

UNIVERSITY OF
CALGARY

**A Project Report for SENG 609.22
Agent Based Software Engineering
Course Instructor: Dr. Behrouz H. Far**

Multi-Agent System for Electronic Realtor

Date: Dec. 20, 2002

Table of Contents

Abstract.....	4
1 Introduction.....	5
2 System Overview.....	5
2.1 System Description.....	5
2.2 System Requirements.....	6
2.3 System Future Enhancement.....	6
2.4 Assumptions.....	7
2.5 System Hardware Specification.....	7
3.1 Account center agent – playing role as registrar.....	8
3.2 Buyer agent – playing role as buyer administrator.....	9
3.3 Seller Agent – play role as seller administrator.....	9
3.4 Banker Agent – play role as bank administrator.....	10
3.5 Negotiation Agent – play role as negotiator.....	10
3.6 Transaction Agent – play role as transactor.....	10
3.7 DB Agent – play role as DBA.....	11
3.8 Agent Internal Architecture.....	11
4 Detail Design.....	12
4.1.2 Use Case: Buyer Agent.....	14
4.1.3 Use case: Seller Agent.....	16
4.1.4 Use Case: Banker Agent.....	18
4.1.5 Use Case: Negotiation Agent.....	19
4.1.7 Use Case: DB Agent.....	22
4.2 Class and Sequence Diagrams.....	25
4.2.1 FrameWork.....	25
4.2.2 Account Center Agent.....	26
4.2.3 Buyer Agent.....	28
4.2.4 Seller Agent.....	29
4.2.5 Banker Agent.....	31
4.2.6 Negotiation Agent.....	32
4.2.7 Transaction Agent.....	33
4.2.8 DB Agent.....	34
5.1 Database Schema.....	35
5.2 Detail Data Elements Definition.....	35
6 Electronic Realtor Inter-Agents Messages.....	41
6.1 Login.....	41
6.2 CreateNewAccount.....	41
6.3 checkExpiredAd.....	42
6.4 queryAd for house.....	42
6.5 postAd for house.....	42
6.6 Offer for house.....	43
6.7 Schedule for meeting.....	43
6.8 postBankRate.....	44
6.9 OpenTransaction.....	44

6.10 closeTransaction 44
7 Conclusions..... 46
 7.1 Experience of the project 46
 7.2 Discussion 46
Reference.....47

Abstract

This document presents a design for the implementation of an Electronic Realtor (ER). The multi-agent design approach is used in this project. This ER allows users to sell/buy houses through Internet.

The design present in this document includes the agent model of the overall system, and detailed design of each agent. The agent's detailed design is captured using Unified Modeling Language (UML).

As discussed in the detailed design section of this document, the ER system is to be implemented using the Simple Object Access Protocol (SOAP) for inter-agent communication, and JAVA as the implementation language.

1 Introduction

Traditionally, property trading is to be conducted through the realtor agent. Home buyer usually has to go through a realtor for a list of houses that are put for sale on the market. This can be quite inconvenient because the buyer will have to contact the realtor every time he/she wants to get the updated list of the houses for sale. Other sources that homebuyer can get the house information are newspaper and magazine. But, sometimes, these sources are not complete or not up-to-date because they are only being distributed or published periodically (e.g., weekly or monthly) And it can be very time consuming to go over the whole list every time you get the updated list from the realtor or other sources. In addition, the commission charged by the realtor for selling the house can be quite high; it can be as much as up to 7% of the house value.

The house buying and selling can be greatly simplified if there is a system in the Internet that allows users to advertise the houses for sale or putting advertisement for buying. The users can log into the system anytime to look for houses to buy and to put advertisement to sell his/her house. The system can help the buyers to shorten the house list by filtering out advertisements that don't meet the criteria specified. It can also be setup to have the system automatically notify the buyers when there is a new advertisement posted that meet the buyer's requirement. And there is no Realtor commission since the house can be sold directly by the owner.

In this document, a design of an electronic realtor application is discussed. The objective of this application is to simulate a virtual electronic marketplace where users can buy and sell houses. The goals of this application are twofold:

- To allow users to handle a lot of the upfront management of buying or selling a house from the convenience of his/her house; and
- To reduce the overhead cost of selling the house.

The latter goal is realized by eliminating the use of a realtor agency; thus, eliminating the commission overhead cost.

2 System Overview

2.1 System Description

The Electronic Realtor (ER) application allows users to advertise house for sale and search for house to buy. In addition, bank can advertise their mortgage rate to potential house buyers via this application. This system is a multi-agent system and is accessible by users through Internet as shown in Figure 1. The activities that take place between users will be logged and stored onto a central database. This record can later be retrieved and processed for billing purpose.

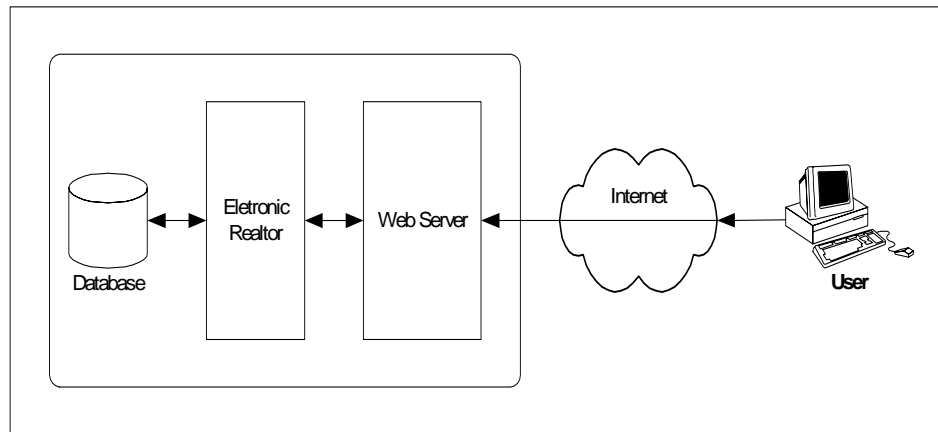


Figure 1 - System Overview

This system simulates a virtual electronic marketplace where users can conduct house trading. It utilizes Internet as a means of communication between buyers and sellers. It provides the users utilities to schedule meeting with each other, and to negotiate price.

2.2 System Requirements

This ER application shall support the following features:

1. It allows users to:
 - a) Search and post house advertisements.
 - b) Schedule meeting to view the house or meet to negotiate for mortgage rate.
 - c) Negotiate price on the house between the buyer and seller.
2. The ER users can be house buyer, house seller or banker. Each of these users will have different access privileges to the system:
 - a) Buyer – can query house advertisements; look for bank that offers the best mortgage rate and setup appointment with the house buyer or banker.
 - b) Seller – can query/post house advertisements and setup appointment with the house buyer.
 - c) Banker – can post mortgage rate and setup appointment with house buyer.
3. It maintains a user profile that includes user's contact information and resident information.

2.3 System Future Enhancement

This document shall not deal with the following features. But these features should be considered as parts of feature enhancements for future release:

1. To provide legal assistance for handling land title transfer. That is, it can provide step-by-step guidance to complete all the legal documentation as required to transfer the ownership of house from the seller to the buyer.

2. To provide an economic analysis for user who is looking to buy a house. For instance, depending on the buyer's income and family size, this application can provide suggestions as what type of house is suitable for him/her.
3. To provide a billing capability to bill the users for using the service of this system.

2.4 Assumptions

The security issue will depend on the deployment of the application. If the agents are to be deployed in a closed environment, then, the security of the communication channels between the agents can be relaxed. However, if the agents are to be connected via Internet, the communication channels between the agents should be encrypted. Due to time constraints, the security issues are not addressed in this document. It is assumed that the web server has already configured and it supports encrypted protocol such as HTTPS. It is also assumed that a Database has already been selected and setup to be used by this system for information storage.

2.5 System Hardware Specification

Depending on the size of the users, each agent discussed in this document can be run on a separate machine such as SUN SparcServer or PC server. However, initially, a lot of these agents can be executed in one server.

3. System Architecture

According to the system's specification, the user of the application will have to login to the "Account Center Agent" before he/she can access to the services of the ER. This system acts as an electronic marketplace where buyer, seller, and banker can meet and conduct house trading.

First, the client logs into the system through the account center agent. The reason that we need an account center agent is because there are three different kinds of users to use this system, we need one agent to act as an account registrar to distinguish different users, authenticate a valid user as well as create a new account for a new user. Next we need different agents to manage different kinds of users, so buyer agent, seller agent, and banker agent will act as the buyer administrator, seller administrator, and banker administrator. After users successfully pass the account verification, they will go to different user agents.

In order to make different users communicate with each other, we need a negotiation agent to act as a negotiator to consult over price, mortgage rate and interview scheduling among the buyer, seller, and banker. Transaction agent's role is to keep track of all the transaction procedure. When the transaction is complete, transaction agent will tell DB

agent to modify database information. Of course, DB agent here acts as DB Administrator to manage all the information and provide appropriate information to the users.

The ER is composed of seven agents as shown in the Figure 2. Obviously, each agent in this system represents one role just as described above. Next each of these agents will be discussed in detail.

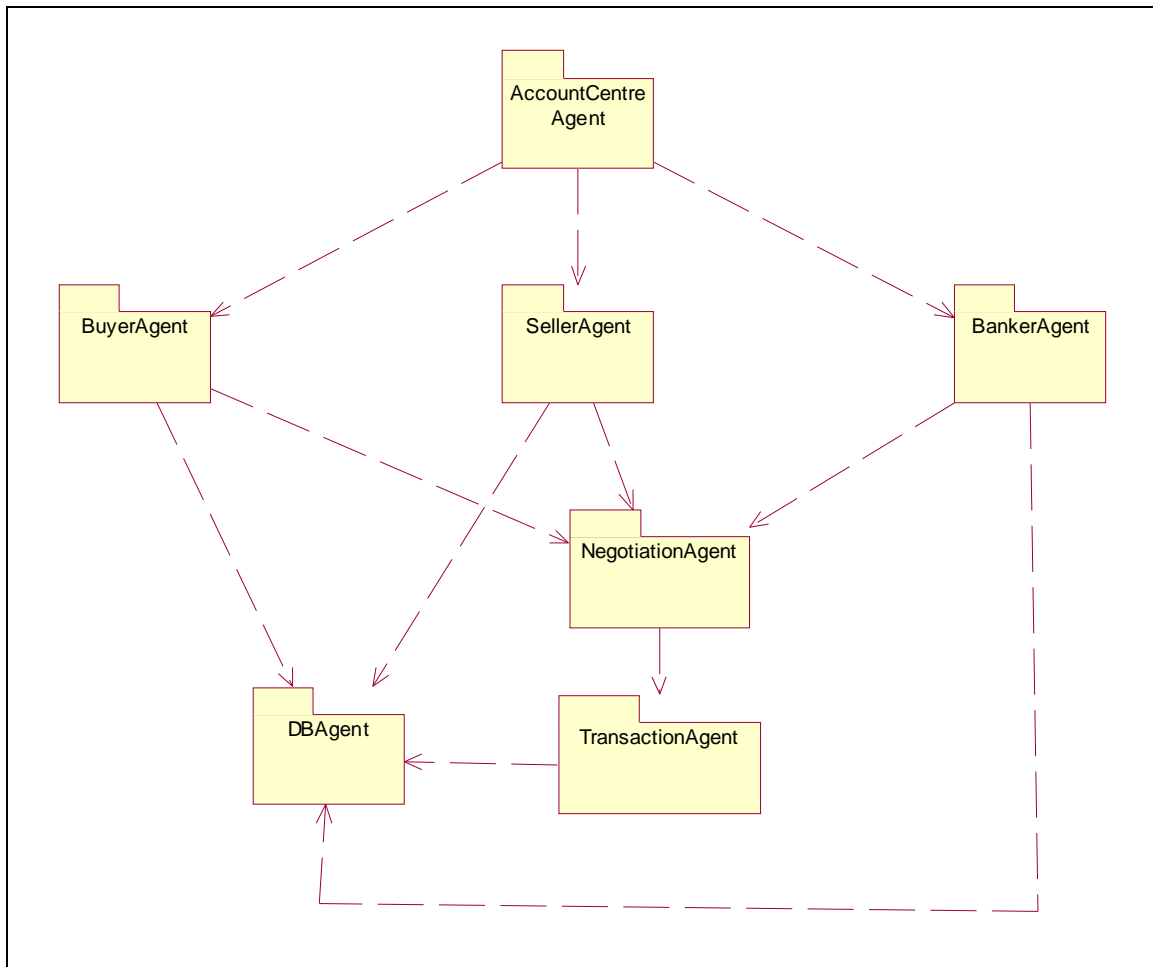


Figure 2 - Agent Model Overview

3.1 Account center agent – playing role as registrar

The account center agent interacts with DB agent to create new account for the new user and set up profile file for every user as well as validate the old user's account.

There are three account types in this account center: buyer account, seller account, and banker account. Each account type has different access privilege and billing cost. For

instance, the seller will be charged on a per house-posting basis. And the buyer will be charged on a monthly or yearly rate.

The account center agent also interacts with the buyer agent, the seller agent, and the banker agent, since the account center agent is responsible for leading the user into the appropriate agent; for example, the buyer goes to the buyer agent after he/she passes the account verification.

The account center agent is responsible for checking database daily through DB agent to remove inactive account and to notify the user whose Ad is going to expire.

3.2 Buyer agent – playing role as buyer administrator

The buyer agent needs to interact with other two agents: DB agent and Negotiation agent. The buyer agent allows the buyer to query house information and bank mortgage information through DB agent. DB agent will send the corresponding data from database to the buyer agent.

The buyer agent allows the buyer to post house purchase Ad and mortgage Ad, and then DB agent will save these Ads into database.

The buyer can schedule a meeting with the seller or the banker through the negotiation agent, and the negotiation agent will give the buyer seller's response or banker's.

The buyer can place an offer of the house or the mortgage through the negotiation agent, and the negotiation agent will feedback the seller's response or banker's.

3.3 Seller Agent – play role as seller administrator

The seller agent needs to interact with other two agents: DB agent and Negotiation agent.

The seller agent allows the seller to query house information and the buyer's Ad through DB Agent. DB agent will send the corresponding data from database to the seller agent.

The seller agent allows the seller to post house selling Ad, and then DB agent will save this Ad into database.

The seller can schedule a meeting with the buyer or give the response for the buyer's schedule through the negotiation agent, and then the negotiation agent will feedback the buyer's response.

The seller can review various buyer's offer and feedback the buyer's offer through the negotiation agent.

3.4 Banker Agent – play role as bank administrator

The banker agent needs to interact with other two agents: DB agent and Negotiation agent.

The banker agent allows various banks to post mortgage rate Ads through DB agent, and then the DB agent will save these Ads into database.

The banker agent allows banks to review buyer's Ad for mortgage rate through DB agent, and then the DB agent will send the corresponding data from database to the banker agent.

The bank can schedule the meeting with buyer or give response to the buyer's schedule through negotiation agent.

The bank can review the buyer's mortgage offer then response the buyer's offer through the negotiation agent.

3.5 Negotiation Agent – play role as negotiator

The negotiation agent collaborates with other four agents: buyer agent, seller agent, banker agent, and the transaction agent.

The negotiation agent negotiates the price between the buyer agent and the seller agent, the mortgage rate between the buyer agent and the banker agent.

The negotiation agent consults the meeting schedule between the buyer and the seller or the bank.

The negotiation agent will send every activity record to the transaction agent.

3.6 Transaction Agent – play role as transactor

The transaction agent interacts with other two agents: negotiation agent and DB agent. As one transaction starts, Transaction Agent tells the DB Agent to open a temporary table in the ER database to record the transaction process.

This temporary table shall be updated dynamically according to transaction process through Transaction Agent.

As one transaction ends, Transaction tells the DB Agent to update ER database according to the temporary table, and then delete the temporary table.

3.7 DB Agent – play role as DBA

The DB agent interacts with other five agents: buyer agent, seller agent, banker agent, transaction agent and account center agent.

The DB agent allows the buyer agent, seller agent, banker agent and account center agent to check appropriate information.

The DB agent can modify the database information according to the activity record from the transaction agent.

The DB agent can receive the information from the buyer agent, seller agent, banker agent and account center agent, and then save the information into database.

3.8 Agent Internal Architecture

Currently, there are a number of standards to handle inter-agent communications. To name a few, there is COBRA, Jini and so on. The technology selected for this application is Simple Object Access Protocol (SOAP). Comparing to the other technologies or standards, SOAP is rather new. It uses XML as the communication language, and HTTP as the transmission protocol. Figure 3 depicts the communication link between two agents using SOAP.

