CSCI927 Service-Oriented Software Engineering (Assignment)

Student Name: Xiao Yao
CCNU Student Number: 2019180015
UOW Student Number: 6641751

Central China Normal University Wollongong Joint Institute

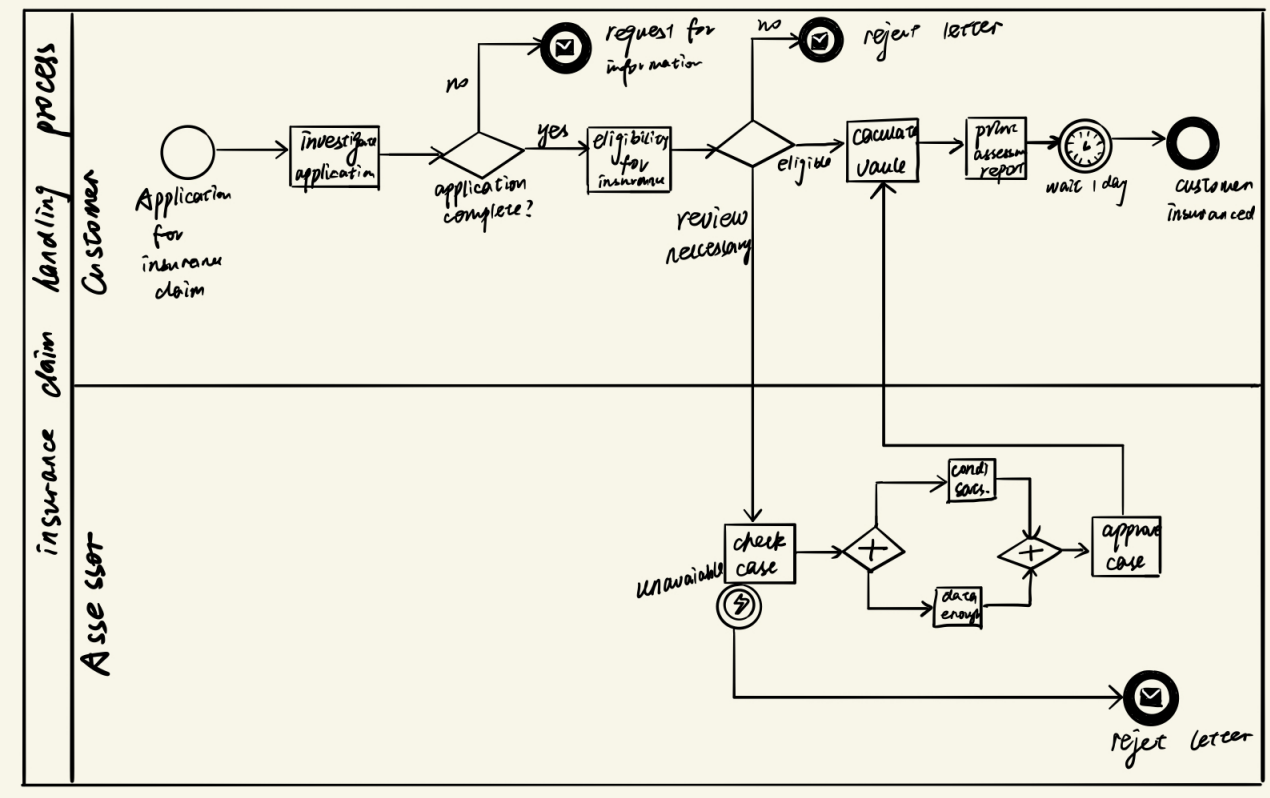
1 PART ONE (7 MARKS)

1.1 Description on Designed Insurance Claim Handling Process

Considering that the insurance claim process is not a single act, but an interactive act, I chose to use lanes to represent different roles, including customers and bank assessors. At the same time, insurance claims are also a complicated and task-intensive process. In order to better understand the situation of each customer, you first need to fill out a detailed application form. If the application form is not completed, a rejection letter will be sent. After the application form is completed, check whether the customer is eligible for insurance according to the content filled out by the customer. If the qualification is not qualified, a rejection letter will be sent. If the qualification is met, the next stage will be entered to calculate the insurance amount. However, if it is impossible to judge the need to re-examine, it is necessary. Transfer it to the bank assessor for review. If the bank assessor is still unable to judge, the exception is thrown directly, and a rejection letter is sent. If the bank assessor checks both the condition and all the information, the application is agreed and the insurance amount is calculated. After calculating according to the payment conditions, the insurance policy is sent to the user for confirmation. After receiving the confirmation email, the user needs to wait for one working day to receive the payment and the payment documents.
Here is my references:

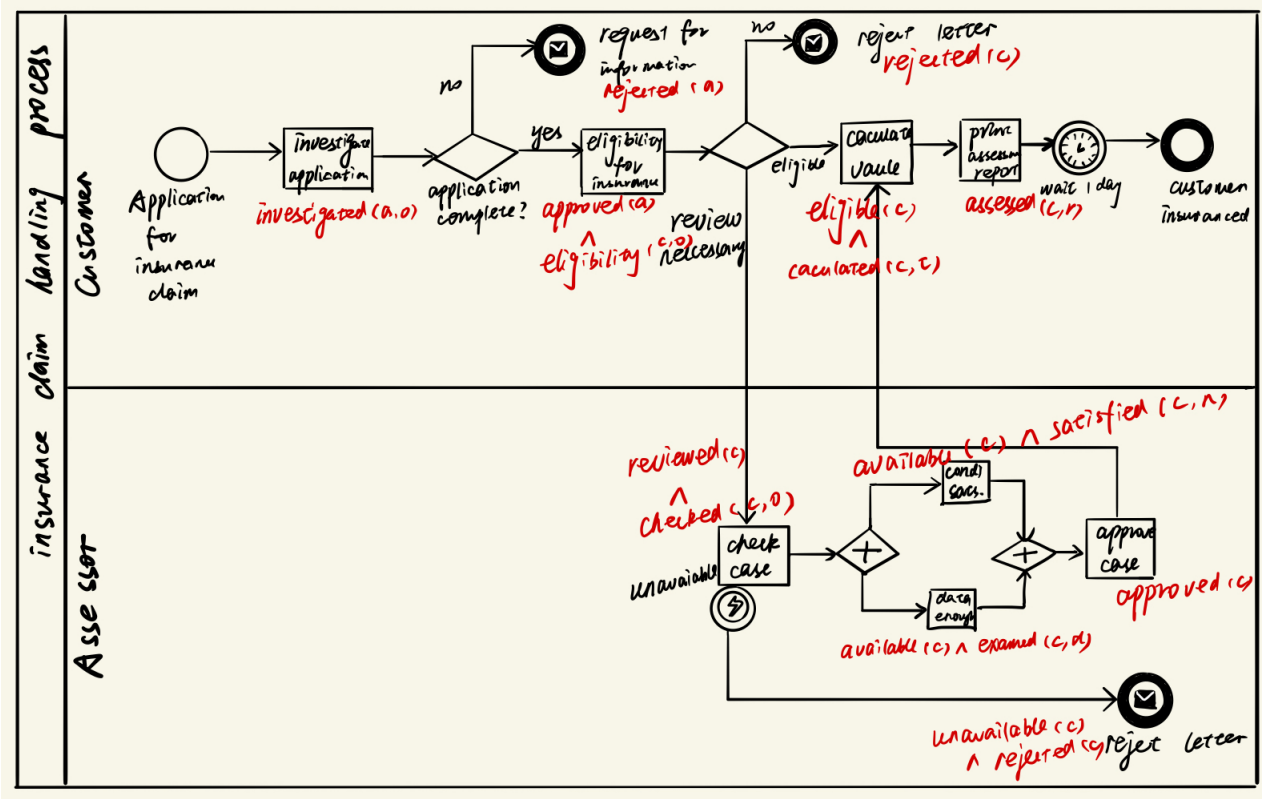
1. https://www.signavio.com/post/insurance-example-processing-a-claim/
2. https://consumer.findlaw.com/insurance/the-insurance-claim-process.html
3. https://www.oecd.org/finance/insurance/33964905.pdf
4. https://www.dir.ca.gov/dwc/InjuredWorkerInfo/Claimsprocess.pdf

