

# CSCI927 Service-Oriented Software Engineering (Progress Report)

## An application for online car-hailing services

*Group Members (Group 17): Wenrui Chen, Lu Xiong, Bing Xia, Yunbo Zhang*

This project is an application for online car-hailing services. We have four group members. Wenrui Chen is responsible for the process of qualifications view, Lu Xiong is responsible for the process of customer hailing and the writing of the report, Bing Xia designs the process of the allotment of the order to cars, Yunbo Zhang designs the process of drivers. Our process would be designed by BPMN and CMMN.

In this stage, what our group has done will be shown below:

Firstly, we modify some details in the part of the process of customer hailing.

### 1) Download apps

Clients know about a sharing car application, they would download the app. The clients need create accounts by using personal information, and set up emergency contacts.

### 2) Ask for sharing cars

When clients need sharing cars, the clients open the application and ask for cars.

### 3) Calling accepted or queue

If orders are accepted, clients can see locations of cars, and know details of cars. If callings are not accepted, clients queue and they can know the number of people queued. When orders are accepted, clients can know locations and details. And if orders are not accepted within 10 minutes, callings would be accepted by the nearest cars. Then clients can wait for cars.

### 4) Drive to destinations

The clients get on sharing cars, and go to destinations.

### 5) Leave the sharing car

When clients are arrived safely, clients pay the money and get invoices. Clients should check their personal things, and get off cars. Clients also can do evaluations for orders. This is the end of transactions.

### 6) Solve possible problems in orders

If there are any emergencies during driving, clients can contact the police and customer services looking for help. After accidents are handled, clients can get some compensation and this is the end of transactions.

Secondly, because BPMN can clearly show our project and is beneficial to understand by other people. We draw BPMN for all parts and XML of BPMN. And there are some activities that may be performed in an unpredictable order in response to evolving situations. So we picture CMMN for some services or small parts in this project. These BPMN, XML and CMMN will be shown in the appendix.

In the next stage, we will describe some more details designed by CMMN and its XML, and do progress mining analysis to describe our project more clearly.

# Appendix

## Project Title: An application for online car-hailing services

*Group Members (Group 17): Wenrui Chen, Lu Xiong, Bing Xia, Yunbo Zhang*

This project is an application for online car-hailing services. We have four group members. Wenrui Chen is responsible for the process of qualifications view, Lu Xiong is responsible for the process of customer hailing and the writing of the report, Bing Xia designs the process of the allotment of the order to cars, Yunbo Zhang designs the process of drivers. Our process would be designed by BPMN and CMMN. In the process of qualifications review, first of all, the driver needs to apply to the regulatory authority and submit personal data, and the regulatory authority receiving the request will review the qualifications of the online car hailing driver. After passing the exam, the driver can take part in the exam. After passing the examination, the driver can be given a license to drive a car online and obtain a three-year valid qualification.

### I) Application

Fill in the application form online and submit photos, ID cards and photocopies of driver's license.

### II) Qualification examination

- 1) The driver has registered permanent residence or temporary residence permit of the city for more than 6 months or has obtained a temporary residence permit for more than 12 months;
- 2) Obtain the driving license of the corresponding quasi driving vehicle type and have more than 3 years of driving experience;
- 3) There is no traffic accident crime, dangerous driving criminal record, drug use record, driving record after drinking, and no record of 12 points in the last three consecutive scoring cycles;
- 4) No violent criminal record;
- 5) There is no record of the revocation of the taxi driver's qualifications certificate in the first three years since the date of application for the examination.

### III) Test appointment

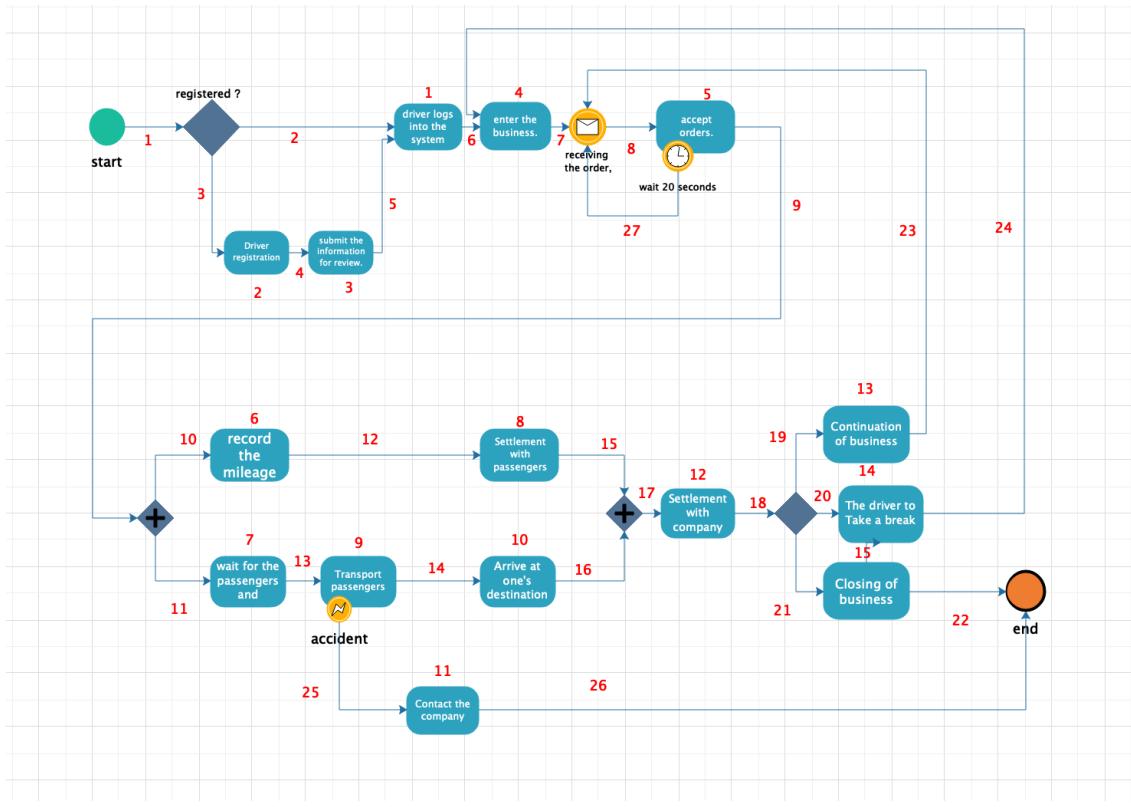
If the examination is passed, the applicant can make an appointment for the examination through the management information system. After the appointment is successful, the examination permit can be printed or saved.

### IV) Taking the exam

The applicant shall take the united online computer examination with the printed or saved examination permit. At the end of the test, the scores will be automatically corrected and the test results will be recognised on the spot. For the corresponding examination subjects, if they fail to pass the examination for the first time, they can make up an appointment 15 days later; if they fail to pass the supplementary examination, they can make up an appointment 30 days later.

### V) Review and issue driver's license

If the national public subjects and regional subjects pass the examination, the municipal transportation administration department shall examine and issue the driver's license, and the applicant can print or save it through the management information system.



XML of BPMN

```

<bpmn:process id="Process_1" isExecutable="true">
  <bpmn:startEvent id="startEvent_1">
    <bpmn:outgoing>SequenceFlow_1</bpmn:outgoing>
    <bpmn:conditionalStartDefinition />
  </bpmn:startEvent>

  <bpmn:SequenceFlow id="SequenceFlow_1" sourceRef="StartEvent_1" targetRef="exclusiveGateway_1">
    <bpmn:exclusiveGateway id="exclusiveGateway_1" name="registered?">
      <bpmn:incoming>SequenceFlow_1</bpmn:incoming>
      <bpmn:outgoing>SequenceFlow_2</bpmn:outgoing>
      <bpmn:outgoing>SequenceFlow_3</bpmn:outgoing>
    </bpmn:exclusiveGateway>
  </bpmn:SequenceFlow>

  <bpmn:SequenceFlow id="SequenceFlow_2" sourceRef="exclusiveGateway_1" targetRef="Task_1">
  <bpmn:SequenceFlow id="SequenceFlow_3" sourceRef="exclusiveGateway_1" targetRef="Task_2">

  <bpmn:task id="Task_1" name="Driver logs the system">
    <bpmn:incoming>SequenceFlow_2</bpmn:incoming>
    <bpmn:incoming>SequenceFlow_5</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_5</bpmn:outgoing>
    <bpmn:outgoing>SequenceFlow_6</bpmn:outgoing>
  </bpmn:task>

  <bpmn:task id="Task_2" name="Driver registration">
    <bpmn:incoming>SequenceFlow_3</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_4</bpmn:outgoing>
  </bpmn:task>

  <bpmn:SequenceFlow id="SequenceFlow_4" sourceRef="Task_2" targetRef="Task_3">
  <bpmn:task id="Task_3" name="submit the information for review.">
    <bpmn:incoming>SequenceFlow_4</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_5</bpmn:outgoing>
  </bpmn:task>

  <bpmn:SequenceFlow id="SequenceFlow_5" sourceRef="Task_3" targetRef="Task_1">
  <bpmn:SequenceFlow id="SequenceFlow_6" sourceRef="Task_1" targetRef="Task_4">

  <bpmn:task id="Task_4" name="enter the business.">
    <bpmn:incoming>SequenceFlow_6</bpmn:incoming>
    <bpmn:incoming>SequenceFlow_24</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_7</bpmn:outgoing>
  </bpmn:task>

  <bpmn:SequenceFlow id="SequenceFlow_7" sourceRef="Task_4" targetRef="IntermediateCatchEvent_1">
  <bpmn:IntermediateCatchEvent id="IntermediateCatchEvent_1">
    <bpmn:incoming>Sequence_23</bpmn:incoming>
  </bpmn:IntermediateCatchEvent>

```

```

<bpmn:IntermediateCatchEvent id="IntermediateCatchEvent_1">
  <bpmn:incoming>Sequence_23</bpmn:incoming>
  <bpmn:incoming>Sequence_27</bpmn:incoming>
  <bpmn:incoming>Sequence_7</bpmn:incoming>
  <bpmn:outgoing>Sequence_8</bpmn:outgoing>
  <bpmn:messageEventDefinition />
</bpmn:IntermediateCatchEvent>

<bpmn:SequenceFlow id="SequenceFlow_8" sourceRef="IntermediateCatchEvent_1" targetRef="Task_5">
<bpmn:task id="Task_5" name="accept orders.">
  <bpmn:incoming>SequenceFlow_8</bpmn:incoming>
  <bpmn:outgoing>SequenceFlow_9</bpmn:outgoing>
</bpmn:task>

<bpmn:boundaryEvent id="BoundaryEvent_1" attachedToRef="Task_5">
  <bpmn:outgoing>SequenceFlow_27</bpmn:outgoing>
  <bpmn:TimerEventDefinition/>
</bpmn:boundaryEvent>

<bpmn:SequenceFlow id="SequenceFlow_27" sourceRef="BoundaryEvent_1" targetRef="IntermediateCatchEvent_1">
<bpmn:SequenceFlow id="SequenceFlow_9" sourceRef="Task_5" targetRef="parallelGateway_1">
<bpmn:parallelGateway id="parallelGateway_1" >
  <bpmn:incoming>SequenceFlow_9</bpmn:incoming>
  <bpmn:outgoing>SequenceFlow_10</bpmn:outgoing>
  <bpmn:outgoing>SequenceFlow_11</bpmn:outgoing>
</bpmn:parallelGateway>
<!-- 10 -->
<bpmn:SequenceFlow id="SequenceFlow_10" sourceRef="parallelGateway_1" targetRef="Task_6">

<bpmn:task id="Task_6" name="record the mileage">
  <bpmn:incoming>SequenceFlow_10</bpmn:incoming>
  <bpmn:outgoing>SequenceFlow_12</bpmn:outgoing>
</bpmn:task>

<bpmn:SequenceFlow id="SequenceFlow_12" sourceRef="Task_6" targetRef="Task_8">
<bpmn:task id="Task_8" name="Settlement with passengers">
  <bpmn:incoming>SequenceFlow_12</bpmn:incoming>
  <bpmn:outgoing>SequenceFlow_15</bpmn:outgoing>
</bpmn:task>

<bpmn:SequenceFlow id="SequenceFlow_15" sourceRef="Task_8" targetRef="parallelGateway_2">

```

```

<bpmn:incoming>SequenceFlow_12</bpmn:incoming>
<bpmn:outgoing>SequenceFlow_15</bpmn:outgoing>
</bpmn:task>

<bpmn:SequenceFlow id="SequenceFlow_15" sourceRef="Task_8" targetRef="parallelGateway_2">
<bpmn:parallelGateway id="parallelGateway_2" >
  <bpmn:incoming>SequenceFlow_15</bpmn:incoming>
  <bpmn:incoming>SequenceFlow_16</bpmn:incoming>
  <bpmn:outgoing>SequenceFlow_17</bpmn:outgoing>
</bpmn:parallelGateway>
<!--11-->
<bpmn:SequenceFlow id="SequenceFlow_11" sourceRef="parallelGateway_1" targetRef="Task_7">
<bpmn:task id="Task_7" name="wait for the passengers and">
  <bpmn:incoming>SequenceFlow_11</bpmn:incoming>
  <bpmn:outgoing>SequenceFlow_13</bpmn:outgoing>
</bpmn:task>

<bpmn:SequenceFlow id="SequenceFlow_13" sourceRef="Task_7" targetRef="Task_9">
<bpmn:task id="Task_9" name="Transport passengers">
  <bpmn:incoming>SequenceFlow_13</bpmn:incoming>
  <bpmn:outgoing>SequenceFlow_14</bpmn:outgoing>
</bpmn:task>
<!--25-->
<bpmn:boundaryEvent id="BoundaryEvent_2" attachedToRef="Task_9">
  <bpmn:outgoing>SequenceFlow_25</bpmn:outgoing>
  <bpmn:errorEventDefinition/>
</bpmn:boundaryEvent>

<bpmn:SequenceFlow id="SequenceFlow_25" sourceRef="BoundaryEvent_2" targetRef="Task_11">
<bpmn:task id="Task_11" name="Contact the company">
  <bpmn:incoming>SequenceFlow_25</bpmn:incoming>
  <bpmn:outgoing>SequenceFlow_26</bpmn:outgoing>
</bpmn:task>

<bpmn:SequenceFlow id="SequenceFlow_26" sourceRef="Task_11" targetRef="endEvent_1">
<!--end-->
<bpmn:endEvent id="endEvent_1">
  <bpmn:incoming>SequenceFlow_22</bpmn:incoming>
  <bpmn:incoming>SequenceFlow_26</bpmn:incoming>
  <bpmn:endEventDefinition />
</bpmn:endEvent>

