Service Register System

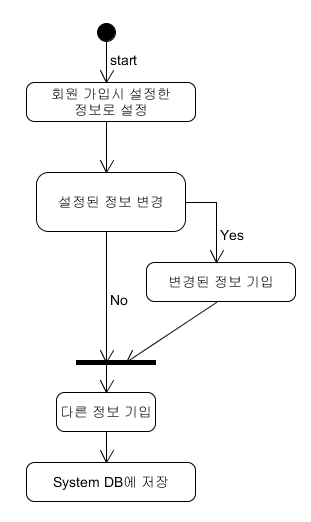
A. Objective

서비스 이용자(buyer)가 특정 양식에 따라 희망하는 서비스에 대한 정보를 입력하고, 이를 데이터 베이스에 저장하는 Service Register System의 설계를 설명한다. Service register system의 sequence diagram과 class diagram, pseudo code를 통해 Service register system의 구조를 표현한다.

B. System Organization

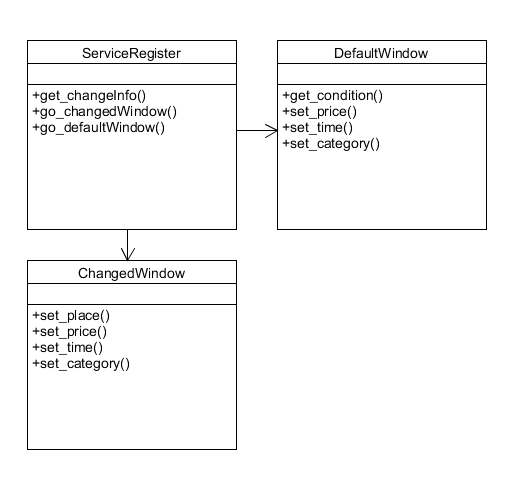
Service Register System은 subsystem을 가지지 않는 단일 system이다. 희망 장소를 설정 이후에, 장소 희망 장소 이외의 정보들을 기입한 후, 희망 서비스 등록 완료 알림을 송출한다.

C. Activity Diagram



C. Class Diagram

Register system을 Class Diagram으로 표현하면 다음과 같다.



|  |  |
| --- | --- |
| get\_changeInfo() | 이용자가 회원가입시 사용한 정보를 그대로 서비스 register 단계에서 사용하는지에 check했는지에 대해 파악하고 그 값을 return 한다. |
| go\_changefInfoWindow() | 이용자가 회원가입시의 정보를 변경하여 service register에 기입하고자 한다면, chagedInfoWIndow를 불러온다. |
| go\_defaultInfoWidow() | 이용자가 회원가입시의 정보를 그대로 기입하고자 한다면, chagedInfoWIndow를 불러온다.. |

Personalization System

A. Objective

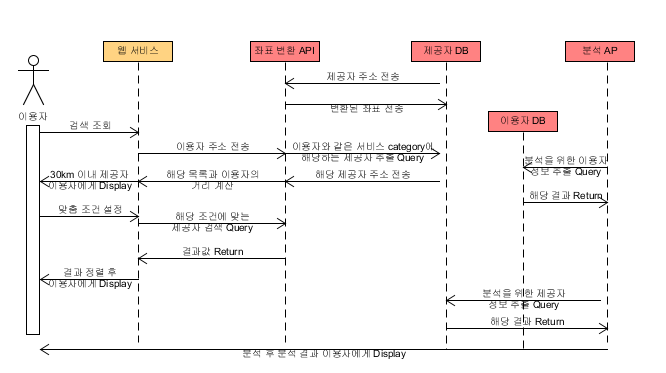
Personalization System은 이용자가 자신이 원하는 서비스에 맞는 제공자를 검색하고, 제공자를 추천해주는 기능을 제공한다. 검색의 경우, 이용자가 입력한 상세정보를 기반으로 filtering을 진행하는 검색 서비스를 제공한다. 추천 서비스의 경우, 이용자의 거주지 혹은 서비스 등록 시에 지정한 장소를 기반으로 30km 이내의 제공자들을 추천해준다. 과거 데이터 기반 추천의 경우, 과거 데이터를 분석하여 제공받을 서비스를 추천한다.

B. System Organization

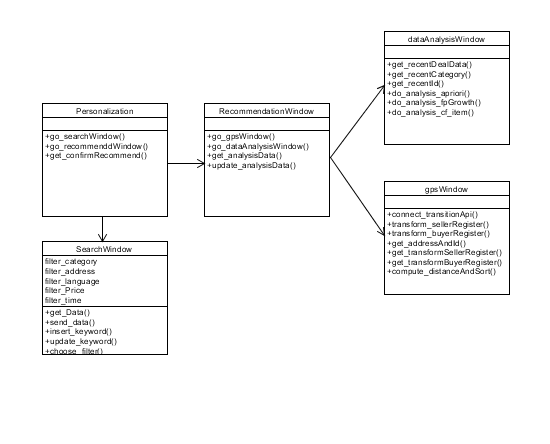
Personalization System은 크게 Search system과 Recommendation system으로 구성된다. Search system은 이용자가 원하는 서비스에 해당하는 제공자를 직접 검색하는 기능이다. Recommendation의 경우 거리 기반 recommendation system과 과거 데이터 기반 recommendation system을 subsystem으로 구성된다.

C. Sequence Diagram

다음 diagram은 personalization system을 표현한 sequence diagram이다.



D. Personalization System Class Diagram



|  |  |
| --- | --- |
| get\_confirmRecommend() | 이용자 DB에서 이용자가 추천에 동의했는지에 대해 파악하고, 그 값을 return한다. |
| go\_searchWindow() | 이용자가 검색 조회를 실시했다면, Search Window를 불러온다. |
| go\_recommendWindow() | 이용자가 추천 서비스에 사전 동의했다면, 검색 조회할 때, Recommend Window를 함께 불러온다. |

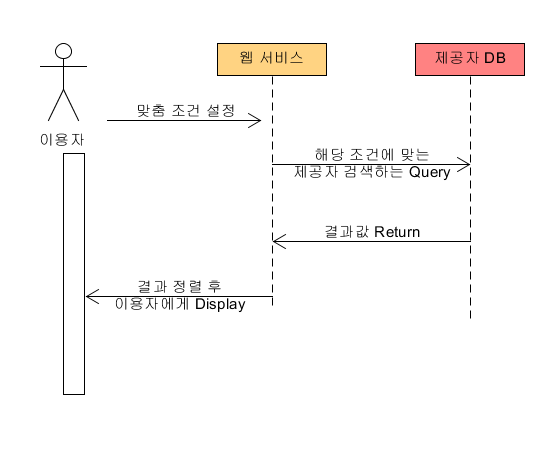
E. Subsystem

E.1 Search System

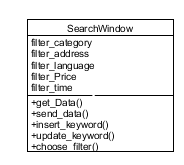
E.1.1. Definition

이용자가 원하는 서비스를 제공받기 위해 검색 조회를 할 때, 이용자가 정한 세부 정보에 따라 filtering하여, 해당 정보를 표시해주는 기능을 한다.

E.1.2.1. Sequence Diagram.



E.1.2.2. Class Diagram



|  |  |
| --- | --- |
| filter\_category | 서비스 항목에 대한 카테고리 값 |
| filter\_address | 제공자 거리 (30km 이내 or not) |
| filter\_language | 제공자 사용 언어 |
| filter\_price | 가격 정보 값 |
| filter\_time | 시간 정보 값 |
| get\_data() | DB로부터 데이터를 가져온다. |
| send\_data() | DB로 데이터를 보낸다 |
| insert\_keyword() | Keyword 입력 |
| update\_keyword() | Keyword 수정 |
| choose\_filter() | Filter 옵션 선택 |

E.1.3. pseudo code

E.2 Recommendation System

E.2.1. Definition

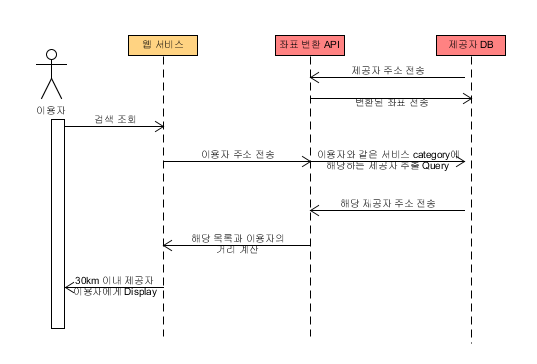
이용자가 원하는 서비스를 제공해주는 제공자를 쉽게 찾을 수 있도록, 이용자의 주소 혹은 지정한 장소를 바탕으로 30km 이내의 제공자들을 추천해주거나, 과거 데이터를 기반으로 다음 서비스를 추천해주는 기능을 한다.

