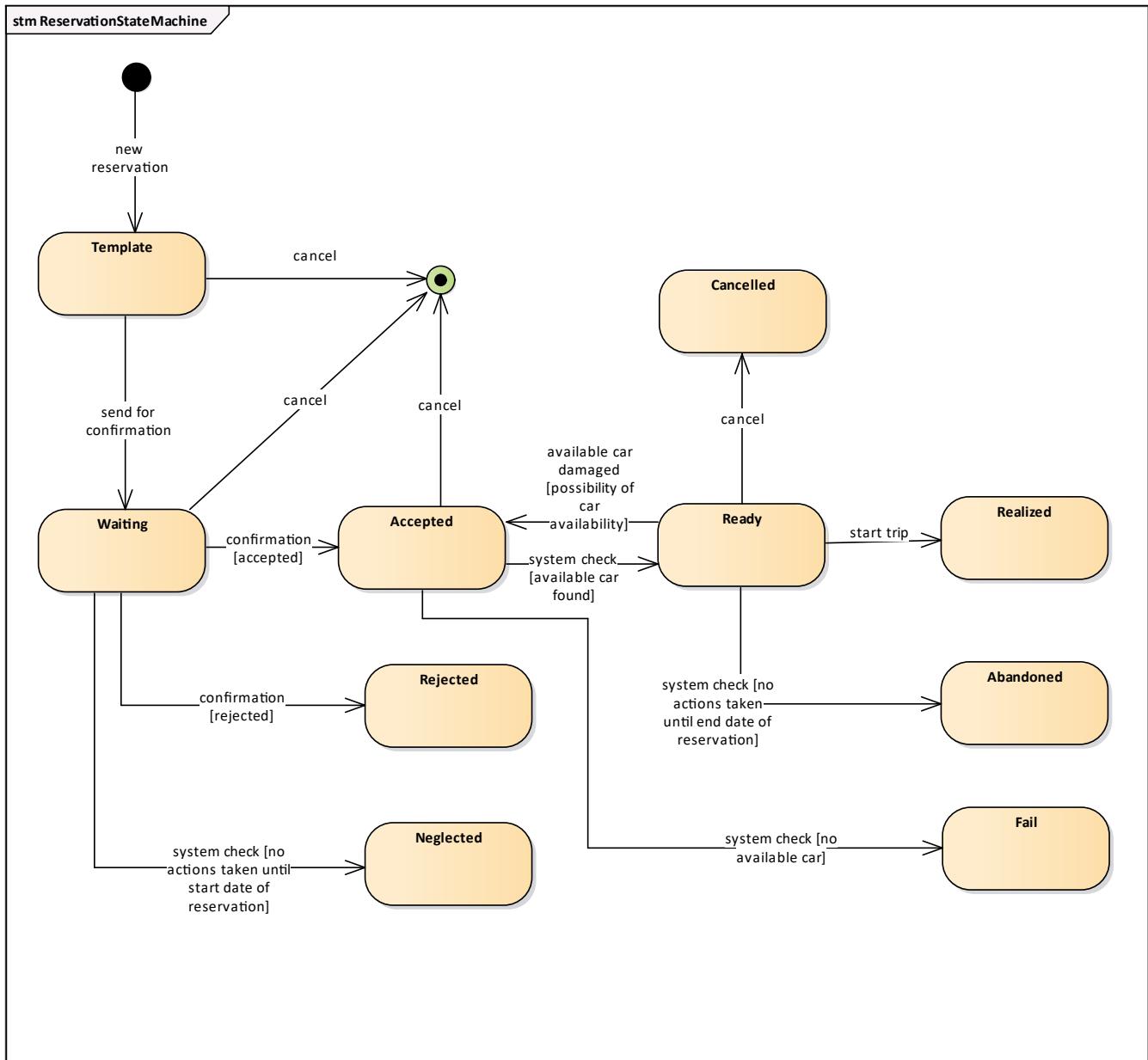


Figure 15 - ReservationStateMachine

### 5.1.6 Role

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



### **5.1.7 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.

<b>Attribute title</b>	<b>Description</b>
start date	Trip start date stored as date.
end date	Trip end date stored as date.
start milage	
end milage	
state	State of the trip(eg. accepted, rejected, cancelled) stored as string.