```

```

<!--14-->
<bpmn:SequenceFlow id="SequenceFlow_14" sourceRef="Task_9" targetRef="Task_10">
<bpmn:task id="Task_10" name="Arrive at one's destination">
<bpmn:incoming>SequenceFlow_14</bpmn:incoming>
<bpmn:outgoing>SequenceFlow_16</bpmn:outgoing>
</bpmn:task>

<bpmn:SequenceFlow id="SequenceFlow_16" sourceRef="Task_10" targetRef="parallelGateway_2">
<bpmn:SequenceFlow id="SequenceFlow_17" sourceRef="parallelGateway_2" targetRef="Task_12">

<bpmn:task id="Task_12" name="Settlement with company">
<bpmn:incoming>SequenceFlow_17</bpmn:incoming>
<bpmn:outgoing>SequenceFlow_18</bpmn:outgoing>
</bpmn:task>

<bpmn:SequenceFlow id="SequenceFlow_18" sourceRef="Task_12" targetRef="exclusiveGateway_2">

<bpmn:exclusiveGateway id="exclusiveGateway_2">
<bpmn:incoming>SequenceFlow_18</bpmn:incoming>
<bpmn:outgoing>SequenceFlow_19</bpmn:outgoing>
<bpmn:outgoing>SequenceFlow_20</bpmn:outgoing>
<bpmn:outgoing>SequenceFlow_21</bpmn:outgoing>
</bpmn:exclusiveGateway>

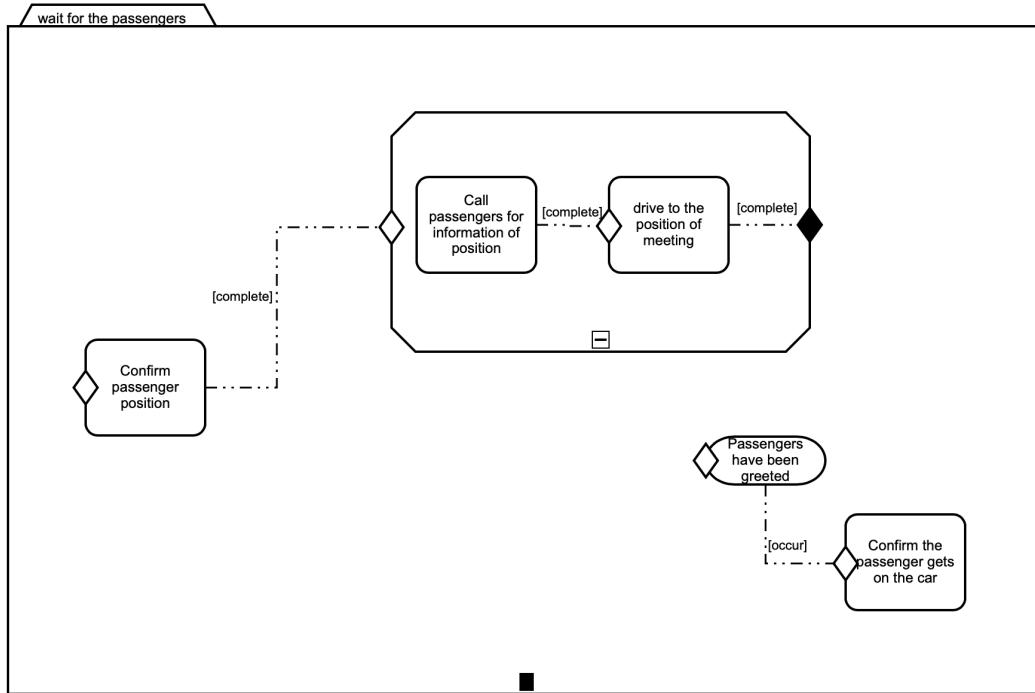
<bpmn:SequenceFlow id="SequenceFlow_19" sourceRef="exclusiveGateway_2" targetRef="Task_13">
<bpmn:SequenceFlow id="SequenceFlow_20" sourceRef="exclusiveGateway_2" targetRef="Task_14">
<bpmn:SequenceFlow id="SequenceFlow_21" sourceRef="exclusiveGateway_2" targetRef="Task_15">

<bpmn:task id="Task_13" name="Continuation of business">
<bpmn:incoming>SequenceFlow_19</bpmn:incoming>
<bpmn:outgoing>SequenceFlow_23</bpmn:outgoing>
</bpmn:task>
<bpmn:task id="Task_14" name="The driver to Take a break">
<bpmn:incoming>SequenceFlow_20</bpmn:incoming>
<bpmn:outgoing>SequenceFlow_24</bpmn:outgoing>
</bpmn:task>
<bpmn:task id="Task_15" name="Closing of business">
<bpmn:incoming>SequenceFlow_21</bpmn:incoming>
<bpmn:outgoing>SequenceFlow_22</bpmn:outgoing>
</bpmn:task>

<bpmn:SequenceFlow id="SequenceFlow_23" sourceRef="Task_13" targetRef="IntermediateCatchEvent_1" />
<bpmn:SequenceFlow id="SequenceFlow_24" sourceRef="Task_14" targetRef="Task_4" />
<bpmn:SequenceFlow id="SequenceFlow_22" sourceRef="Task_15" targetRef="endEvent_1" />
</bpmn:process>

```

CMMN of process of drivers



In the process of customer hailing, the customer should download the application firstly, and create an account. Then, when the client wants a sharing car, the client opens the application, and ask for a car. Maybe queueing is needed. Then the customer gets on the car and arrives at the destination. After the payment, the client leaves. If there are any problems, the client can contact the customer service, and solve the possible problems.

1) Download apps

Clients know about a sharing car application, they would download the app. The clients need to create accounts by using personal information, and set up emergency contacts.

2) Ask for sharing cars

When clients need sharing cars, the clients open the application and ask for cars.

3) Calling accepted or queue

If orders are accepted, clients can see locations of cars, and know details of cars. If callings are not accepted, clients queue and they can know the number of people queued. When orders are accepted, clients can know locations and details. And if orders are not accepted within 10 minutes, callings would be accepted by the nearest cars. Then clients can wait for cars.

4) Drive to destinations

The clients get on sharing cars, and go to destinations.

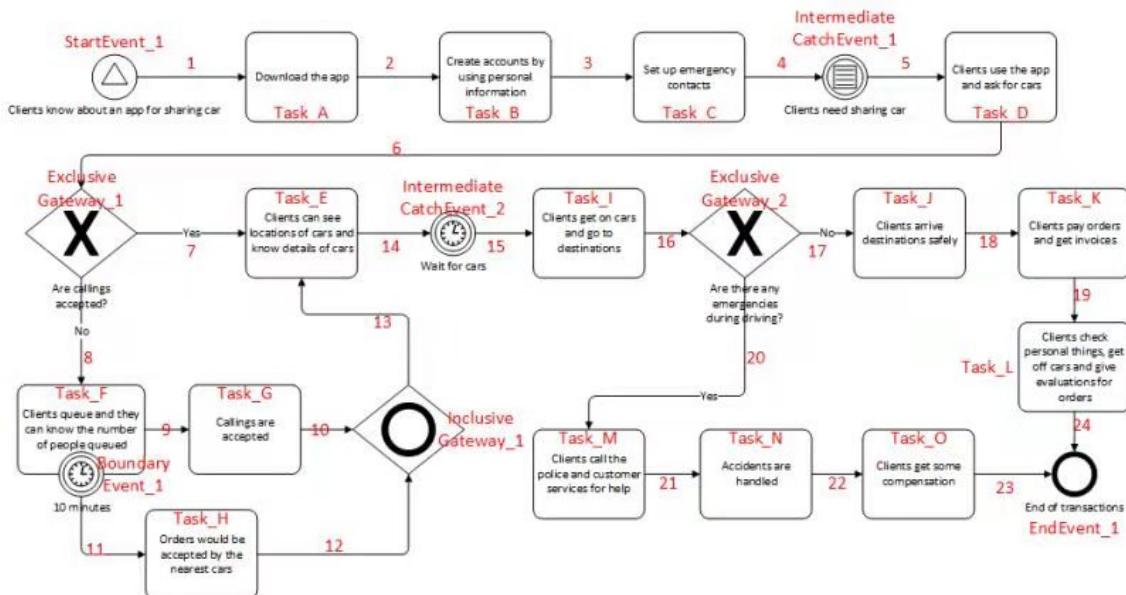
5) Leave the sharing car

When clients are arrived safely, clients pay the money and get invoices. Clients should check their personal things, and get off cars. Clients also can do evaluations for orders. This is the end of transactions.

6) Solve possible problems in orders

If there are any emergencies during driving, clients can contact the police and customer services looking for help. After accidents are handled, clients can get some compensation and this is the end of transactions.

BPMN of process of client



XML of BPMN

```

<bpmn:process id="Process" isExecutable="true">
  <bpmn:task id="Task_A" name="Task_A">
    <bpmn:incoming>SequenceFlow_1</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_2</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_B" name="Task_B">
    <bpmn:incoming>SequenceFlow_2</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_3</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_C" name="Task_C">
    <bpmn:incoming>SequenceFlow_3</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_4</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_D" name="Task_D">
    <bpmn:incoming>SequenceFlow_5</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_6</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_E" name="Task_E">
    <bpmn:incoming>SequenceFlow_7</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_14</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_F" name="Task_F">
    <bpmn:incoming>SequenceFlow_8</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_9</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_G" name="Task_G">
    <bpmn:incoming>SequenceFlow_9</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_10</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_H" name="Task_H">
    <bpmn:incoming>SequenceFlow_11</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_12</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_I" name="Task_I">
    <bpmn:incoming>SequenceFlow_15</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_16</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_J" name="Task_J">
    <bpmn:incoming>SequenceFlow_17</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_18</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_K" name="Task_K">
    <bpmn:incoming>SequenceFlow_18</bpmn:incoming>
    <bpmn:outgoing>SequenceFlow_19</bpmn:outgoing>
  </bpmn:task>

```

```

<bpmn:task id="Task_L" name="Task_L">„
  <bpmn:incoming>SequenceFlow_19</bpmn:incoming>„
  <bpmn:outgoing>SequenceFlow_24</bpmn:outgoing>„
</bpmn:task>„
<bpmn:task id="Task_M" name="Task_M">„
  <bpmn:incoming>SequenceFlow_20</bpmn:incoming>„
  <bpmn:outgoing>SequenceFlow_21</bpmn:outgoing>„
</bpmn:task>„
<bpmn:task id="Task_N" name="Task_N">„
  <bpmn:incoming>SequenceFlow_21</bpmn:incoming>„
  <bpmn:outgoing>SequenceFlow_22</bpmn:outgoing>„
</bpmn:task>„
<bpmn:task id="Task_O" name="Task_O">„
  <bpmn:incoming>SequenceFlow_22</bpmn:incoming>„
  <bpmn:outgoing>SequenceFlow_23</bpmn:outgoing>„
</bpmn:task>„
<bpmn:sequenceFlow id="SequenceFlow_1" sourceRef="StartEvent_1" „
  targetRef="Task_A" />„
<bpmn:sequenceFlow id="SequenceFlow_2" sourceRef="Task_A" „
  targetRef="Task_B" />„
<bpmn:sequenceFlow id="SequenceFlow_3" sourceRef="Task_B" „
  targetRef="Task_C" />„
<bpmn:sequenceFlow id="SequenceFlow_4" sourceRef="Task_C" „
  targetRef="IntermediateCatchEvent_1" />„
<bpmn:sequenceFlow id="SequenceFlow_5" sourceRef="IntermediateCatchEvent_1" „
  targetRef="Task_D" />„
<bpmn:sequenceFlow id="SequenceFlow_6" sourceRef="Task_D" „
  targetRef="ExclusiveGateway_1" />„
<bpmn:sequenceFlow id="SequenceFlow_7" sourceRef="ExclusiveGateway_1" „
  targetRef="Task_E" />„
  <bpmn:conditionExpression>„
    xsi:type="bpmn:tFormalExpression">Yes</bpmn:conditionExpression>„
  </bpmn:sequenceFlow>„
<bpmn:sequenceFlow id="SequenceFlow_8" sourceRef="ExclusiveGateway_1" „
  targetRef="Task_E" />„
  <bpmn:conditionExpression>„
    xsi:type="bpmn:tFormalExpression">No</bpmn:conditionExpression>„
  </bpmn:sequenceFlow>„
<bpmn:sequenceFlow id="SequenceFlow_9" sourceRef="Task_E" „
  targetRef="Task_G" />„
<bpmn:sequenceFlow id="SequenceFlow_10" sourceRef="Task_G" „
  targetRef="InclusiveGateway_1" />„
<bpmn:sequenceFlow id="SequenceFlow_11" sourceRef="BoundaryEvent_1" „
  targetRef="Task_H" />„

