**Software Design Specification**

1. **Introduction**

This section describes the design of a simple rental car booking API service for the new car rental company(CRC).

**1.1 Goals and Objectives**

The main purpose of CRC is to help company manage to serve customers(using the mobile app, or web client) to reserve a car for a period of time.

The goals of CRC are:

* To rent a car online.
* To promote efficiency of daily business.

The goals of CRC API are:

* Keep it simple

**1.2 System Statement of Scope**

In fact, the company is too new to involve complex business, therefore, concurrent connection will not be an issue for implementation.

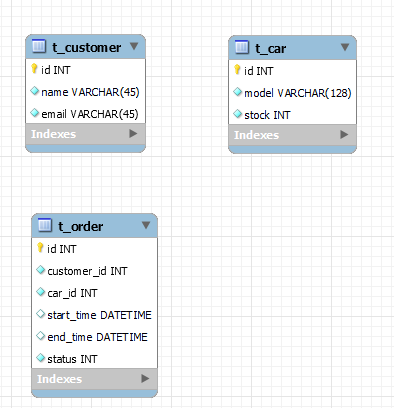
**1.3 Major Constraints**

Time

Only have about four days to finish all documentation, coding, test and deployment. A lot of ideas and functions cannot be implemented due to time constraint. Only complete some basic functionalities.

1. **Data Design**

**2.1 database description**



**2.2 DDL**

|  |
| --- |
| CREATE TABLE IF NOT EXISTS `test`.`t\_customer` (  `id` INT NOT NULL AUTO\_INCREMENT,  `name` VARCHAR(45) NOT NULL,  `email` VARCHAR(45) NOT NULL,  PRIMARY KEY (`id`))  ENGINE = InnoDB;  CREATE TABLE IF NOT EXISTS `test`.`t\_car` (  `id` INT NOT NULL AUTO\_INCREMENT,  `model` VARCHAR(128) NOT NULL,  `stock` INT UNSIGNED NOT NULL,  PRIMARY KEY (`id`))  ENGINE = InnoDB;  CREATE TABLE IF NOT EXISTS `test`.`t\_order` (  `id` INT NOT NULL AUTO\_INCREMENT,  `customer\_id` INT NOT NULL,  `car\_id` INT NOT NULL,  `start\_time` DATETIME NULL,  `end\_time` DATETIME NULL,  `status` INT NOT NULL COMMENT '0-renting 1-returned 2-should be returned',  PRIMARY KEY (`id`))  ENGINE = InnoDB; |

1. **Architectural and Component-level Design**

**3.1 program structure**

Using MVC pattern to build the whole project.

MVC pattern stands for Model-View-Controller. In this project, view layer is separated to front-end, mainly design the model and controller layers.

**3.2 Objects recognition and analysis**

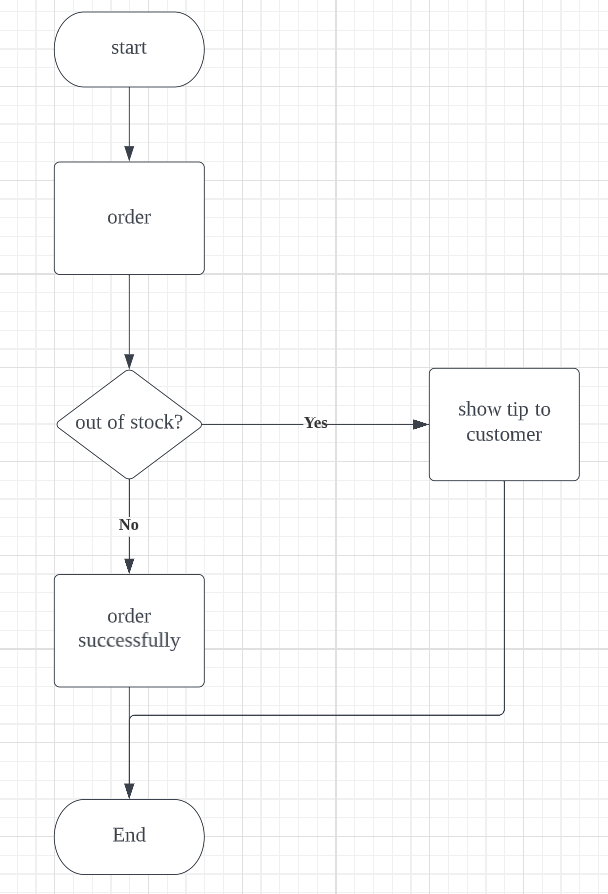
The company daily service is to provide cars for customers. So, there are basic three objects. the core business objects, car, customer, order