#### **5.1.7.1 TripStateMachine**

This diagram shows all possible state transitions for the trip class.

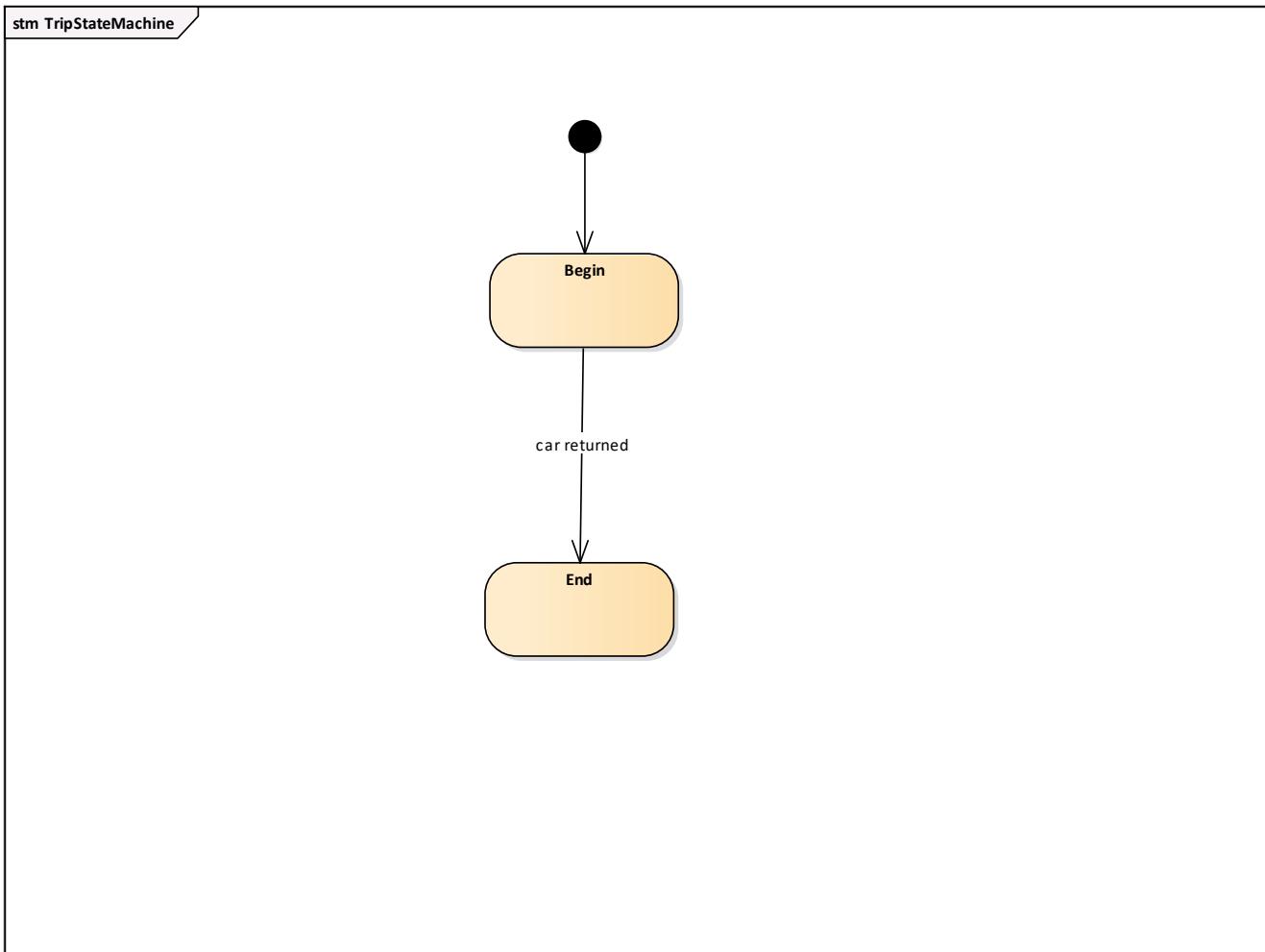


Figure 16 - TripStateMachine



FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE