

CSCI927 Service-Oriented Software Engineering (Project Report)

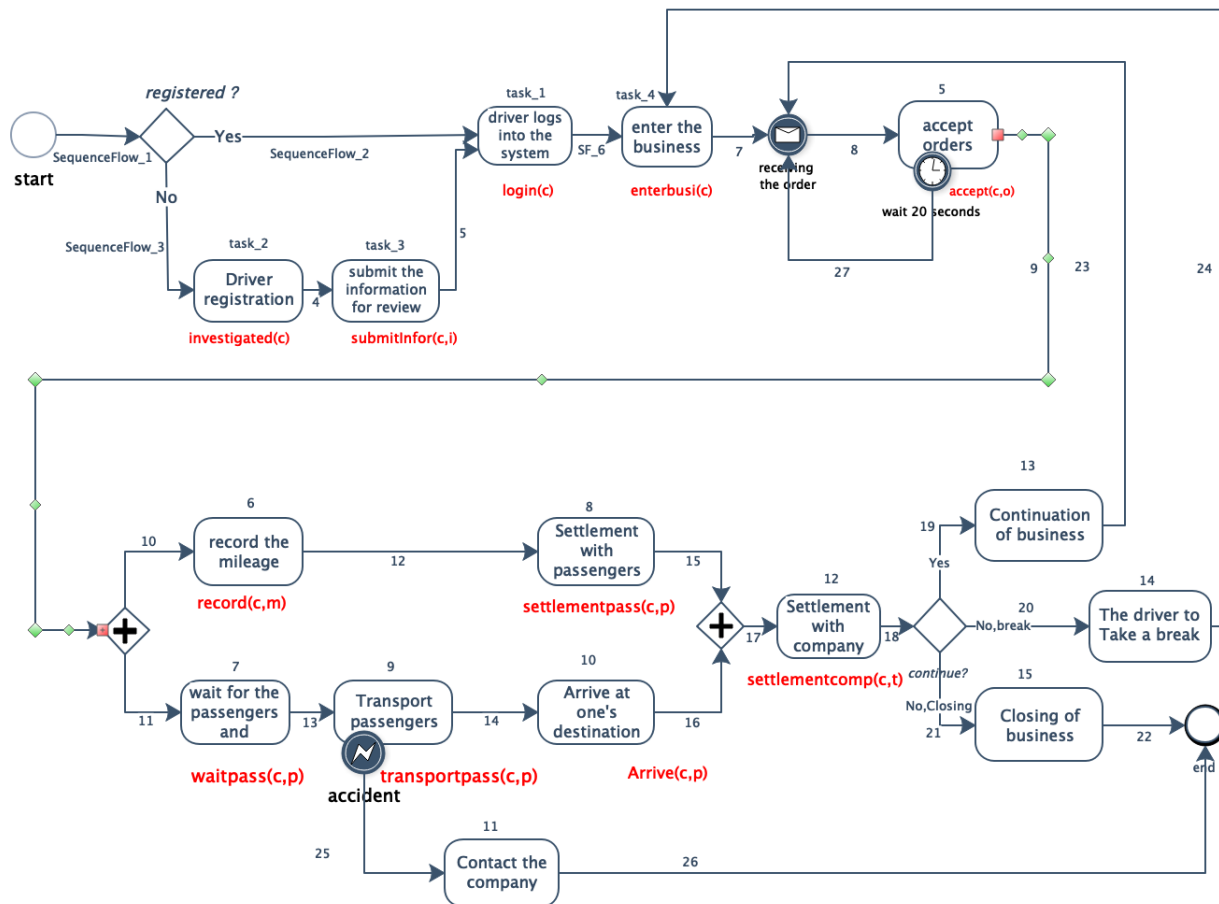
Project Title

Group Members (Group number): Name1, Name2, Name3, Name4

1 driver bpmn

This project describes an application for online car-hailing services which need 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. Waiting for the driver to pass the audit before receiving the order. The driver starts online and then waiting for message about a order. When the driver receives the order, The driver could choose to Confirm or cancel. Driver need to connect with the passenger and waiting for the passenger if confirmed. When the passenger gets on the car. Driver start to record the mileage. If there is a traffic accident, the system need to throw an exception. Under normal circumstances the car reached its destination without any trouble, then the order information is submitted to the company for settlement. Drivers can suspend or close business without receiving orders.