E.2.2. Subsystem Organization.

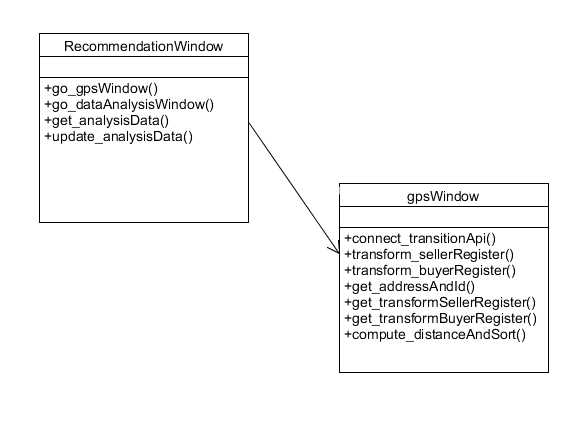
Recommendation System은 거리 기반 추천 system과 과거 데이터 기반 system을 subsystem으로 구성된다.

E.2.3.1. 거리 기반 추천 System Sequence Diagram

거리 기반 추천의 경우, 좌표 변환 API를 사용한다. 제공자 주소를 미리 좌표 변환하여 제공자 DB에 저장해두고, 이용자가 검색 조회를 실시할 때, 이용자의 주소를 좌표 변환한다. 이용자가 지정한 service category에 해당되는 제공자의 주소와 id를 추출하고, 제공자 목록의 변환된 좌표와 이용자의 변환된 좌표를 계산하고, 30km 이내의 제공자들을 추출하는 query로 filtering하여 웹에 표시한다.



E.2.3.2. 거리 기반 추천 System Class Diagram



|  |  |
| --- | --- |
| go\_gpsWindow() | 검색 조회시 gpsWindow를 불러온다 |
| connect\_transitionApi() | 좌표변환 api 연결 |
| transform\_sellerRegister() | 제공자 주소 좌표로 변환 |
| transform\_buyerRegister() | 이용자 주소 좌표로 변환 |
| get\_addressAndId() | 제공자 address 와 Id 추출 |
| get\_transformSellerRegister() | 변환된 제공자 주소 불러오기 |
| get\_transformBuyerRegister() | 변환된 이용자 주소 불러오기 |
| compute\_distanceAndSort() | 이용자와 제공자 목록의 거리 계산 및 내림차순으로 정렬 |

E.2.3.3 pseudo code (Python 기반)

# connect\_transitionApi()

# transform\_sellerRegister()

# transform\_buyerRegister()

# compute\_distance

from math import sin, cos, sqrt, atan2, radians

R = 6373.0

// buyerdata=get\_transformBuyerRegister()

for i in buyerData:

i\_lon = i[4]

i\_lat = i[3]

// sellerData=get\_transfromSellerRegister()

for row in sellerData:

sellerData\_lon = row[3]

sellerData \_lat = row[4]

lat1 = radians(float(i\_lat))

lon1 = radians(float(i\_lon))

lat2 = radians(float(popular\_lat))

lon2 = radians(float(popular\_lon))

dlon = lon2 - lon1

dlat = lat2 - lat1

a = sin(dlat / 2)\*\*2 + cos(lat1) \* cos(lat2) \* sin(dlon / 2)\*\*2

c = 2 \* atan2(sqrt(a), sqrt(1 - a))

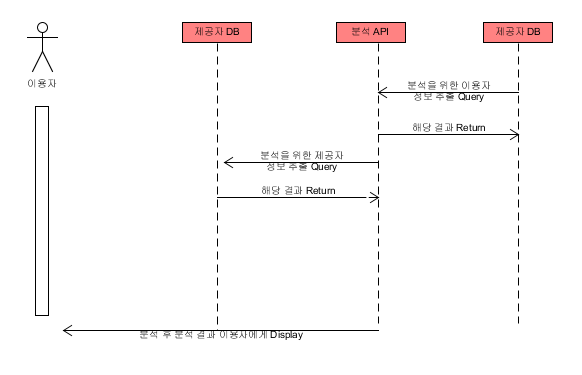
distance = R \* c

if(distance < 30):

