UNIVERSITY OF WOLLONGONG AUSTRALIA

School of Computer Science

CSCI992-Requirement/domain analysis

2 April 2018

Group B

Qiusheng Chu	qc851@uowmail.edu.au
Jianbo Zhao	jz491@uowmail.edu.au
Hoang Kim Nguyen	hkn193@uowmail.edu.au
Junxin Ren	jr239@uowmail.edu.au
Wenqiang Xu	wx432@uowmail.edu.au
Ashwin Dilip Kharat	adk829@uowmail.edu.au
Sameer Vijay Maharjan	svm636@uowmail.edu.au
Xinyu Zhang	xz906@uowmail.edu.au
Syed Ali Abbas Naqvi	saan977@uowmail.edu.au

Document Control

Title: Requirement/domain analysis

Document Name:

Owner	Current Version	Last Change on		Approved by
		Date	Time	
	1.6	02/04/2018	19:30	Xinyu Zhang

Record of Revision				
Revision Date	Description	Section Affected	Changes Made by	Version after Revision
01/10/2017	First draft of the document	Whole document	Qiusheng Chu	1.1
02/10/2017	Add Format to the draft version	Whole document	Xinyu Zhang	1.2
02/10/2017	Add Definitions and acronyms part	Definitions and acronyms	Xinyu Zhang	1.3
02/10/2017	Add 4 user case diagram and modify some detail of layout	Whole document	Wenqiang Xu	1.4
01/04/2018	Refine the document	Whole document	Qiusheng Chu	1.5
02/04/2018	Add UML diagrams	Whole document	Wenqiang Xu	1.6

Table of Contents

Introduction

Purpose:

With the development of the e-commerce, there are more and more websites or online platforms which can provide tailored service for clients. By doing the research in commonly used websites which includes Amazon, E-bay, UberEats, etc., we find that all of them do not have the functionality of service bartering. In this project, we will design and implement the service trading online platform which is based on both main functionalities from service to currency and from service to service in two ways exclusively for students of university of Wollongong.

Scope:

From the perspective of the low-income users, instead of trading service with money, it would be quite acceptable and affordable for them to receive the specific services by bartering their services to the providers in return. Undoubtedly, low-income people would be one of beneficiary groups, like international students etc. However, we still preserve the currency and virtual credits exchanging for some high-income users, which could be convenient for them to buying service directly, not by services bartering.

This document is exclusive for online commercial platforms designers and website technicians.

Definitions and acronyms:

CSI: composite satisfactory index.

CEBSL: conditional entropy based supervised learning.

H(Y|X): conditional entropy.

I(Y|X): information gain.

Pr(Y|X): the conditional probability.

N: nationality.

M: Major

L: last search result.

Class: the excepted output.

Overall Description:

Our service bartering platform mainly focus on the UOW students. So, from the design point of view, most functionalities will suffice that particular group of people.

Requirements:

From the perspective of project, we mainly have the following four parts of main functional requirements, which are user management, service management, payment management and database management.

1. User management

(1) User profiles creation, modification, querying and account close

For each new user, they must register their accounts by providing the authentic information including the mandatory and optional fields.

Mandatory:

-Nickname, first name & last name;

- E-mail address (UOW) (log in)

- Passwords (complexity check)

- Zip code

- Suburb

- link (via e-mail)

Optional:

- Mobile phone

- Nationality

- Date of birth

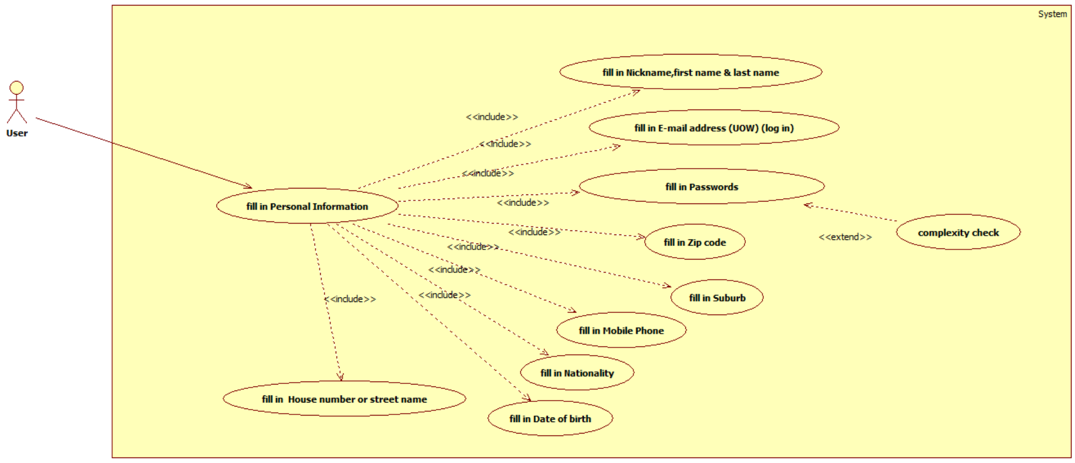
- Potential interests

The noticeable thing is that due to that our service trading platform is exclusive for UOW students, all users should provide their UOW e-mail address for authentication and identification. After filling the field with qualified e-mail address, hyper link would be sent to their mail box. By clicking the provided link, identity of users would be authenticated. The optional fields would not only help users to complete their profiles, but also become the source for data mining in the future. Basically, user profile modification, information query and account close will also be functionalized.

Inputs: field data

Outputs: User's profile

User case:

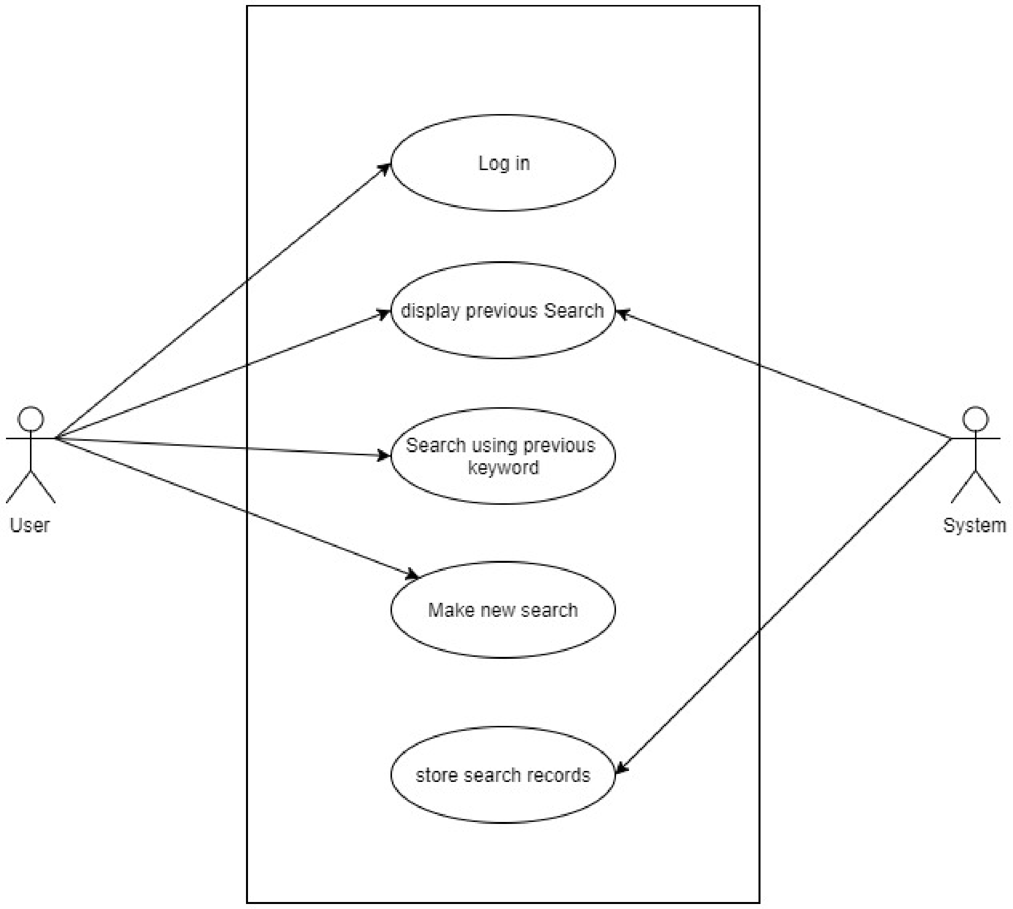


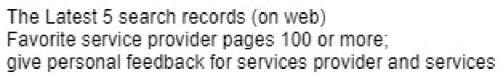
(2) User personal searching records.

User personal searching records would be saved in the database. Every time searching services by keywords, users will be informed with the latest 5 records.

Inputs: history search records

User case: selecting the latest 5 records




The Latest 5 search records (on web)
Favorite service provider pages 100 or more;
give personal feedback for services provider and services

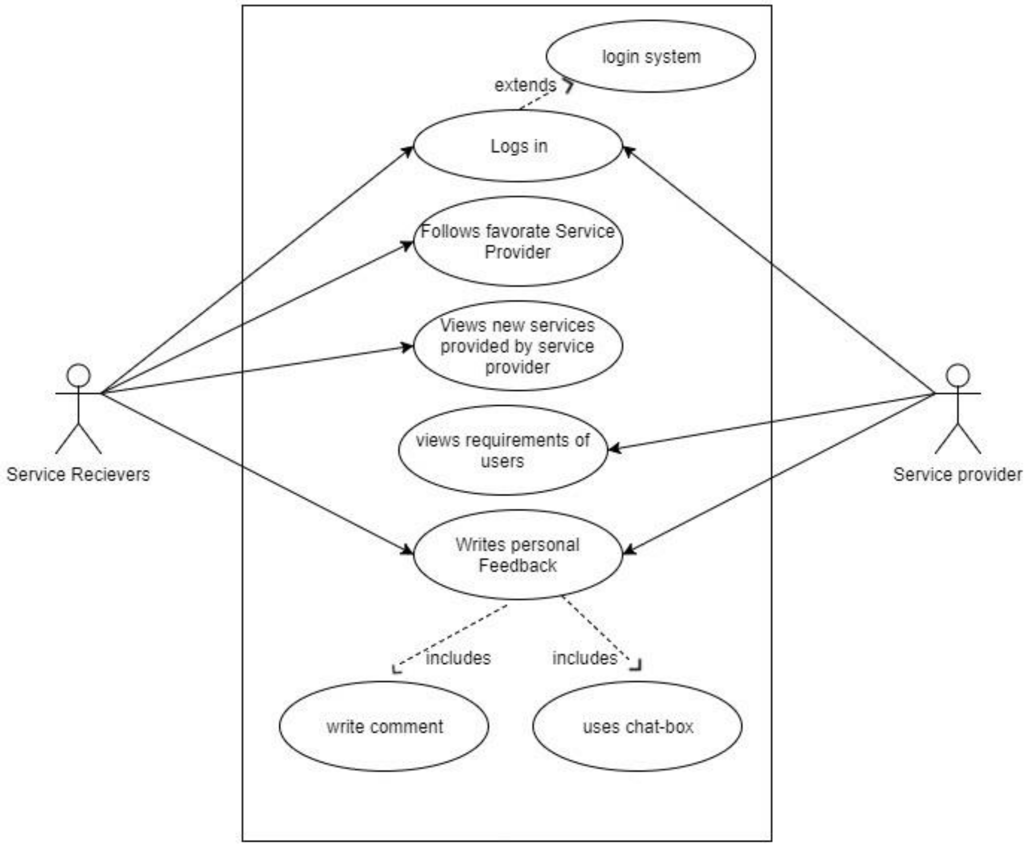
(3) Collect favorite services.

Users could collect at most 150 favorite services. In each service section, users can provide feedback, suggestions or compliments to it.

Inputs: the different favorite services

Outputs: retrieve of the favorite services collection

User case: collecting favorite services



(4) User log in and log off.

Log in and log off are two basic functions for users to connect and disconnect with their accounts. When users log in, they must provide the right e-mail address as the user name and matched password.

Inputs: username & password

Outputs: authentication results

User case: users log in and off

(5) User credit assessment.