```

```

<bpmn:sequenceFlow id="SequenceFlow_12" sourceRef="Task_H">
  targetRef="InclusiveGateway_1" />
<bpmn:sequenceFlow id="SequenceFlow_13" sourceRef="InclusiveGateway_1">
  targetRef="Task_E" />
<bpmn:sequenceFlow id="SequenceFlow_14" sourceRef="Task_E">
  targetRef="IntermediateCatchEvent_2" />
<bpmn:sequenceFlow id="SequenceFlow_15" sourceRef="IntermediateCatchEvent_2">
  targetRef="Task_I" />
<bpmn:sequenceFlow id="SequenceFlow_16" sourceRef="Task_K">
  targetRef="ExclusiveGateway_2" />
<bpmn:sequenceFlow id="SequenceFlow_17" sourceRef="ExclusiveGateway_2">
  targetRef="Task_J" />
  <bpmn:conditionExpression>
    xsi:type="bpmn:tFormalExpression">No</bpmn:conditionExpression>
  </bpmn:sequenceFlow>
<bpmn:sequenceFlow id="SequenceFlow_18" sourceRef="Task_J">
  targetRef="Task_K" />
<bpmn:sequenceFlow id="SequenceFlow_19" sourceRef="Task_K">
  targetRef="Task_L" />
<bpmn:sequenceFlow id="SequenceFlow_20" sourceRef="ExclusiveGateway_2">
  targetRef="Task_M" />
  <bpmn:conditionExpression>
    xsi:type="bpmn:tFormalExpression">Yes</bpmn:conditionExpression>
  </bpmn:sequenceFlow>
<bpmn:sequenceFlow id="SequenceFlow_21" sourceRef="Task_M">
  targetRef="Task_N" />
<bpmn:sequenceFlow id="SequenceFlow_22" sourceRef="Task_N">
  targetRef="Task_O" />
<bpmn:sequenceFlow id="SequenceFlow_23" sourceRef="Task_O">
  targetRef="EndEvent_1" />
<bpmn:sequenceFlow id="SequenceFlow_24" sourceRef="Task_L">
  targetRef="EndEvent_1" />
<bpmn:startEvent id="StartEvent_1">
  <bpmn:outgoing>SequenceFlow_1</bpmn:outgoing>
  <bpmn:signalEventDefinition />
</bpmn:startEvent>
<bpmn:intermediateCatchEvent id="IntermediateCatchEvent_1">
  <bpmn:incoming>SequenceFlow_4</bpmn:incoming>
  <bpmn:outgoing>SequenceFlow_5</bpmn:outgoing>
  <bpmn:conditionalEventDefinition />
</bpmn:intermediateCatchEvent>
<bpmn:intermediateCatchEvent id="IntermediateCatchEvent_2">
  <bpmn:incoming>SequenceFlow_14</bpmn:incoming>
  <bpmn:outgoing>SequenceFlow_15</bpmn:outgoing>

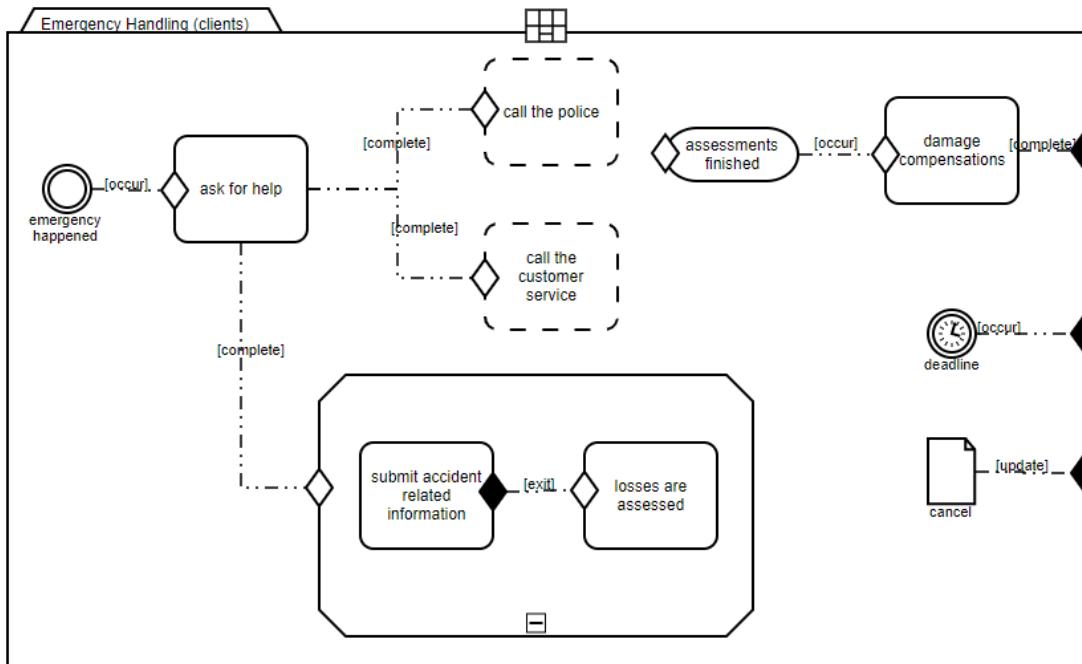
```

```

<bpmn:timerEventDefinition />
</bpmn:intermediateCatchEvent>
<bpmn:boundaryEvent id="BoundaryEvent_1" attachedToRef="Task_F">
  <bpmn:outgoing>SequenceFlow_11</bpmn:outgoing>
  <bpmn:timerEventDefinition />
</bpmn:boundaryEvent>
<bpmn:endEvent id="EndEvent_1">
  <bpmn:incoming>SequenceFlow_23</bpmn:incoming>
  <bpmn:incoming>SequenceFlow_24</bpmn:incoming>
</bpmn:endEvent>
<bpmn:exclusiveGateway id="ExclusiveGateway_1">
  <bpmn:incoming> SequenceFlow_6</bpmn:incoming>
  <bpmn:outgoing> SequenceFlow_7</bpmn:outgoing>
  <bpmn:outgoing> SequenceFlow_8</bpmn:outgoing>
</bpmn:exclusiveGateway>
<bpmn:inclusiveGateway id="InclusiveGateway_1">
  <bpmn:incoming> SequenceFlow_10</bpmn:incoming>
  <bpmn:incoming> SequenceFlow_12</bpmn:incoming>
  <bpmn:outgoing> SequenceFlow_13</bpmn:outgoing>
</bpmn:inclusiveGateway>
<bpmn:exclusiveGateway id="ExclusiveGateway_2">
  <bpmn:incoming> SequenceFlow_16</bpmn:incoming>
  <bpmn:outgoing> SequenceFlow_17</bpmn:outgoing>
  <bpmn:outgoing> SequenceFlow_20</bpmn:outgoing>
</bpmn:exclusiveGateway>
</bpmn:process>

```

### CMMN of process of drivers



10 In the process of the allotment of the order, the system of this company will allot the order to the car. Firstly, the system will judge the types of orders, which may be private cars and taxies. Secondly, the system will allot the order according to the types of orders. If it is a private car order, the order will be sent to the nearest nine drivers, or if it is a taxi order, the order will be sent to the drivers in 1.5 kilometres. After the order is finished, the system will get money from the client and pay money to the driver. If the client has any problems with this order, the client can call the company. The type of order: Immediately call a car, Book a car, Freeride.

a) Immediately call a car

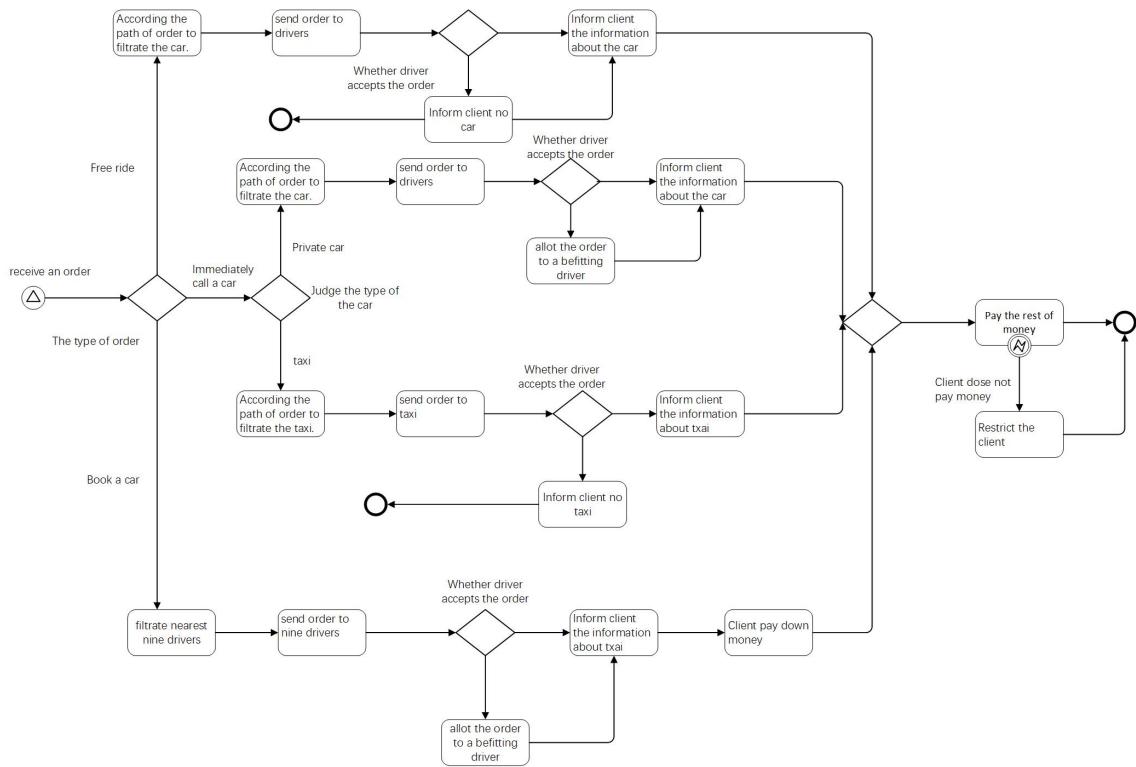
Private car: Receive an order, filtrate the nearest nine drivers, send the order to nine drivers. If no driver accepts the order, the system will allot the order to a benefit driver. When the order has accomplished, the system will get money from the client and pay money to the driver. If the client didn't pay money to the system, the client will not be able to use this system until he pays money. Taxi: Receive an order, filtrate taxis that the distance between client and taxi is less than 1.5km. Send order to these taxis. If no taxi accepts the order in ten minutes, the system will inform the client to change the style of the car. When the order has accomplished, the system will get money from the client and pay money to the driver. If the client didn't pay money to the system, the client will not be able to use this system until he pays money. Exception handling: If no car can meet the order, the system will inform the client the error message. If the client cancels the order and a driver or a taxi has accepted more than two minutes, the client should pay Liquidated damages.

b) Book a car

Receive an order. Issue the order to drivers that the distance between client and driver is less than 10km. If no driver accepts the order, the system will allot the order to a benefit driver. The client pays down payment to the system. When the order has accomplished, the system will get money from the client and pay money to the driver. If the client didn't pay money to the system, the client will not be able to use this system until he pays money.

c) Freeride

Receive an order. According to the path of order to filtrate the car. Issue the order to drivers that satisfy the path. If no driver accepts the order, the system will inform the client no car. When the order has accomplished, the system will get money from the client and pay money to the driver. If the client didn't pay money to the system, the client will not be able to use this system until he pays money.