This BPMN describes the driver's process. The first part is qualification certification and the second part is order confirmation. The third part is that the driver picks up the passenger after confirming the order and records the mileage. After finishing this order, the driver can choose whether to continue the business or not. In case of any accident in the delivery process, the driver needs to contact the company to deal with it.



picture 1. driver bpmn

2 bpmn xml code

```

1 <bpmn:process id="Process_1" isExecutable="true">
2   <bpmn:startEvent id="startEvent_1">
3     <bpmn:outgoing>SequenceFlow_1</bpmn:outgoing>
4     <bpmn:conditionalStartDefinition />
5   </bpmn:startEvent>
6
7   <bpmn:SequenceFlow id="SequenceFlow_1" sourceRef="StartEvet_1"
8     targetRef="exclusiveGateway_1">
9
10    <bpmn:exclusiveGateway id="exclusiveGateway_1" name="
11      regisitered?">
12      <bpmn:incoming>SequenceFlow_1</bpmn:incoming>
13      <bpmn:outgoing>SequenceFlow_2</bpmn:outgoing>
14      <bpmn:outgoing>SequenceFlow_3</bpmn:outgoing>
15    </bpmn:exclusiveGateway>
16
17    <bpmn:SequenceFlow id="SequenceFlow_2" sourceRef="
18      exclusiveGateway_1" targetRef="Task_1">

```

```

16      <bpmn:SequenceFlow id="SequenceFlow_3" sourceRef="
17          exclusiveGateway_1" targetRef="Task_2">
18
19      ...
20      ...
21
22      ...
23
24
25
26      <bpmn:task id="Task_14" name="The_driver_to_Take_a_break">
27          <bpmn:incoming>SequenceFlow_20</bpmn:incoming>
28          <bpmn:outgoing>SequenceFlow_24</bpmn:outgoing>
29      </bpmn:task>
30      <bpmn:task id="Task_15" name="Closing_of_business">
31          <bpmn:incoming>SequenceFlow_21</bpmn:incoming>
32          <bpmn:outgoing>SequenceFlow_22</bpmn:outgoing>
33      </bpmn:task>
34
35      <bpmn:SequenceFlow id="SequenceFlow_23" sourceRef="Task_13"
36          targetRef="IntermediateCatchEvent_1" />
37      <bpmn:SequenceFlow id="SequenceFlow_24" sourceRef="Task_14"
38          targetRef="Task_4" />
39      <bpmn:SequenceFlow id="SequenceFlow_22" sourceRef="Task_15"
40          targetRef="endEvent_1" />
41  </bpmn:process>

```

3 bpmn effect scenarios:

t13: Scenario(1): ((t1,t4,t5,{ <t6,t8>,<t7,t9,t10> },t12,t13), { <t2, t3,> })
 t13: Secnario(2): ((t2,t3,t1,{<t6,t8>,<t7,t9,t10> }},t12,t13))
 t14: Secnario(1): ((t1,t4,t5,{<t6,t8>,<t7,t9,t10> }},t12,t14), { <t2, t3,> })
 t14: Secnario(2): ((t2,t3,t1,{<t6,t8>,<t7,t9,t10> }},t12,t14))
 t15: Secnario(1): ((t1,t4,t5,{<t6,t8>,<t7,t9,t10> } },t12,t15), { <t2, t3,> })
 t15: Secnario(2): ((t2,t3,t1,{<t6,t8>,<t7,t9,t10> } },t12,t15))