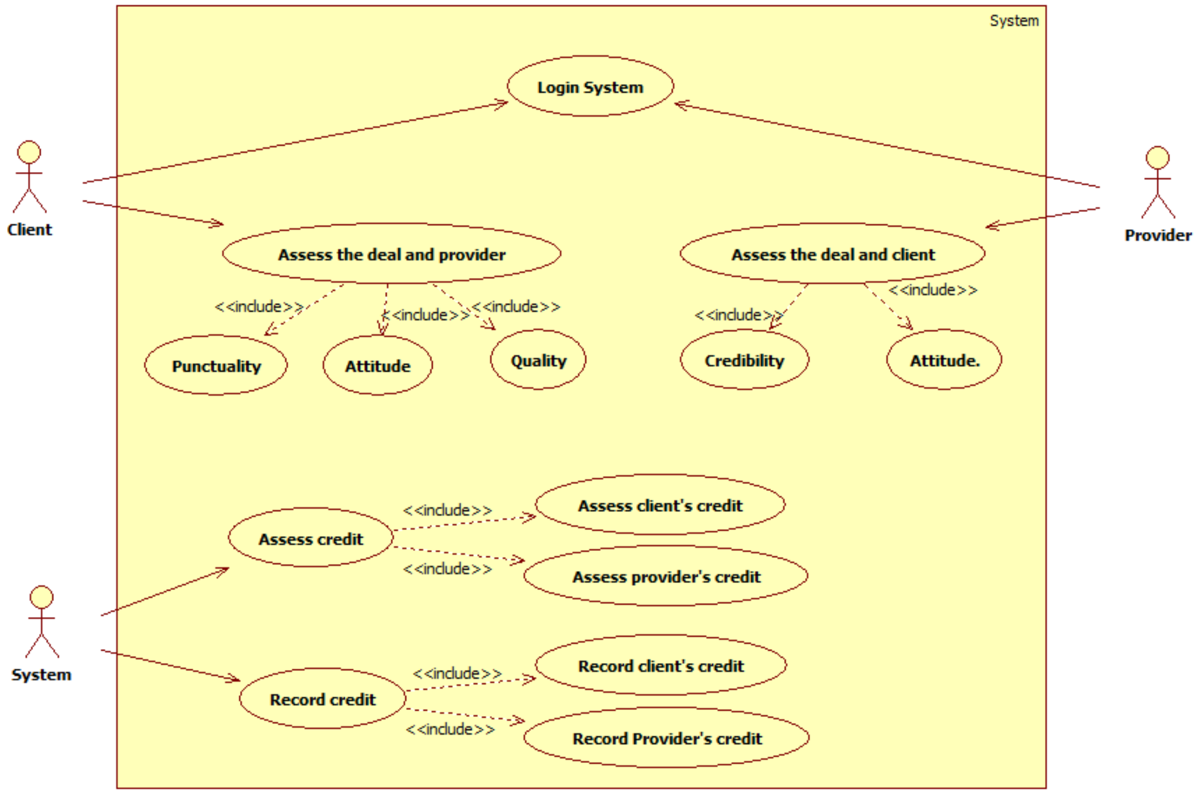
Periodically, users' credit will be assessed by the functionality based on credit marks given by service providers or receivers. After delivering service, both providers and clients will leave the assessments to each other in a numeric way which include punctuality, attitude and quality. By granted with different weights (Wp, Wa and Wq), we could get the composite satisfactory index (CSI) for each service from the both sides of providers and clients. For example,

CSI = Wp x punctuality + Wa x attitude + Wq x quality.

Inputs: users' conduct data

Outputs: CSI (composite satisfactory index)

User case: credit assessment



2. Service management

(1) Personal interest recommendation.

For new users, personal preferences setting can show the relevant service which they might be interested in. After that certain times of searching, we could provide users with the more precious predictions by employing the conditional entropy based supervised learning (CEBSL).

Inputs: previous searching records and all potential factors

Outputs: the prediction of interests

User case:

Conditional entropy H(Y|X): With the occurrence of condition A, the amount of uncertainty provided by condition B.

We assume that all data is discrete which means the happening of all events are independent, so the discrete conditional entropy formulation will be employed. Pr(Y|X) is the conditional probability.

$$H(Y \mid X = v) = \sum_{i=1}^{k} - \Pr(Y = y_i \mid X = v) \log_2 \Pr(Y = y_i \mid X = v)$$

$$H(Y \mid X) = \sum_{v: \text{values of } X} \Pr(X = v) H(Y \mid X = v)$$

Information gain manifests the certainty of X for the occurrence of the condition Y.

I(Y|X) = H(Y)-H(Y|X).

So, based on the information gain, we can figure out how much contribution each factor or condition has towards specific output which we expect.

After pre-processing raw user data and measuring the information gain for each factor, we can set up the decision tree which is a k-ary tree. Each node has less or equal thank children. Layers from top to down follow the sequence of information gain value.

Here is a small example for CEBSL.

If in the database, we get the pre-processed data as the following table.