XML of BPMN

```
<bpmn:process id="Process_order" isExecutable="true">
  <bpmn:task id="Task_1" name="Task_1">
    <bpmn:incoming>F2</bpmn:incoming>
    <bpmn:outgoing>F3</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_2" name="Task_2">
    <bpmn:incoming>F3</bpmn:incoming>
    <bpmn:outgoing>F4</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_3" name="Task_3">
    <bpmn:incoming>F6</bpmn:incoming>
    <bpmn:outgoing>F8</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_4" name="Task_4">
    <bpmn:incoming>F5</bpmn:incoming>
    <bpmn:outgoing>F9</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_5" name="Task_5">
    <bpmn:incoming>F11</bpmn:incoming>
    <bpmn:outgoing>F12</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_6" name="Task_6">
    <bpmn:incoming>F12</bpmn:incoming>
    <bpmn:outgoing>F13</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_7" name="Task_7">
    <bpmn:incoming>F14</bpmn:incoming>
    <bpmn:outgoing>F17</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_8" name="Task_8">
    <bpmn:incoming>F15</bpmn:incoming>
    <bpmn:outgoing>F16</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_9" name="Task_9">
    <bpmn:incoming>F18</bpmn:incoming>
    <bpmn:outgoing>F19</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_10" name="Task_10">
    <bpmn:incoming>F19</bpmn:incoming>
    <bpmn:outgoing>F20</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_11" name="Task_11">
    <bpmn:incoming>F23</bpmn:incoming>
    <bpmn:outgoing>F24</bpmn:outgoing>
  </bpmn:task>
  <bpmn:task id="Task_12" name="Task_12">
    <bpmn:incoming>F22</bpmn:incoming>
    <bpmn:outgoing>F22</bpmn:outgoing>
```

```
</bpmn:task>
<bpmn:task id="Task_13" name="Task_13">
    <bpmn:incoming>F25</bpmn:incoming>
    <bpmn:outgoing>F26</bpmn:outgoing>
</bpmn:task>
<bpmn:task id="Task_14" name="Task_14">
    <bpmn:incoming>F26</bpmn:incoming>
    <bpmn:outgoing>F27</bpmn:outgoing>
</bpmn:task>
<bpmn:task id="Task_15" name="Task_15">
    <bpmn:incoming>F28</bpmn:incoming>
    <bpmn:outgoing>F30</bpmn:outgoing>
</bpmn:task>
<bpmn:task id="Task_16" name="Task_16">
    <bpmn:incoming>F29</bpmn:incoming>
    <bpmn:outgoing>F30</bpmn:outgoing>
</bpmn:task>
<bpmn:task id="Task_17" name="Task_17">
    <bpmn:incoming>F31</bpmn:incoming>
    <bpmn:outgoing>F32</bpmn:outgoing>
</bpmn:task>
<bpmn:task id="Task_18" name="Task_18">
    <bpmn:incoming>F34</bpmn:incoming>
    <bpmn:outgoing>F35</bpmn:outgoing>
</bpmn:task>
<bpmn:task id="Task_19" name="Task_19">
    <bpmn:incoming>F33</bpmn:incoming>
    <bpmn:outgoing>F36</bpmn:outgoing>
</bpmn:task>
<bpmn:sequenceFlow id="F1" sourceRef="E1" targetRef="G1"/>
<bpmn:sequenceFlow id="F2" sourceRef="G1" targetRef="Task_1"/>
<bpmn:sequenceFlow id="F3" sourceRef="Task_1" targetRef="Task_2"/>
<bpmn:sequenceFlow id="F4" sourceRef="Task_2" targetRef="G2"/>
<bpmn:sequenceFlow id="F5" sourceRef="G2" targetRef="Task_4"/>
<bpmn:sequenceFlow id="F6" sourceRef="G2" targetRef="Task_3"/>
<bpmn:sequenceFlow id="F8" sourceRef="Task_3" targetRef="G7"/>
<bpmn:sequenceFlow id="F9" sourceRef="Task_4" targetRef="E2"/>
<bpmn:sequenceFlow id="F10" sourceRef="G1" targetRef="G3"/>
<bpmn:sequenceFlow id="F11" sourceRef="G3" targetRef="Task_5"/>
<bpmn:sequenceFlow id="F12" sourceRef="Task_5" targetRef="Task_6"/>
<bpmn:sequenceFlow id="F13" sourceRef="Task_6" targetRef="G4"/>
<bpmn:sequenceFlow id="F14" sourceRef="G4" targetRef="Task_7"/>
<bpmn:sequenceFlow id="F15" sourceRef="G4" targetRef="Task_7"/>
<bpmn:sequenceFlow id="F16" sourceRef="Task_8" targetRef="Task_7"/>
<bpmn:sequenceFlow id="F17" sourceRef="Task_7" targetRef="G7"/>
<bpmn:sequenceFlow id="F18" sourceRef="G3" targetRef="Task_9"/>
<bpmn:sequenceFlow id="F19" sourceRef="Task_9" targetRef="Task_10"/>
<bpmn:sequenceFlow id="F20" sourceRef="Task_10" targetRef="G5"/>
```

```

<bpmn:sequenceFlow id="F21" sourceRef="G5" targetRef="Task_12"/>
<bpmn:sequenceFlow id="F22" sourceRef="Task_12" targetRef="E3"/>
<bpmn:sequenceFlow id="F23" sourceRef="G5" targetRef="Task_11"/>
<bpmn:sequenceFlow id="F24" sourceRef="Task_11" targetRef="G7"/>
<bpmn:sequenceFlow id="F25" sourceRef="G1" targetRef="Task_13"/>
<bpmn:sequenceFlow id="F26" sourceRef="Task_13" targetRef="Task_14"/>
<bpmn:sequenceFlow id="F27" sourceRef="Task_14" targetRef="G6"/>
<bpmn:sequenceFlow id="F28" sourceRef="G6" targetRef="Task_15"/>
<bpmn:sequenceFlow id="F29" sourceRef="G6" targetRef="Task_16"/>
<bpmn:sequenceFlow id="F30" sourceRef="Task_15" targetRef="Task_16"/>
<bpmn:sequenceFlow id="F31" sourceRef="Task_16" targetRef="Task_17"/>
<bpmn:sequenceFlow id="F32" sourceRef="Task_17" targetRef="G7"/>
<bpmn:sequenceFlow id="F33" sourceRef="G7" targetRef="Task_19"/>
<bpmn:sequenceFlow id="F34" sourceRef="boundaryEvent_1" targetRef="Task_18"/>
<bpmn:sequenceFlow id="F35" sourceRef="Task_18" targetRef="E4"/>
<bpmn:sequenceFlow id="F36" sourceRef="Task_19" targetRef="E4"/>
<bpmn:startEvent id="E1">
    <bpmn:outgoing></bpmn:outgoing>
    <bpmn:signalEventDefinition/>
</bpmn:startEvent>
<bpmn:boundaryEvent id="boundaryEvent_1">
    <bpmn:outgoing>F34</bpmn:outgoing>
    <bpmn:errorEventDefinition/>
</bpmn:boundaryEvent>
<bpmn:endEvent id="E2">
    <bpmn:outgoing>F9</bpmn:outgoing>
    <bpmn:endEventDefinition/>
</bpmn:endEvent>
<bpmn:endEvent id="E3">
    <bpmn:outgoing>F22</bpmn:outgoing>
    <bpmn:endEventDefinition/>
</bpmn:endEvent>
<bpmn:endEvent id="E4">
    <bpmn:outgoing>F35</bpmn:outgoing>
    <bpmn:endEventDefinition/>
</bpmn:endEvent>
<bpmn:exclusiveGateway id="G1">
    <bpmn:incoming>F1</bpmn:incoming>
    <bpmn:outgoing>F2</bpmn:outgoing>
    <bpmn:outgoing>F10</bpmn:outgoing>
    <bpmn:outgoing>F25</bpmn:outgoing>
</bpmn:exclusiveGateway>
<bpmn:exclusiveGateway id="G2">
    <bpmn:incoming>F4</bpmn:incoming>
    <bpmn:outgoing>F5</bpmn:outgoing>
    <bpmn:outgoing>F6</bpmn:outgoing>
</bpmn:exclusiveGateway>
<bpmn:exclusiveGateway id="G3">