1.2 Designed BPMN Model



1.3 Semantic Effect Annotation

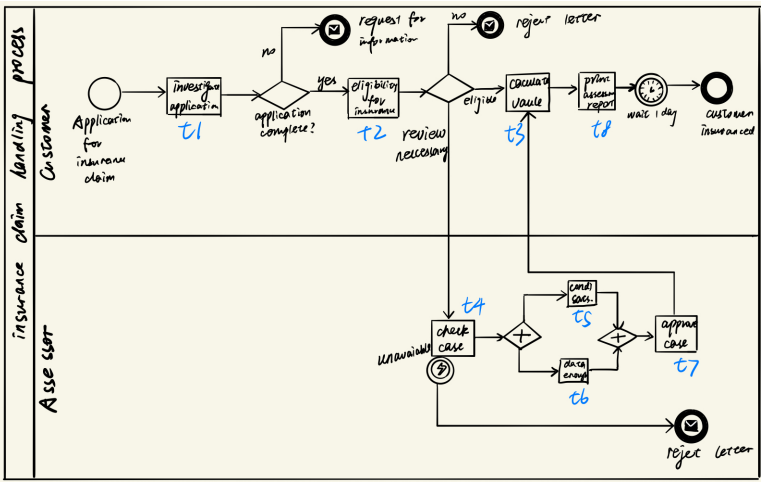
1. Objects of interests: Application a, Claim c, Outcome c, Amount t
2. States:
   - Application rejected or approved - rejected(a), approved(a)
   - Claim rejected or approved or eligible or reviewed - rejected(c), approved(c), eligible(c), reviewed(c)
3. Relationships between objects:
   - Application investigated with an outcome - investigated(a,o)
   - Claim eligibility with an outcome - eligibility(c,o)
   - Claim checked with an outcome - checked(c,o)
   - Claim satisfied with conditions - satisfied(c,n)
   - Claim examed with data - examed(c,d)
   - Claim calculated with an amount - calculated(c,t)
   - Claim assessed with a report - assessed(c,r)



(a) Cumulative Effects of Tasks/Activities

1. Reject Application: investigated(a,o) ∧ rejected(a)
2. Store Username: investigated(a,o) ∧ approved(a) ∧ approved(u) ∧ stored(u,s)
3. Reject Update Username: investigated(a,o) ∧ approved(a) ∧ approved)(u) ∧ interrupted(u) ∧ rejected(u)
4. Update Username: investigated(a,o) ∧ approved(a) ∧ approved(u) ∧ stored(u,s) ∧ updated(u,s)
5.