Record number	Nationality(N)	Major(M)	Last search result(L)	Class (the expected output)
1	Chinese	CS	C++	Y=searching Java
2	Indian	CS	C++	Y=searching Java
3	Chinese	Bio	C++	Y=searching Java
4	Chinese	Bio	Cuisine tutoring	-Y=other
5	Australia	Mining	Cuisine tutoring	-Y=other
6	Australia	Mining	C++	-Y=other

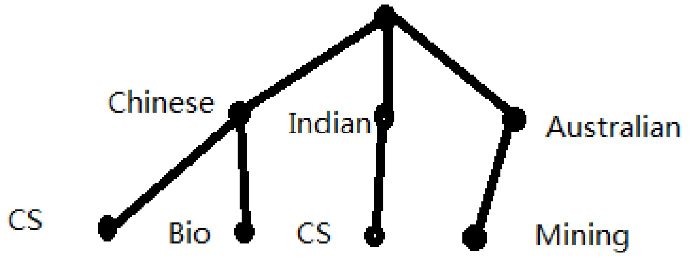
H(Class|Major)=4/6xH(1/2,1/2)+2/6xH(1/2,1/2)

I(Class; M) =H(Class)-H(Class|M)=0 bits

I(Class; L)=H(class)-H(Class|L)=0.46 bits

I(Class; N)=H(Class)-H(Class|N)=0.54 bits

From the information gain of all factors, we know that in all three conditions, nationality has the largest repercussion towards the output Y. So, we build the decision tree below with the nationality as the first layer and last search result as the second layer. Since major does not have any impact on the output Y, it is eliminated.



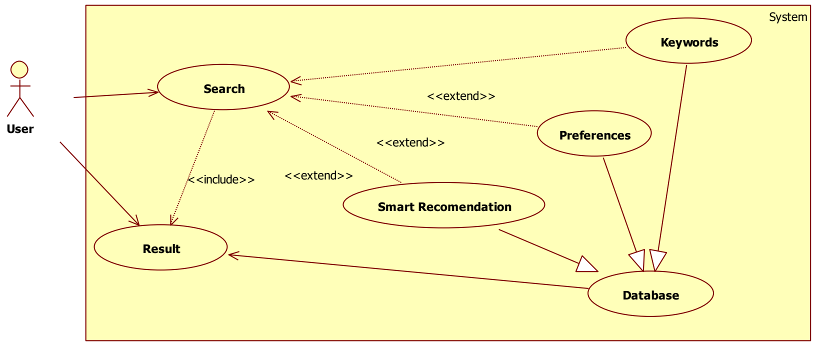
Since the implementation of the algorithm part will be time-consuming and require high familiarity in Python and all related API, in our project, we will mainly put emphasis on algorithm design, potential problems solving and program it as much as we can.

(2) Key words services searching and default displaying

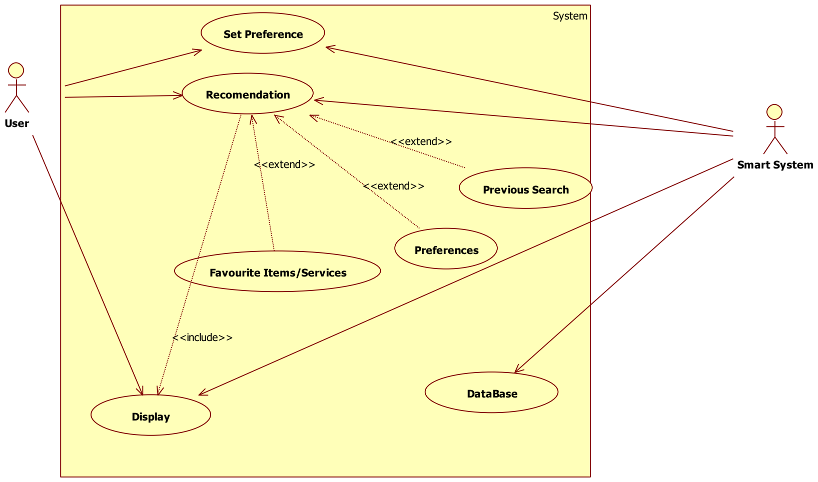
Our platform offers multiple rules for searching and users could define rules by themselves. For example, services could be sorted out by categories, service providers' nationality and highest "like" or "dislike" rates. If users do not define any rules for searching, by default, all services will be ranked by the compound index which takes certain service parameters and their weights. The higher compound index services get, the topper positions services will be in.