```

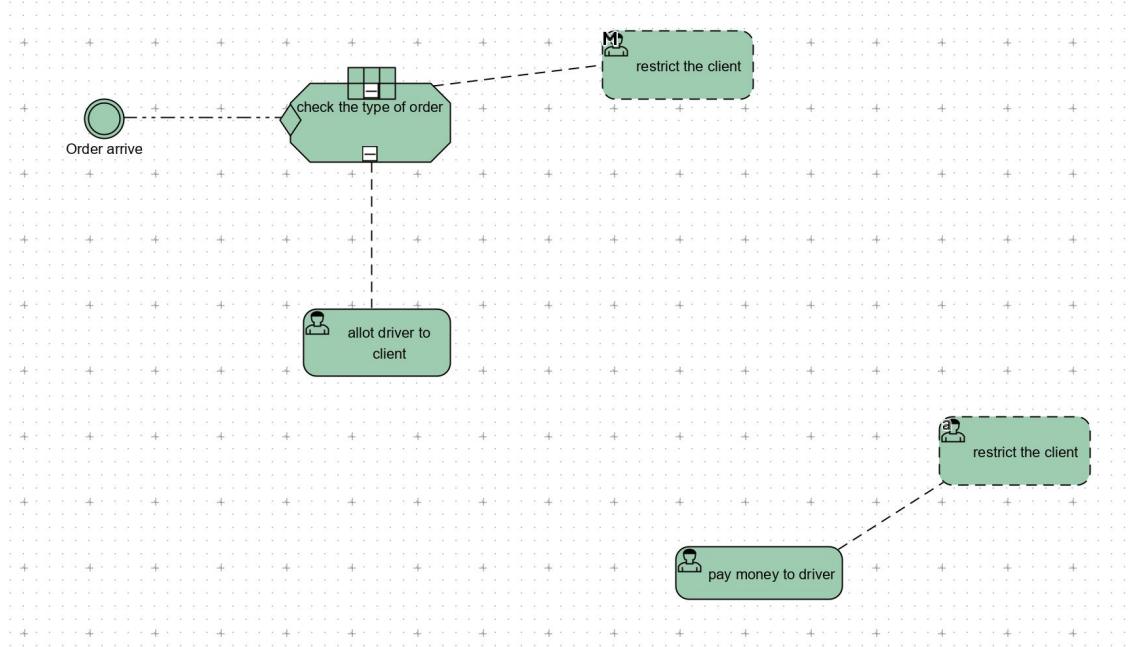
---

```

</bpmn:endEvent>
<bpmn:endEvent id="E3">
    <bpmn:outgoing>F22</bpmn:outgoing>
    <bpmn:endEventDefinition/>
</bpmn:endEvent>
<bpmn:endEvent id="E4">
    <bpmn:outgoing>F35</bpmn:outgoing>
    <bpmn:endEventDefinition/>
</bpmn:endEvent>
<bpmn:exclusiveGateway id="G1">
    <bpmn:incoming>F1</bpmn:incoming>
    <bpmn:outgoing>F2</bpmn:outgoing>
    <bpmn:outgoing>F10</bpmn:outgoing>
    <bpmn:outgoing>F25</bpmn:outgoing>
</bpmn:exclusiveGateway>
<bpmn:exclusiveGateway id="G2">
    <bpmn:incoming>F4</bpmn:incoming>
    <bpmn:outgoing>F5</bpmn:outgoing>
    <bpmn:outgoing>F6</bpmn:outgoing>
</bpmn:exclusiveGateway>
<bpmn:exclusiveGateway id="G3">
    <bpmn:incoming>F10</bpmn:incoming>
    <bpmn:outgoing>F11</bpmn:outgoing>
    <bpmn:outgoing>F18</bpmn:outgoing>
</bpmn:exclusiveGateway>
<bpmn:exclusiveGateway id="G4">
    <bpmn:incoming>F13</bpmn:incoming>
    <bpmn:outgoing>F14</bpmn:outgoing>
    <bpmn:outgoing>F15</bpmn:outgoing>
</bpmn:exclusiveGateway>
<bpmn:exclusiveGateway id="G5">
    <bpmn:incoming>F20</bpmn:incoming>
    <bpmn:outgoing>F21</bpmn:outgoing>
    <bpmn:outgoing>F23</bpmn:outgoing>
</bpmn:exclusiveGateway>
<bpmn:exclusiveGateway id="G6">
    <bpmn:incoming>F27</bpmn:incoming>
    <bpmn:outgoing>F28</bpmn:outgoing>
    <bpmn:outgoing>F29</bpmn:outgoing>
</bpmn:exclusiveGateway>
<bpmn:exclusiveGateway id="G7">
    <bpmn:incoming>F8</bpmn:incoming>
    <bpmn:incoming>F17</bpmn:incoming>
    <bpmn:incoming>F24</bpmn:incoming>
    <bpmn:incoming>F32</bpmn:incoming>
    <bpmn:outgoing>F33</bpmn:outgoing>
</bpmn:exclusiveGateway>
</bpmn:process>

```

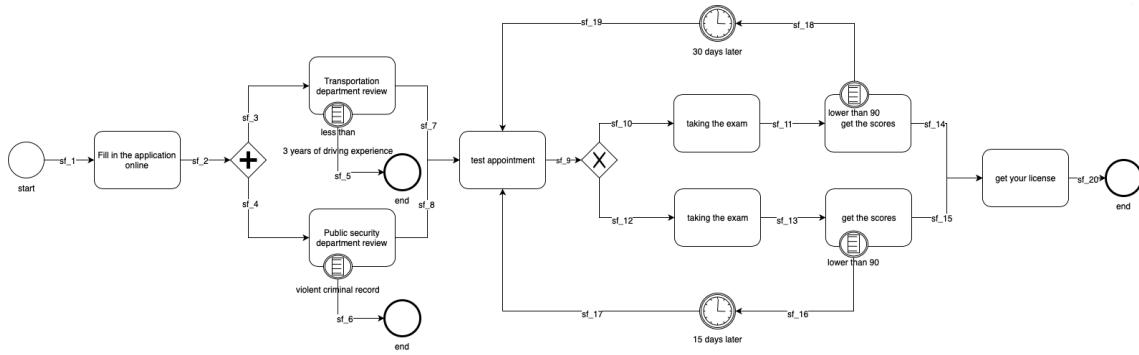
---



This project describes an application for online car-hailing services which needs four part. The part which a driver's process is as follow. When a driver wants to start a business, it is important to register and review qualification. Until the driver passes the audit before receiving the order, the driver starts online and then waits for message about an order. When the driver receives the order, the driver could choose to confirm or cancel. Driver needs to connect with the passenger and waits for the passenger if confirmed. When the passenger gets on the car, driver starts to record the mileage. If there is a traffic accident, the system needs to throw an exception. Under normal circumstances, the car reaches its destination without any trouble, then the order information is submitted to the company for settlement. Business can be suspended or closed if there are no orders in progress. Specie process:

- The driver logs into the system and needs to verify that he is a registered user. If the driver has been registered, he can enter the business. If there is no registration, he needs to submit information such as driver's license, identity information and vehicle information for registration, and then submit the information for review.
- After logging in the system, confirm the start of business, can accept orders. After receiving the order, if the driver does not confirm, the default wait 20 seconds to cancel. After confirming the order, wait for the passengers and contact with the passengers. After receiving the passengers, confirm the passengers get on the bus and record the mileage by meter.
- Abnormal handling and settlement of driving process. In case of accident, contact company, hospital, insurance company, etc. After successfully reaching the destination, the formation information will be sent to the company for settlement. Then the next order can be placed.
- Drivers can choose to enter the rest stage and off duty stage; the company will not send orders to the driver.

BPMN of process of qualification view:



XML of BPMN

```

1 <bpmn:process id="Process_1" isExecutable="true">
2   <bpmn:task id="Task_1" name="Task A">
3     <bpmn:incoming>SequenceFlow_1</bpmn:incoming>
4     <bpmn:outgoing>SequenceFlow_2</bpmn:outgoing>
5   </bpmn:task>
6   <bpmn:sequenceFlow id="SequenceFlow_1" sourceRef="StartEvent_1" targetRef="Task_1"/>
7   <bpmn:exclusiveGateway id="ExclusiveGateway_1">
8     <bpmn:incoming>SequenceFlow_2</bpmn:incoming>
9     <bpmn:outgoing>SequenceFlow_3</bpmn:outgoing>
10    <bpmn:outgoing>SequenceFlow_4</bpmn:outgoing>
11  </bpmn:exclusiveGateway>
12  <bpmn:sequenceFlow id="SequenceFlow_2" sourceRef="Task_1" targetRef="ExclusiveGateway_1"/>
13  <bpmn:incoming>SequenceFlow_3</bpmn:incoming>
14  <bpmn:task id="Task_2" name="Task B">
15    <bpmn:incoming>SequenceFlow_3</bpmn:incoming>
16    <bpmn:outcoming>SequenceFlow_7</bpmn:outcoming>
17  </bpmn:task>
18  <bpmn:task id="Task_3" name="Task C">
19    <bpmn:incoming>SequenceFlow_4</bpmn:incoming>
20    <bpmn:outcoming>SequenceFlow_8</bpmn:outcoming>
21  </bpmn:task>
22  <bpmn:sequenceFlow id="SequenceFlow_3" sourceRef="ExclusiveGateway_1" targetRef="Task_2"/>

```

```

45      <bpmn:outgoing>SequenceFlow_10</bpmn:outgoing>
46      <bpmn:outgoing>SequenceFlow_12</bpmn:outgoing>
47    </bpmn:exclusiveGateway>
48    <bpmn:task id="Task_5" name="Task E">
49      <bpmn:incoming>SequenceFlow_10</bpmn:incoming>
50      <bpmn:outgoing>SequenceFlow_12</bpmn:outgoing>
51    </bpmn:task>
52    <bpmn:task id="Task_6" name="Task F">
53      <bpmn:incoming>SequenceFlow_11</bpmn:incoming>
54      <bpmn:outgoing>SequenceFlow_14</bpmn:outgoing>
55    </bpmn:task>
56    <bpmn:task id="Task_7" name="Task G">
57      <bpmn:incoming>SequenceFlow_12</bpmn:incoming>
58      <bpmn:outgoing>SequenceFlow_15</bpmn:outgoing>
59    </bpmn:task>
60    <bpmn:task id="Task_8" name="Task H">
61      <bpmn:incoming>SequenceFlow_13</bpmn:incoming>
62      <bpmn:outgoing>SequenceFlow_15</bpmn:outgoing>
63    </bpmn:task>
64    <bpmn:boundaryEvent id="IntermediateCondition_3">
65      <bpmn:outcoming>Sequenceflow_16</bpmn:outcoming>
66    </bpmn:boundaryEvent>

```

```

67    <bpmn:intermediateCatchEvent id="IntermediateThrowEvent_1" name="15 days later">
68      <bpmn:incoming>Sequenceflow_16</bpmn:incoming>
69      <bpmn:outgoing>Sequenceflow_17</bpmn:outgoing>
70      <bpmn:timerEventDefinition />
71    </bpmn:intermediateCatchEvent>
72    <bpmn:boundaryEvent id="IntermediateCondition_4">
73      <bpmn:outcoming>Sequenceflow_18</bpmn:outcoming>
74    </bpmn:boundaryEvent>
75    <bpmn:intermediateCatchEvent id="IntermediateThrowEvent_2" name="30 days later">
76      <bpmn:incoming>Sequenceflow_18</bpmn:incoming>
77      <bpmn:outgoing>Sequenceflow_19</bpmn:outgoing>
78      <bpmn:timerEventDefinition />
79    </bpmn:intermediateCatchEvent>
80    <bpmn:task id="Task_9" name="Task I">
81      <bpmn:incoming>Sequenceflow_14</bpmn:incoming>
82      <bpmn:incoming>Sequenceflow_15</bpmn:incoming>
83      <bpmn:outgoing>Sequenceflow_20</bpmn:outgoing>
84    </bpmn:task>
85    <bpmn:endEvent id="EndEvent_3">
86      <bpmn:incoming>Sequenceflow_20</bpmn:incoming>
87    </bpmn:endEvent>
88  </bpmn:process>

```

```

22   <bpmn:sequenceFlow id="SequenceFlow_3" sourceRef="ExclusiveGateway_1" targetRef="Task_2"/>
23   <bpmn:sequenceFlow id="SequenceFlow_4" sourceRef="ExclusiveGateway_1" targetRef="Task_3"/>
24   <bpmn:boundaryEvent id="intermediateCondition_1">
25     <bpmn:outgoing>Sequenceflow_5</bpmn:outgoing>
26   </boundaryEvent>
27   <bpmn:endEvent id="EndEvent_1">
28     <bpmn:incoming>Sequenceflow_5</bpmn:incoming>
29   </bpmn:endEvent>
30   <bpmn:boundaryEvent id="intermediateCondition_2">
31     <bpmn:outgoing>Sequenceflow_6</bpmn:outgoing>
32   </boundaryEvent>
33   <bpmn:endEvent id="EndEvent_2">
34     <bpmn:incoming>Sequenceflow_6</bpmn:incoming>
35   </bpmn:endEvent>
36   <bpmn:task id="Task_4" name="Task_D">
37     <bpmn:incoming>Sequenceflow_7</bpmn:incoming>
38     <bpmn:incoming>Sequenceflow_8</bpmn:incoming>
39     <bpmn:incoming>Sequenceflow_17</bpmn:incoming>
40     <bpmn:incoming>Sequenceflow_19</bpmn:incoming>
41     <bpmn:outgoing>Sequenceflow_9</bpmn:outgoing>
42   </bpmn:task>
43   <bpmn:exclusiveGateway id="ExclusiveGateway_2">
44     <bpmn:incoming>Sequenceflow_9</bpmn:incoming>

```

CMMN of process of qualification view