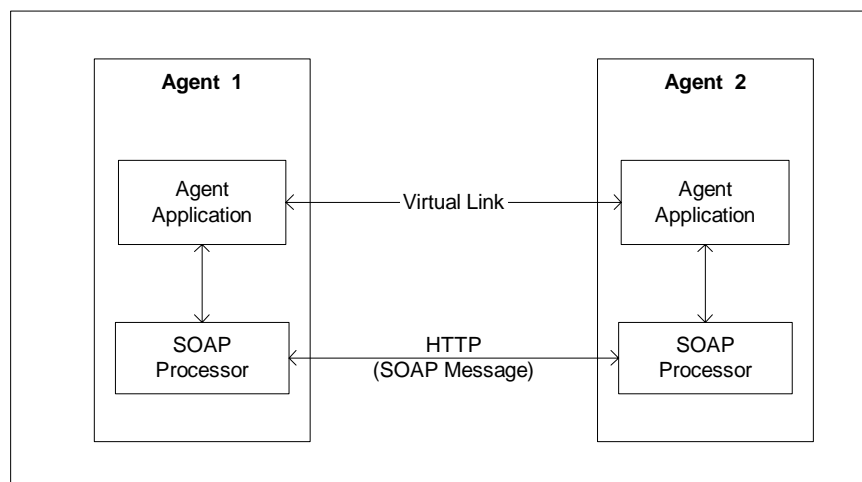


Figure 3 - SOAP Overview

The reason SOAP is chosen in this system is that SOAP is quite easy to use, and there are quite a lot of Open-Source products that support this technology. For instance, there are

products such as Apache SOAP and JDOM. Apache SOAP is an open-source implementation of the SOAP standard submitted to W3C (<http://xml.apache.org/soap/>). Whereas JDOM is a development utility with a set of API for reading, writing, and manipulating XML from within Java code (<http://www.jdom.org/>).

At startup, each agent will register itself to a “well-known” Directory facilitator. The facilitator will help the agents to locate each other in the network as shown in Figure 4.

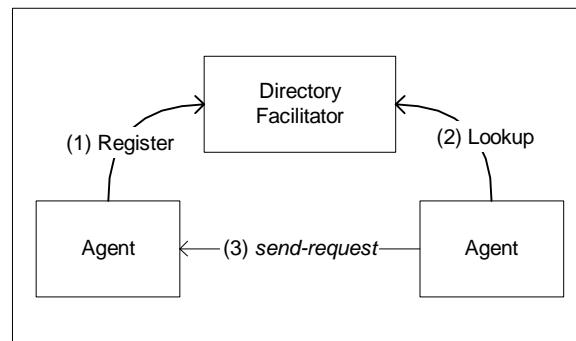


Figure 4 - Directory Service Overview

The design details of this framework will be discussed in the sections below.

4 Detail Design

In this part, we will give the detail design of the system. It includes the use case diagram, class diagram and sequence diagram for the system. The explanation related to each part will be given where it is relevant.

4.1 Use Case Diagram

Use case diagram will be given in this part, which includes seven use case diagrams.

4.1.1 Use Case: AccountCenter Agent

Use case diagram for AccountCenter Agent is illustrated in following figure 5:

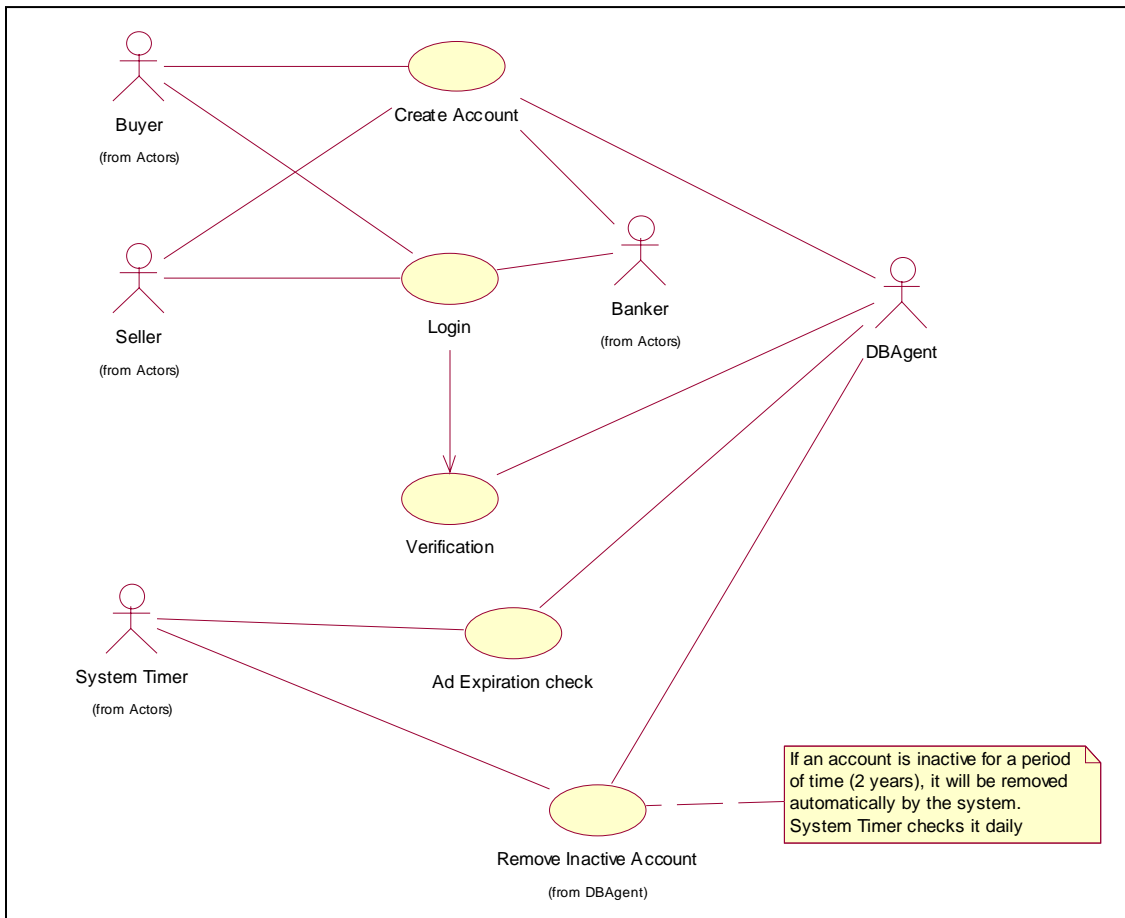


Figure 5 - Use Case Diagram for AccountCenter Agent

The detail use case definition for AccountCenter Agent is given in following Table 1:

Brief Description:	The actor uses this agent to get into ER system.
Precondition(s):	Old users' information has been stored in the database
Post condition(s):	Users go into next corresponding agent
Process Steps	
1.	The old user put into his/her login name and password through the interface
2	Accountcenter Agent goes to DB Agent to check the information, and then validate this user.
3	Accountcenter Agent leads the valid user into next corresponding agent
4	Accountcenter Agent creates new account for the new user and save new user's information into database through DB Agent
5	Accountcenter Agent daily checks database through DB Agent to remove inactive account if an account is inactive more than 2 years.
6	Accountcenter Agent daily checks database through DB Agent to

	check the expired Ad to notify the user who post this AD.
Exceptions	
3a	Accountcenter Agent rejects invalid user
Relationship	
Initiating	Actor (buyer, seller, banker)
Collaborating	DB Agent, Buyer Agent, Seller Agent, Banker Agent
Data Requirements	
Data Required:	Login name, password, username, Ad expiration, last login time

Table 1 - Detail Use Case Definition for AccountCenter Agent

4.1.2 Use Case: Buyer Agent

Use case diagram for Buyer Agent is illustrated in figure 6. Buyer Agent and Sell Agent are very important agents for the system, a lot of events involve in these two agents.

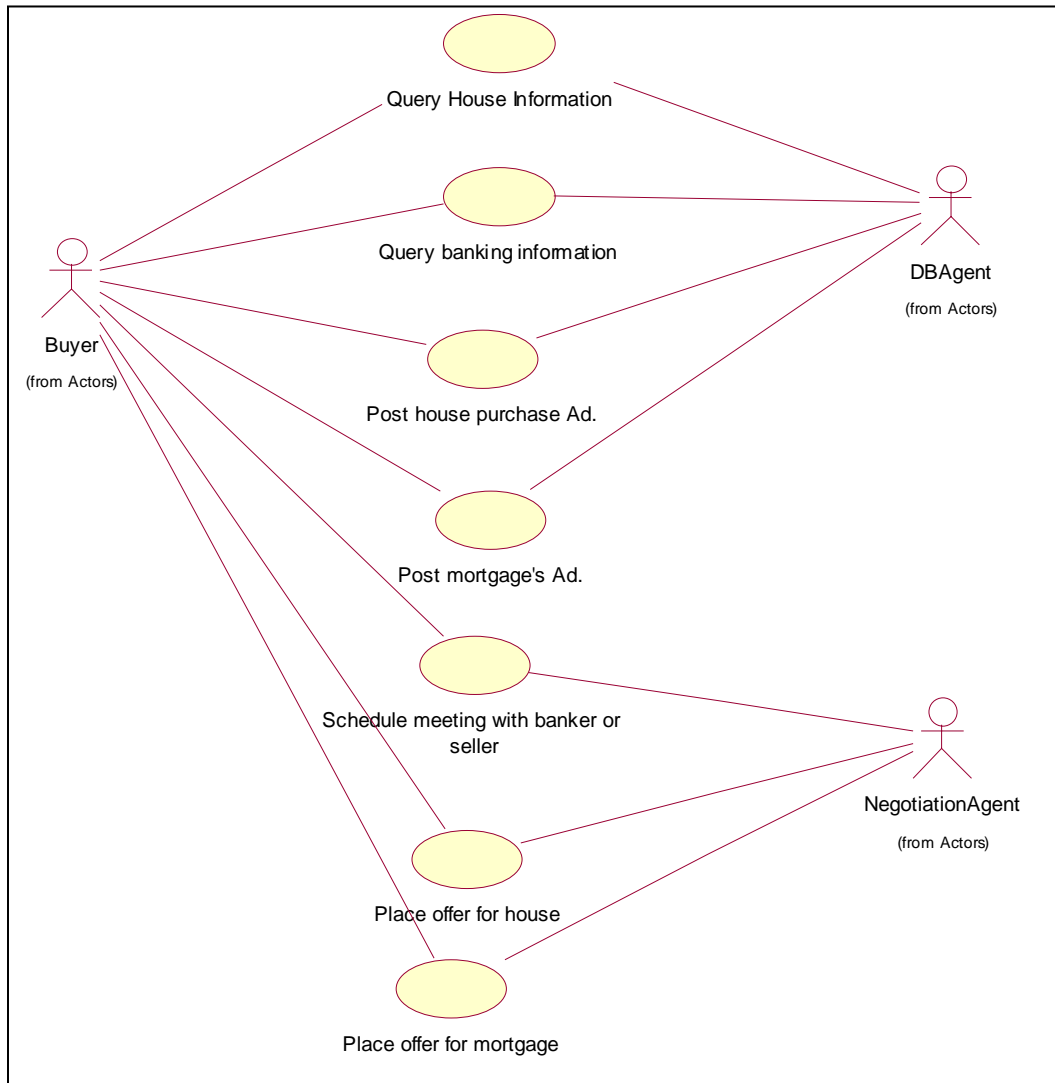


Figure 6 - Use Case Diagram for Buyer Agent

The detail use case definition for Buyer Agent is listed in following Table 2:

Brief Description:	Buyer Agent provides all kinds of service to the buyers
Precondition(s):	Buyer must be a valid user
Post condition(s):	N/A
Process Steps	
1.	The buyer makes a request for checking house information. And Buyer Agent goes to DB Agent to get a list of appropriate information.
2	The buyer wants to post a house purchase Ad. And Buyer Agent

	sends this Ad to DB Agent to save this Ad information.
3	The buyer makes a request for checking bank information. And Buyer Agent goes to DB Agent to get a list of appropriate information.
4	The buyer wants to post a mortgage Ad. And Buyer Agent sends this Ad to DB Agent to save this Ad information.
5	The buyer initiates a meeting with the seller or the bank or gives response to the other party's meeting schedule through Negotiation Agent.
6	The buyer places an offer for house or gives response to the seller's feedback through Negotiation Agent.
7	The buyer places an offer for mortgage or gives response to the banker's feedback through Negotiation Agent.
Exceptions	
	Error message is generated stating that system is not accessible at this point in case of that database is not accessible
Relationship	
Initiating	Actor (buyer)
Collaborating	DB Agent, Negotiation Agent
Data Requirements	
Data Required:	House information, seller Ad information, bank mortgage information

Table 2 - Detail View Use Case Definition for Buyer Agent

4.1.3 Use case: Seller Agent

Use case diagram for Seller Agent is illustrated in following Figure 7:

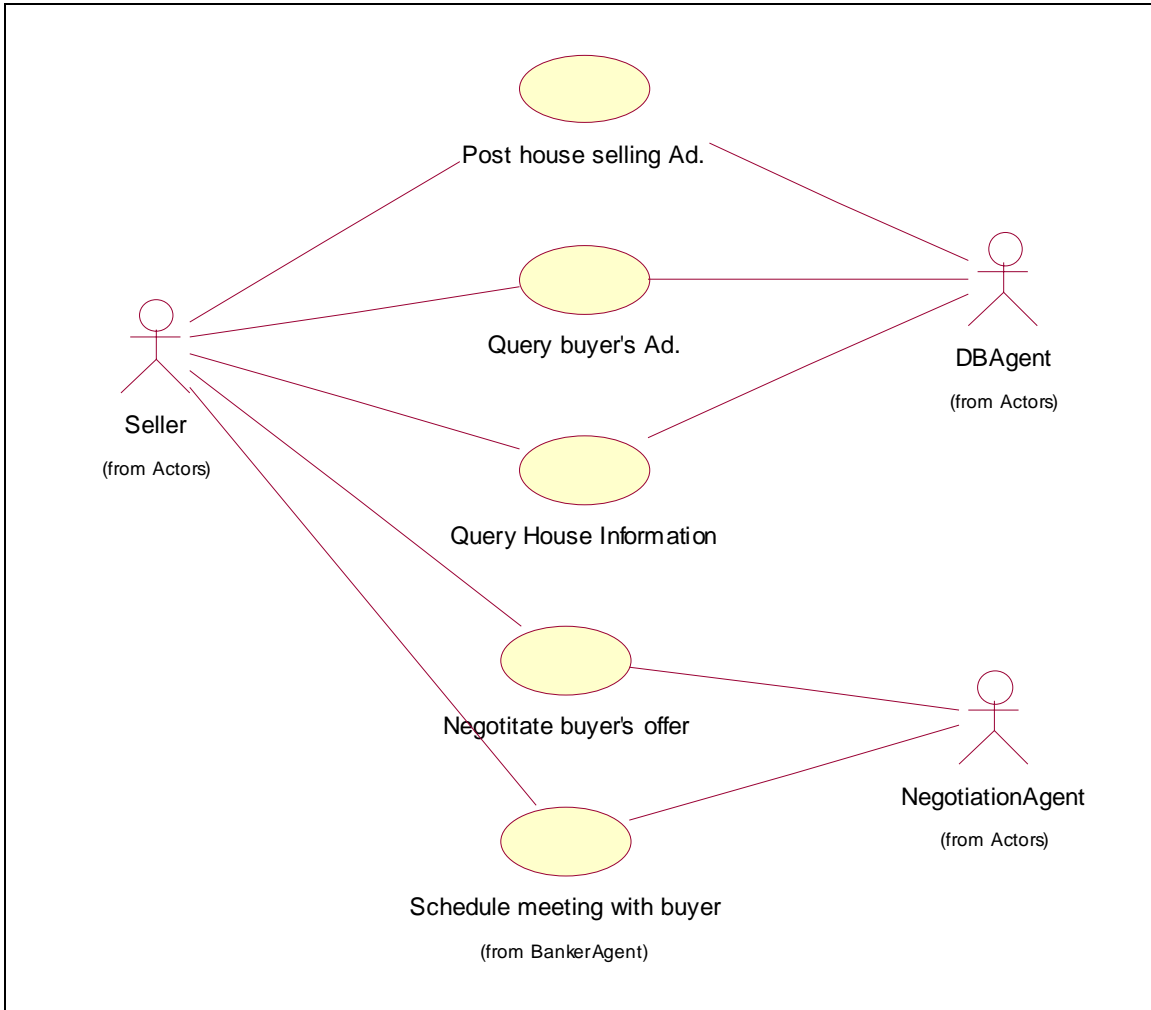


Figure 7 - Use Case Diagram for Seller Agent

The detail use case definition for Seller Agent is listed at following Table 3:

Brief Description:	Seller Agent provides all kinds of service to the sellers
Precondition(s):	Seller must be a valid user
Post condition(s):	N/A
Process Steps	
1.	The seller makes a request for checking house information. And Seller Agent goes to DB Agent to get a list of appropriate information.
2	The seller wants to post a house selling Ad. And Seller Agent sends this Ad to DB Agent to save this Ad information.
3	The seller makes a request for checking buyer’s Ad. And Seller Agent goes to DB Agent to get a list of Ad information.
4	The seller initiates a meeting with the buyer or gives response to the buyer’s meeting schedule through Negotiation Agent.