**3.3 Technology selection**

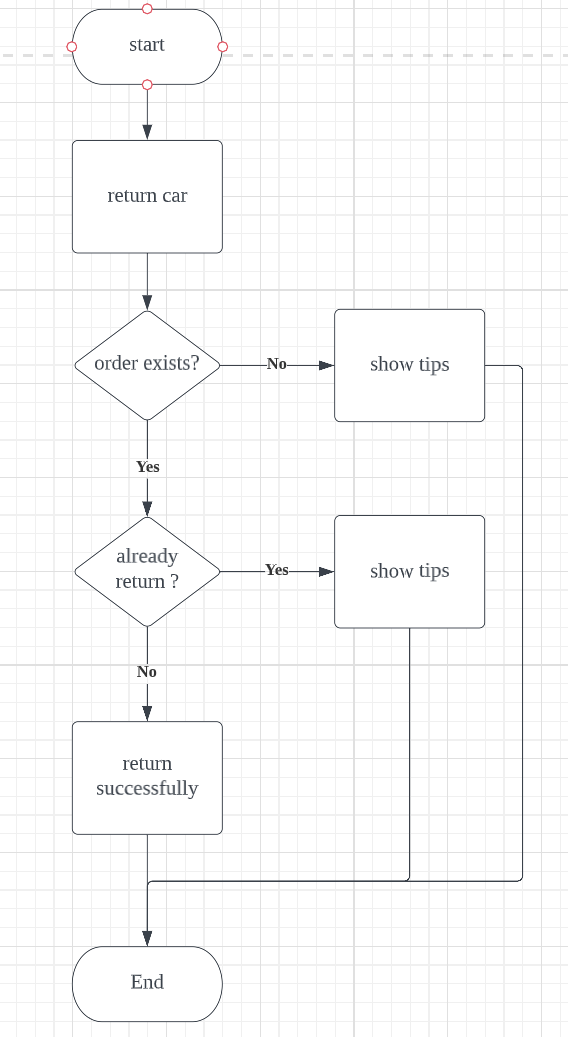
* Programming language: java
* Framework: Spring Boot + JPA
* Deployment environment: docker

1. **business logic design**

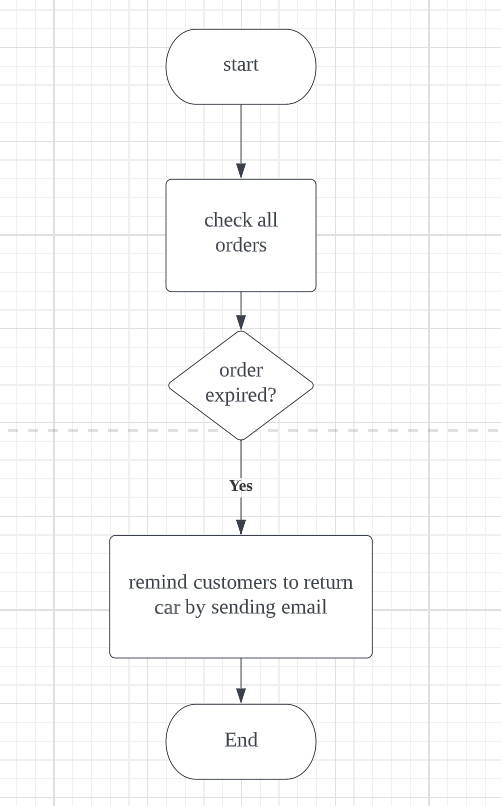
**4.1 Rent a car**



**4.2 Return a car**



**4.3 Remind customer to return car**



**5.0 API spec**

The response data’s format is JSON. Basic structure is

|  |
| --- |
| {  “code”: 200,  “msg”: “some tips”  “data”: {…}  } |

**5.1 Rent a car**

* Path: /rent/car
* Method: POST
* Parameters

|  |  |
| --- | --- |
| Parameter Name | Remark |
| carId | The id of the table car |
| customerId | The id of the table customer |
| startTime | Renting start time |
| endTime | Renting end time |

**5.2 Return a car**

* Path: /return/car
* Method: POST
* Parameters

|  |  |
| --- | --- |
| Parameter Name | Remark |
| customerId | The id of the table customer |
| orderId | The id of the table order |