



**FACULTY  
OF MATHEMATICS  
AND PHYSICS**  
Charles University

## **BACHELOR THESIS**

Lukáš Březina

## **Taxi service back-end**

Department of Distributed and Dependable Systems

Supervisor of the bachelor thesis: doc. RNDr. Tomáš Bureš, Ph.D.

Study programme: Softwarové a datové inženýrství

Study branch: Databáze a web

Prague 2018

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date .....

signature of the author

Dedication.

Title: Taxi service back-end

Author: Lukáš Březina

Department: Department of Distributed and Dependable Systems

Supervisor: doc. RNDr. Tomáš Bureš, Ph.D., Department of Distributed and Dependable Systems

Abstract: Nowadays services like Uber start to surpass taxi companies in comfort of transport. The goal of this thesis is to create a back-end part of the application, which will increase the taxi services efficiency and give the users new and more comfortable ways to use them.

Keywords: Taxi Ruby on Rails

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Goals . . . . .	2
1.2	Outline . . . . .	2
<b>2</b>	<b>Requirements</b>	<b>3</b>
<b>3</b>	<b>Analysis</b>	<b>4</b>
3.1	Application software stack . . . . .	4
3.1.1	Ruby on Rails . . . . .	4
3.1.2	PosrtgreSQL . . . . .	5
3.1.3	Sidekiq . . . . .	5
3.2	Customers . . . . .	5
3.3	Employees . . . . .	5
3.4	Vehicles . . . . .	5
<b>4</b>	<b>Title of the first chapter</b>	<b>6</b>
4.1	Title of the first subchapter of the first chapter . . . . .	6
4.2	Title of the second subchapter of the first chapter . . . . .	6
<b>5</b>	<b>Title of the second chapter</b>	<b>7</b>
5.1	Title of the first subchapter of the second chapter . . . . .	7
5.2	Title of the second subchapter of the second chapter . . . . .	7
	<b>Conclusion</b>	<b>8</b>
	<b>Bibliography</b>	<b>9</b>
	<b>List of Figures</b>	<b>10</b>
	<b>List of Tables</b>	<b>11</b>
	<b>List of Abbreviations</b>	<b>12</b>
<b>A</b>	<b>Attachments</b>	<b>13</b>
A.1	First Attachment . . . . .	13

# 1. Introduction

We have implemented first version of application in Individual Software Project which is up and running since July 2017. When I refer to the first version of application I mean this.

## 1.1 Goals

This theses has three main goals

- Create application covering order management with respect to existing taxi company processes
- Provide API for front-end(s) which is
  - secure,
  - well documented,
  - easy to use,
  - general enough for later extension to other front-end services
- Put together stack of tools which allows:
  - easy deployment with minimal down time,
  - quick installation,
  - operation system independence,
  - developer-friendly errors & system monitoring,
  - back-ups

## 1.2 Outline

## 2. Requirements

- we have 4 types of users - customers, drivers, dispatchers, administrators
- creating and va
- authorization check on all resources and endpoints,
- creation
- distribution to drivers with respect to their status and order properties
- current status modification
- canceling
- fraud management
- info for customers and dispatchers

## 3. Analysis

In the beginning of this chapter we analyze more general things like what frameworks and tools we have used and why we have made such decision. Later we go through specific parts of application and describe them.

### 3.1 Application software stack

For our application we decided to use Ruby on Rails<sup>1</sup> framework written in Ruby<sup>2</sup>. Our asynchronous jobs are handled via Sidekiq<sup>3</sup>. We decided to use PostgreSQL<sup>4</sup> as our main database engine. We also run Redis<sup>5</sup> as it is required database for Sidekiq.

#### 3.1.1 Ruby on Rails

There are many frameworks in which could be this type of application written equally well - for example ASP.NET(C#), Spring(Java), Laravel(PHP), Django(Python), ExpressJS(JavaScript).

Here are some advantages and disadvantages of Ruby on Rails which has led us to choose it.

Advantages:

- Simplicity and expressibility of Ruby - optional parenthesis, return keyword, no semicolons, combination with functional programming
- Strong Convention over Configuration influence - you have strictly given where to place models, controllers, how to name classes, database tables etc. and you are forced to do it that way. It may seem limiting at first but it brings to project clarity and most of the times it gives you good way to solve your problem without reinventing wheel
- plenty of tools built in - from the routing and security through development-testing-production configurations to the highly sophisticated ORM
- global repository of libraries (Ruby Gems) - most of them in very good quality with clear documentation and test covered
- We are using it for 3 years, so we know proven libraries and ecosystem

Disadvantages:

- it is more difficult to set it up than PHP
- impossible to use standard web hosting
- small base of programmers knowing Ruby
- efficiency compared to some framework

---

<sup>1</sup> Ruby on Rails framework main page <https://rubyonrails.org/>

<sup>2</sup> Ruby language main page <https://www.ruby-lang.org/>

<sup>3</sup> Sidekiq wiki page <https://github.com/mperham/sidekiq/wiki>

<sup>4</sup> PostgreSQL database main page <https://www.postgresql.org/>

<sup>5</sup> Redis database main page <https://redis.io/>



### **3.1.2 PostgreSQL**

We decided to use PostgreSQL, because it is open source and unlike MySQL it supports natively storing JSON, arrays and it has many plugins - for example for storing geo data. None of these features we use in our application now but why not to have this possibility when we would like to optimize something or extend it. Since we use Rails ORM (ActiveRecord), choice of the database is not so critical - we can migrate later to other database.

### **3.1.3 Sidekiq**

There are many job processors for Ruby<sup>6</sup>. Ruby on Rails has own abstraction called ActiveJob.

## **3.2 Customers**

## **3.3 Employees**

## **3.4 Vehicles**

---

<sup>6</sup> Job processors comparasion <http://api.rubyonrails.org/classes/ActiveJob/QueueAdapters.html>

## 4. Title of the first chapter

An example citation: Anděl [2007]

### 4.1 Title of the first subchapter of the first chapter

### 4.2 Title of the second subchapter of the first chapter

## 5. Title of the second chapter

5.1 Title of the first subchapter of the second chapter

5.2 Title of the second subchapter of the second chapter

# Conclusion

# Bibliography

J. Anděl. *Základy matematické statistiky*. Druhé opravené vydání. Matfyzpress, Praha, 2007. ISBN 80-7378-001-1.

# List of Figures

# List of Tables

# List of Abbreviations



# A. Attachments

## A.1 First Attachment