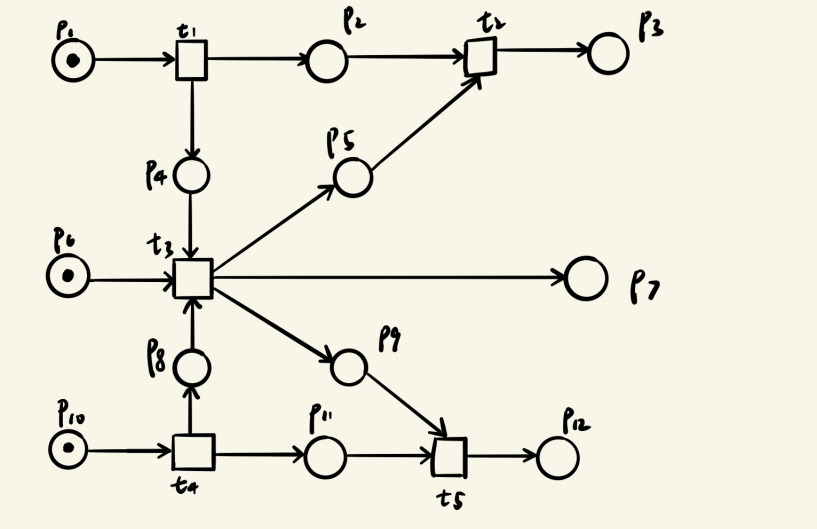
(b) Cumulative Effect Scenarios



1. t_3:
   - Scenario(1): $\langle\{t_1, t_2, \{\langle t_3\rangle\}\rangle, \{\langle t_1, t_2, t_4, \{\langle t_5, t_6\rangle\}, t_7\rangle\}\rangle$
   - Scenario(2): $\langle\{t_1, t_2, t_4, \{\langle t_5, t_6\rangle\}, t_7, \{\langle t_3\rangle\}\rangle, \{\langle t_1, t_2\rangle\}\rangle$
2. t_8:
   - Scenario(1): $\langle\{t_1, t_2, t_3, \{\langle t_8\rangle\}\rangle, \{\langle t_1, t_2, t_4, \{\langle t_5, t_6\rangle\}, t_7, t_3\rangle\}\rangle$
   - Scenario(2): $\langle\{t_1, t_2, t_4, \{\langle t_5, t_6\rangle\}, t_7, t_3, \{\langle t_8\rangle\}\rangle, \{\langle t_1, t_2, t_3\rangle\}\rangle$

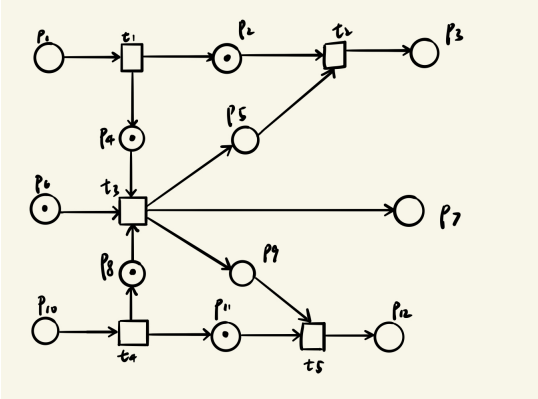
2 PART TWO (3 MARKS)

2.1 Designed Petri Net

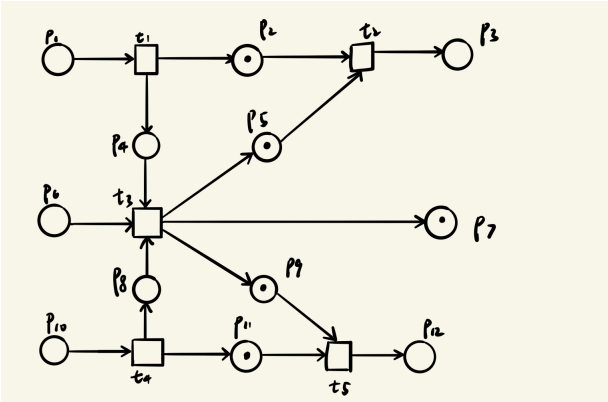


2.2 Analysis on the Petri Net

1. It can see that the t_1 conditions$\{p_1, p_2, p_4\}$, so the token start with the p_1 and p_{10}, and pass by concurrent actions t_1 and t_4.



2. And then input p of t_3 is marked with $w((p,t))$ tokens, so the transition t_3 is enabled, the token pass by t_3 to p_7.



3. At last, the same as above, transition t_2 and t_5 is enabled, the racing game is over.