Inputs: keywords

Outputs: different sequences and ranks for services



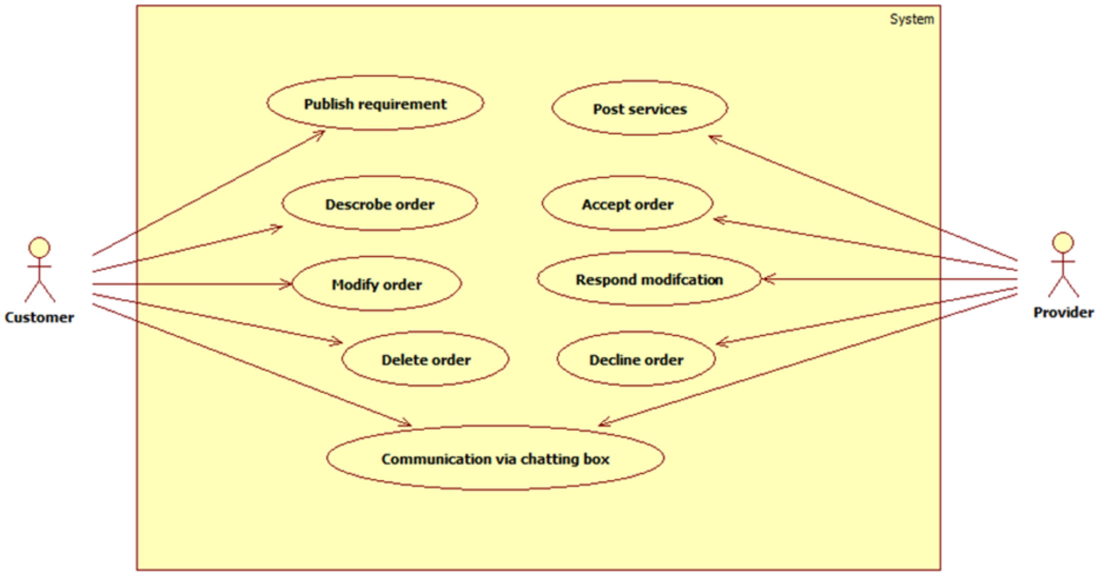
For smart search



(3) Service barter and chat box

Service bartering can provide a way for negotiating between service providers and clients by chatting box.

User case:



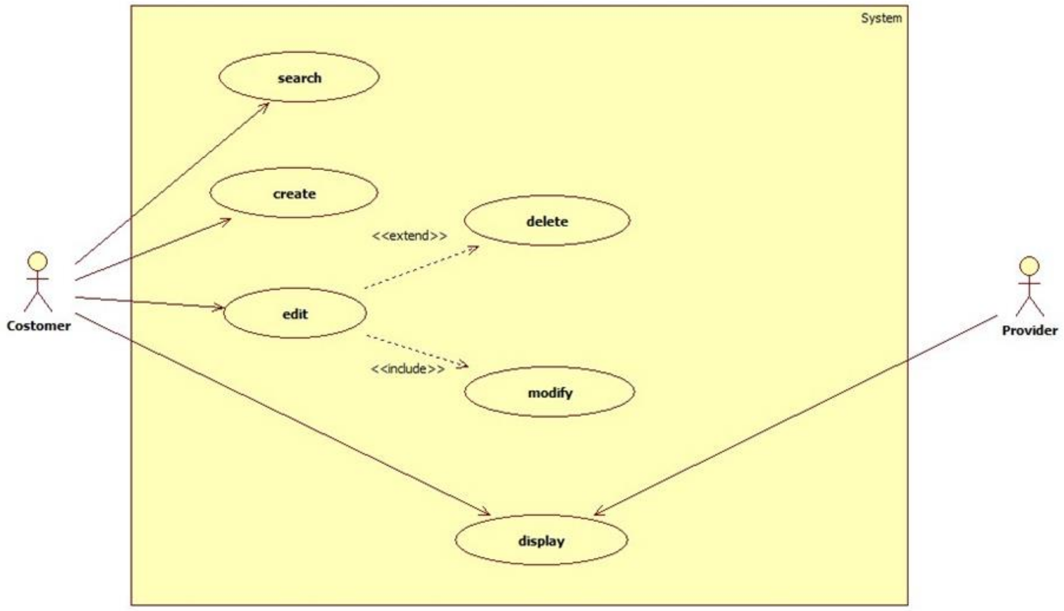
(4) Leave comment and comment display.

Everyone can commend services and they all have rights to delete or modify their own words. However, all commends will be recorded in database for the potential conflict reconciliation between providers and clients.