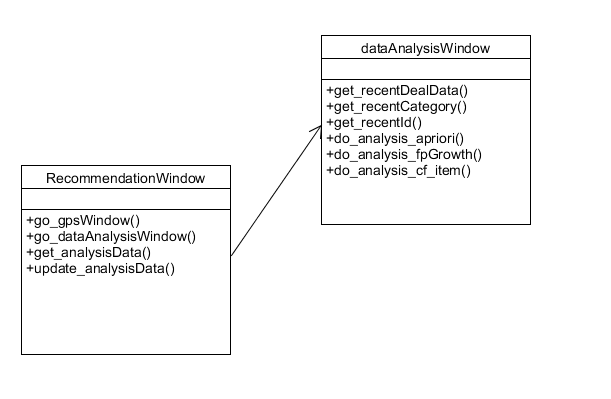
print("Result: ", sellerData[2], ”:” , distance)

E.2.3.1. 과거 데이터 기반 추천 System Sequence Diagram

과거 데이터 기반 추천 system의 경우, 이용자와 제공자의 DB에서 최근 거래 목록과 서비스 Category, Rating을 추출한다. 추출한 데이터를 바탕으로, 연관성을 분석하는 apriori package의 연관성 분석 혹은 조금 더 빠른 FP growth 기법을 사용하여, 항목 간의 서비스를 분석한다. 추가적으로 데이터가 모이게 되면, item 기반의 collaborative filtering을 사용할 예정이다. 이후에는 다음 서비스 항목 추천을 넘어서, 다음 서비스 항목에, 알맞은 제공자를 추천해주는 시스템을 개발 예정이다.



E.2.3.2. 과거 데이터 기반 추천 System Class Diagram



|  |  |
| --- | --- |
| go\_dataAnalysisWindow() | 이용자의 추천을 위한 데이터 분석 동의 여부 파악 후에 dataAnalysisWindow를 불러들임 |
| get\_analysusData() | 분석용 데이터 불러오기 |
| update\_analysisData() | 분석용 데이터 수정 |
| get\_recentDealData() | 최근 거래 내역 데이터 불러오기 |
| get\_recentCategory() | 최근 거래 내역의 서비스 항목 불러오기 |
| get\_recendId() | 최근 거래 내역의 id 불러오기 |
| do\_analysis\_apriori() | Apriori package의 연관성 분석 시행 |
| do\_analysis\_fpGrowth() | FPgrowth 분석 시행 |
| do\_analysis\_cf\_item() | Collaborative filtering (item 기반) 분석 시행 |

E.2.3.3 pseudo code (R 환경 기반)

###apriori 분석

#install.packages("arules")

library(arules)

library(dplyr)

# transaction 형태로 가져오기 # long을 r environment로 가져와서 형태 변환

model\_for\_transaction<-function(){

 model\_for\_split <- split(서비스 항목 데이터 features, 해당 서비스 이용자)

//transaction 형태로 데이터 분리 위해, 서비스 항목 데이터 features, 이용자 분리

 linkorea\_transaction<-as(model\_for\_split,"transactions") // transaction으로 변환

return(linkorea \_transaction)

}

#####

rules\_model<-function(linkorea \_transaction){

 rules=apriori(linkorea \_transaction, parameter=list(supp=0.1,conf=0.1))#parameter 조정

 rule.list[order(rule.list$lift, decreasing=TRUE), ] // 연관성 상위 값 sorting 만 보기

 #dbWriteQuery(associates,"file\_path",fileEncoding = "EUC-KR")

}

####

<Test Case>

Register System

Register 기능에 대한 기능은 다음과 같은 기능을 검사한다.

희망 서비스 등록시, 해당 정보들이 database에 모두 저장되었는지 test

* 필수 기재 항목을 빠트리고 입력했을 경우 경고 메시지를 띄울 것
* 성공적으로 회원 가입되었을 경우 성공 메시지를 띄울 것.

Search System

희망 서비스에서 등록한 세부 정보대로 filtering 되었는지 test

Recommend System

* 거리 계산 및 sorting 과정이 3초 이내로 computing 되는지 test
* Apriori 및 FPGrowth algorithm에 들어가는 data type 이 적절하고, 제대로 분석이 돌아가는지 test
* Apriori 및 FPGrowth algorithm computing 시간이 5초 이내로 걸리는지 test