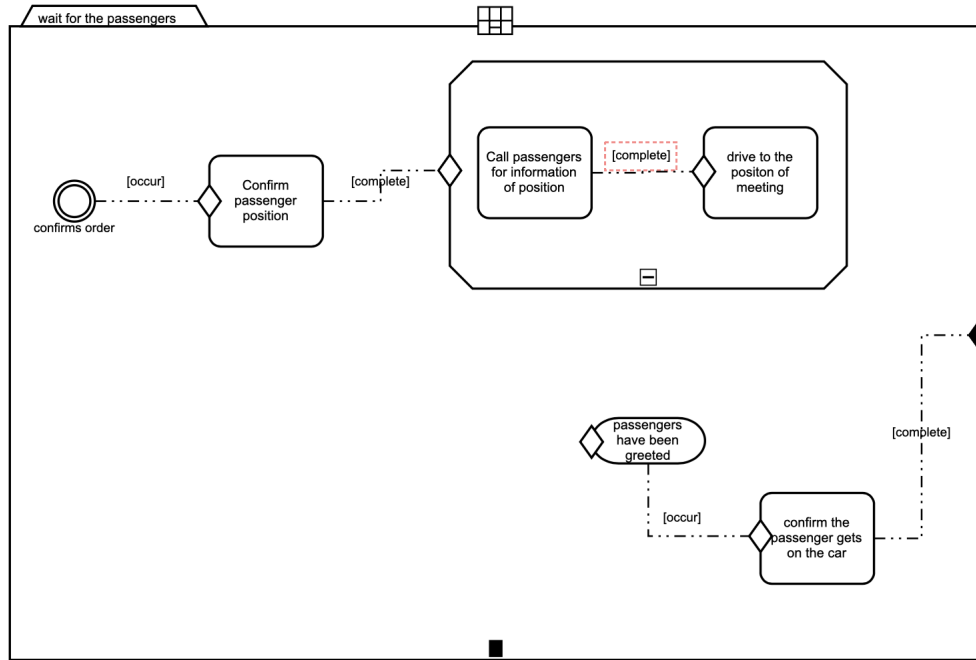
4 bpmn cumulative effects:

enter business: (login(c) \vee investigated(c) \wedge submitInfor(c,i)) \wedge enterbusi(c)
 accept order: (login(c) \vee investigated(c) \wedge submitInfor(c,i)) \wedge enterbusi(c) \wedge accept(c,o)
 settlement with company: (login(c) \vee investigated(c) \wedge submitInfor(c,i)) \wedge enterbusi(c) \wedge accept(c,o) \wedge ((record(c,m) \wedge settlementpass(c,p)) \vee (waitpass(c,p)) \wedge transportpass(c,p) \wedge Arrive(c,p))
 \wedge settlementcomp(c,t)

5 cmmn

Subprocess of driver bpmn: wait for the passengers: First, the driver needs to confirm passengers' position, there is a cmmn stage to describe the process. It is necessary to connect with passengers

by phone to get information about the position, then the driver drives to the position. It is a common milestone when meeting the passenger, the next task is confirming the passengers get on the car.



picture 2. driver cmmn

6 rule

r:Driver receive an order \vdash OBLconform the order \otimes OBLwait for the next order
 v:Driver conform an order \vdash OBLpick up the passengers

Driver receive an order is obligatory to conform, If not conform the order, driver has to wait for the next order. Driver conform an order is obligatory to pick up the passengers.

```

1 <Imp label='r'>
2   <body>
3     Driver receive an order
4   </body>
5   <head>
6     <Behaviour>
7       <Obligation>confirm the order</Obligation>
8       <Obligation>wait for next order</Obligation>
9     </Behaviour>
10  </head>
11 </Imp>
12 <Imp label='v'>
13   <body>
14     confirm the order
15   </body>
16   <head>
17     <Behaviour>
18       <Obligation>pick up the passengers</Obligation>
  
```

```

19 |         </Behaviour>
20 |     </head>
21 | </Imp>

```

7 dmn discribe

This dmn describes the situation of the driver accepting the order. If the order distance is greater than 2 km, the driver would refuse. If the order shows bad passenger credit, the driver would refuse. If the order shows more than 5 passengers, the driver would refuse. If the order is greater than 50km, the driver would refuse. If the order is greater than 50km, but the driver received a surcharge, the driver would accept. If the order shows good passenger credit, the driver would accept.

Order							View DRI
decision_order							
U	Input +						Output +
	distance	credit	people number	destination	surcharge	order	
	integer	string	integer	integer	integer	string	
1	≥2	-	-	-	-	Refuse	
2	-	bad	-	-	-	Refuse	
3	-	-	≥5	-	-	Refuse	
4	-	-	-	≥50	-	Refuse	
5	-	-	-	≥50	≥5	Accept	
6	≥2	good	-	-	-	Accept	

8 Petri Net

This Petri Net describes the process of picking up the passenger and record the mileage until the settlement.