Inputs: comments

Outputs: comments retrieval

User case:



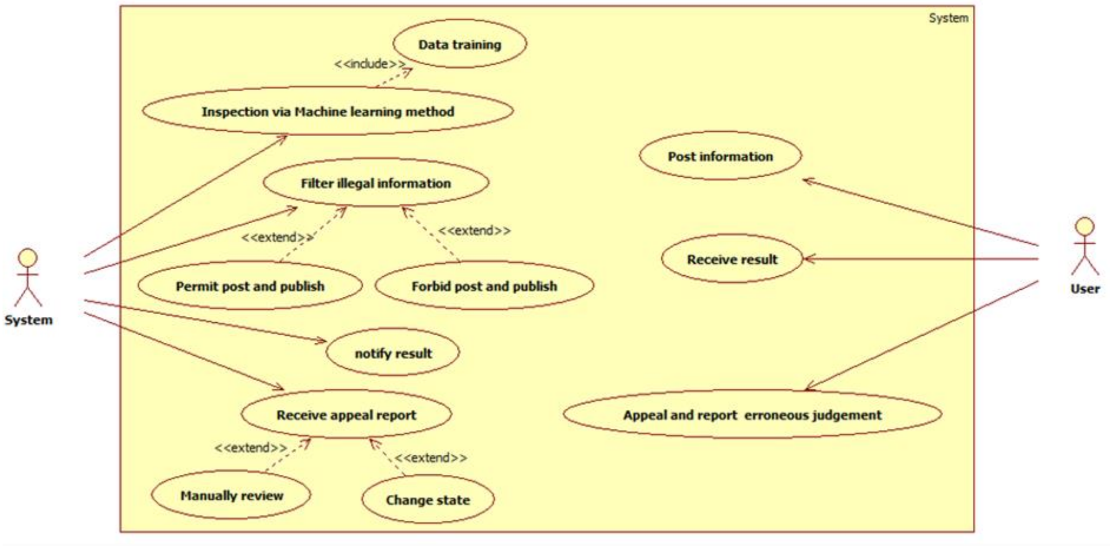
(5) Service transferring

Among all different service providers, service would be recommended and transferred. The reason for using this functionality is that people are not always available. The any service transferring must get clients' agreements. Or, providers must re-negotiate with clients again.

Inputs: service agreement between providers to clients

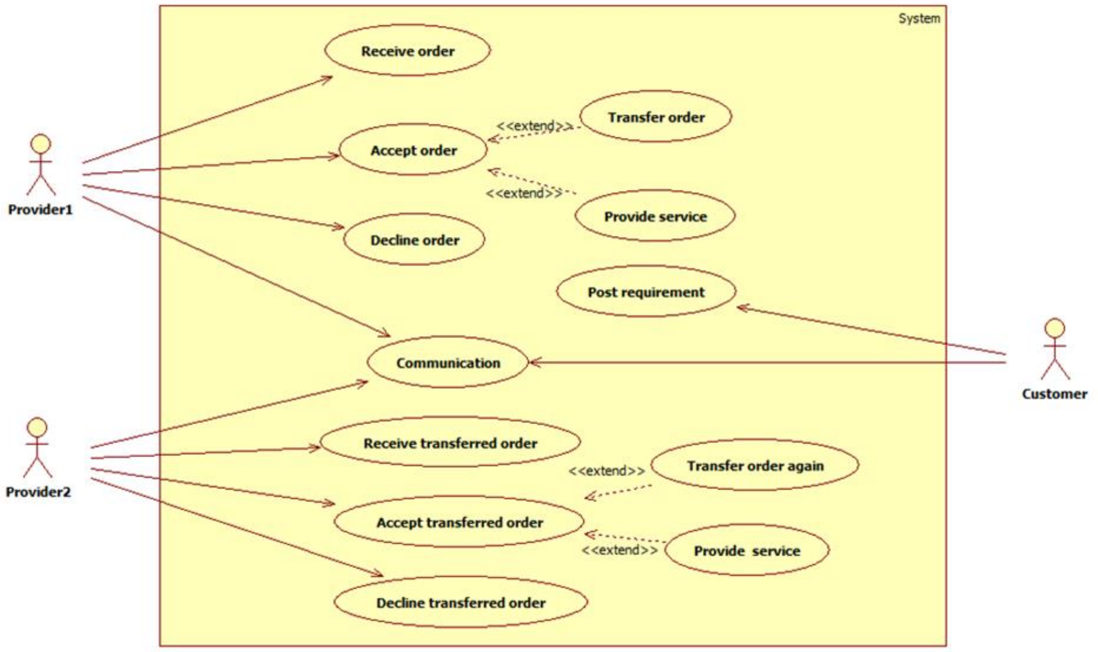
Outputs: service agreement between other providers to clients

User case:



(6) Service conformation

After delivering and receiving services, both clients and providers will input the pre-define service conformation code for finishing the delivery.



3. Payment management

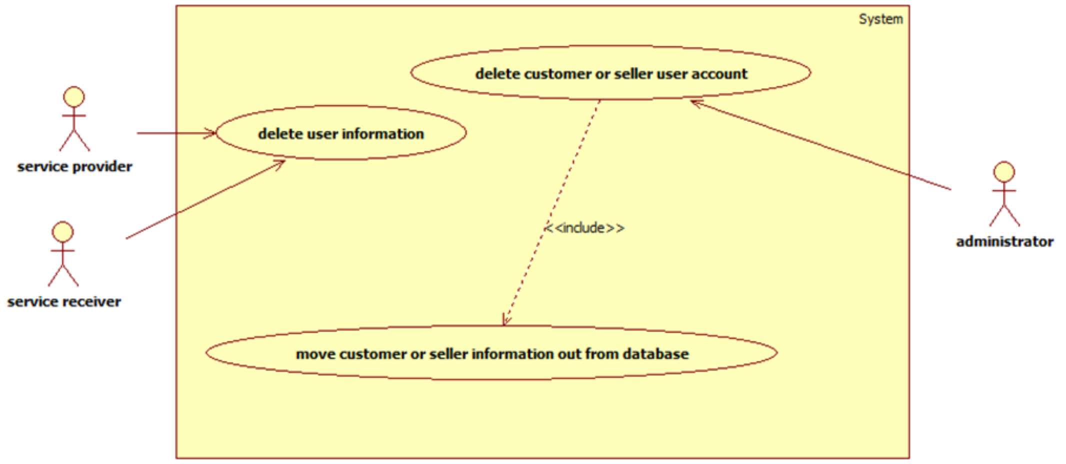
(1) Virtual credit transferring and credit to cash bi-exchange.

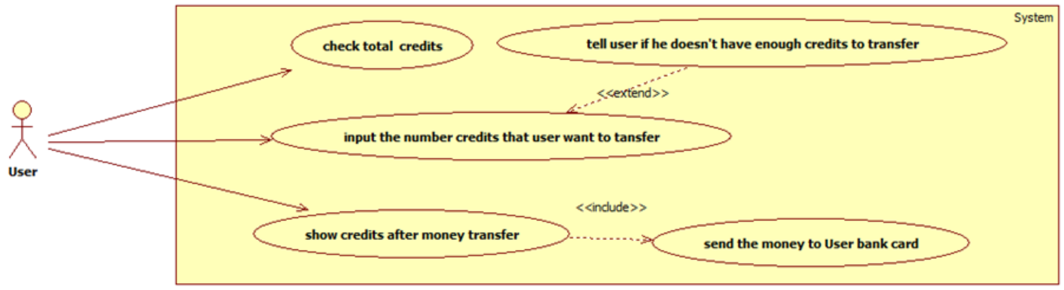
For high-income users, they can exchange virtual credit by cash and buy services and virtual credits could be exchanged with cash. However, the exchange rate from cash to credits is a bit higher than the rate from credits to cash for making profits.

Inputs: service conformation code

Outputs: Service conformation judgment

User case:


System
delete customer or seller user account
delete user information
service provider
service receiver
<<include>>
administrator
move customer or seller information out from database


System
check total credits
tell user if he doesn't have enough credits to transfer
<<extend>>
User
input the number credits that user want to tansfer
<<include>>
show credits after money transfer
send the money to User bank card

4. Non-functional requirements which contains Database management

(1) Database management

Data redundancy:

manually data backup, automatic database backup and duplicate the data by way different ways

Database model (round robin or master & slave model):

find a better way to access database.

(2) Webpage requirements

Web page beautifying and cosmetic effect

(3) Platform security

Mainly, there are two security concerns, database crash down or data loss and user conduct security. To deal with the first concern, redundant database design can be a good choice. In database, all tables will have the service duplicated backup copies. Even when database crashes down, manual redundancy will preserve all previous records at least.

All user conducts will be preserved for conduct assessment, which includes users searching records, comments , service bartering exchanges and so on.

References:

Greengard, S 2016, 'How AI Can Improve Searching for Data', CIO Insight, p. 1.

Maddux, S 2017, 'Searching in a Smarter Way', Multichannel News, vol. 38, no. 12, p. 24.

STACKPOLE, B 2017, '100 SMARTER BETTER FASTER: HOW CIO 100 HONOREES DELIVER BUSINESS RESULTS WITH AI & MACHINE LEARNING', CIO, pp. 2-15.