5	The seller reviews buyer’s offer and gives response to the buyer’s offer though Negotiation Agent
Exceptions	
	Error message is generated stating that system is not accessible at this point in case of that database is not accessible
Relationship	
Initiating	Actor (seller)
Collaborating	DB Agent, Negotiation Agent
Data Requirements	
Data Required:	House information, buyer Ad information

Table 3 - Detail Use Case Definition for Seller Agent

4.1.4 Use Case: Banker Agent

Use case design for Banker Agent is illustrated in following figure 8:

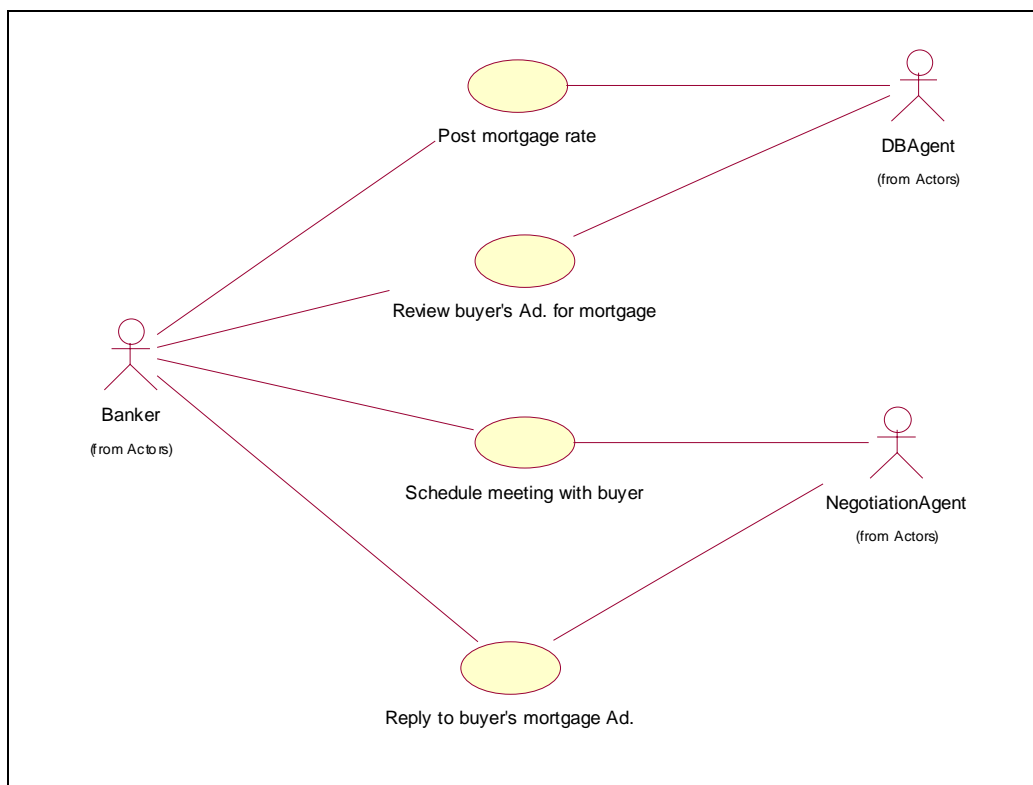


Figure 8 - Use Case Diagram for Banker Agent

The detail use case definition for Banker Agent is listed below Table 4:

Brief Description:	Banker Agent provides all kinds of service to the banks
Precondition(s):	Banker must be a valid user
Post condition(s):	N/A
Process Steps	
1.	The banker makes a request for checking buyer's Ad for mortgage. And Banker Agent goes to DB Agent to get a list of appropriate information.
2	The banker wants to post mortgage rate. And banker Agent sends this information to DB Agent to save it
3	The banker initiates a meeting with the buyer or gives response to the buyer's meeting schedule through Negotiation Agent.
4	The banker reviews buyer's mortgage offer and gives response to the buyer's offer through Negotiation Agent
Exceptions	
	Error message is generated stating that system is not accessible at this point in case of that database is not accessible
Relationship	
Initiating	Actor (banker)
Collaborating	DB Agent, Negotiation Agent
Data Requirements	
Data Required:	Mortgage information, buyer Ad information

Table 4 - Detailed Used Case Definition for Banker Agent

4.1.5 Use Case: Negotiation Agent

Use case for Negotiation Agent is illustrated in following Figure 9:

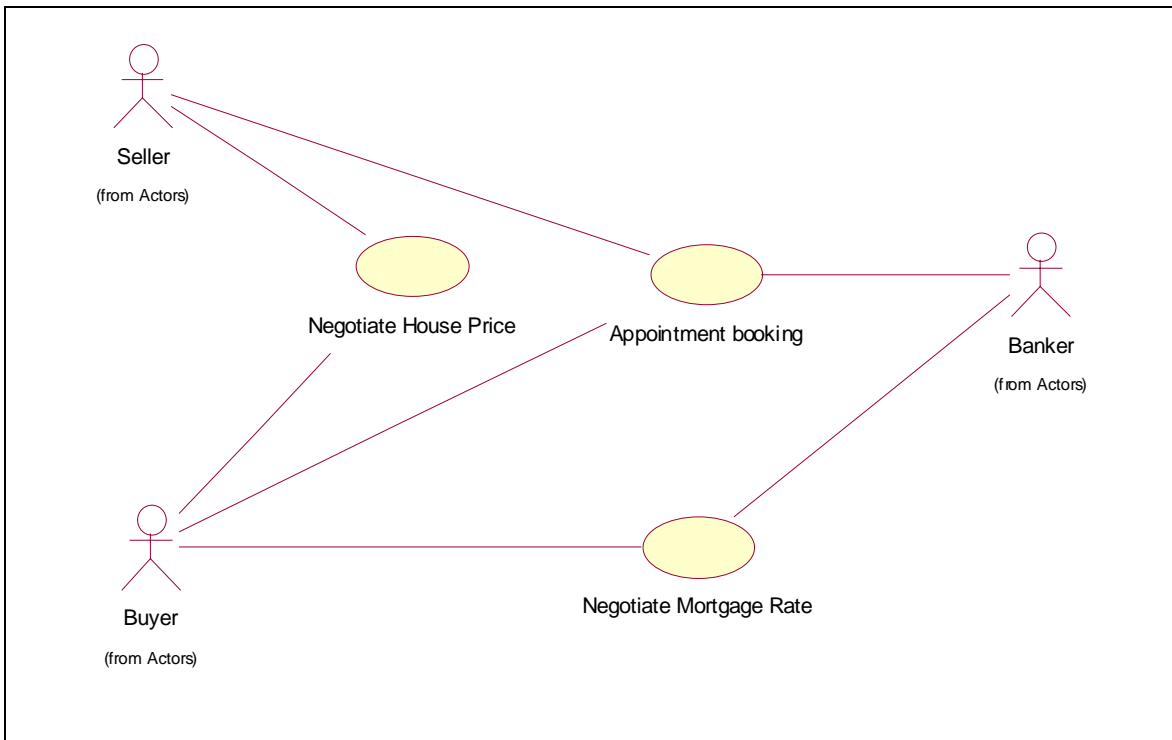


Figure 9 - Use Case Diagram for Negotiation Agent

The detail use case definition for Negotiation Agent is listed in following Table 5:

Brief Description:	Negotiation Agent provides service to let buyer, seller and banker communicate with each other.
Precondition(s):	Negotiation Agent is always on waiting status to be triggered by buyer, seller and banker
Post condition(s):	N/A
Process Steps	
1.	Negotiation Agent gets meeting schedule invitation from one side and then transfers this invitation to the other side. Negotiation Agent waits for the other side’s response, and then feeds back this response.
2	Negotiation Agent gets house offer from the Buyer Agent, and then sends this offer to the Seller Agent. Negotiation Agent waits for the seller’s response, and then feeds back this response.
3	Negotiation Agent gets mortgage offer from the Buyer Agent, and then sends this offer to the Banker Agent. Negotiation Agents waits for the bank’s response, and then feeds back this response.
4	Negotiation Agent sends records for negotiation procedure to Transaction Agent.
Exceptions	
2a	Negotiation failed message is generated in case of no response from

	other side.
Relationship	
Initiating	Actor (buyer, seller and banker)
Collaborating	Buyer Agent, Seller Agent, Banker Agent and Transaction Agent
Data Requirements	
Data Required:	Time, place, offer price

Table 5 - Detail Use Case Definition for Negotiation Agent

4.1.6 Use Case: Transaction Agent

Use case for Transaction Agent is illustrated in following Figure 10:

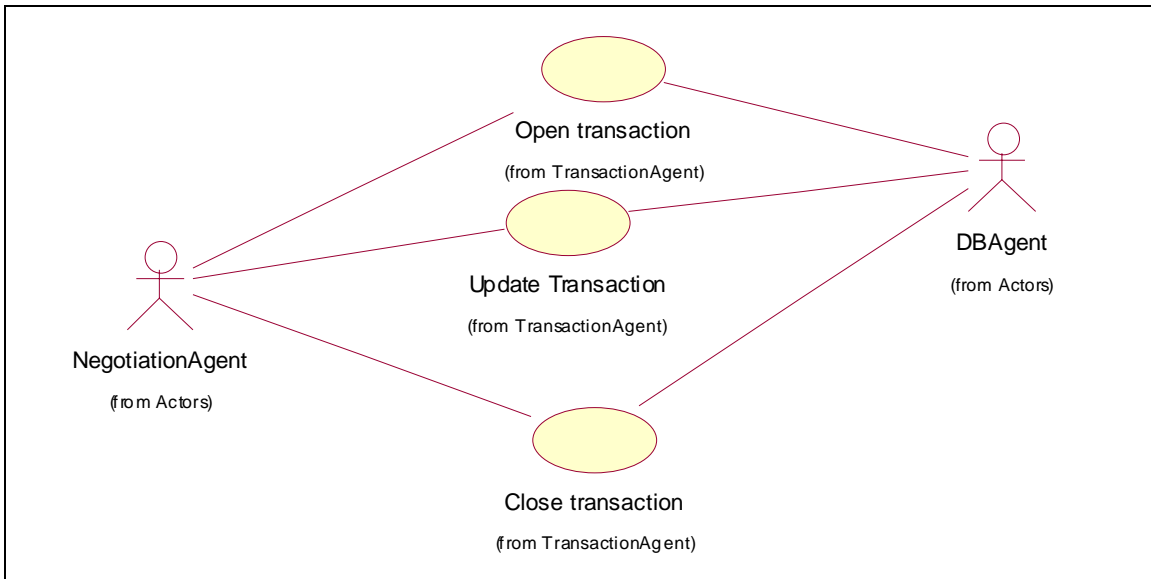


Figure 10 - Use Case Diagram for Transaction Agent

The detail use case definition for Transaction Agent is listed in following Table 6:

Brief Description:	Transaction Agent keeps trace of all the transaction process.
Precondition(s):	Connection with Negotiation Agent and DB Agent
Post condition(s):	N/A
Process Steps	
1.	Any negotiation event between buyer, seller and banker must be passed to transaction agent.
2	As one transaction starts, Transaction Agent tells the DB Agent to open a temporal table in the ER database to record the transaction

	process.
3	This temporary table shall be updated dynamically according to transaction process through Transaction Agent.
4	As one transaction ends, Transaction tells the DB Agent to update ER database according to the temporary table, and then delete the temporary table.
Exceptions	
	Error message is generated stating that system is not accessible at this point in case of that database is not accessible
Relationship	
Initiating	Actor (Negotiation Agent)
Collaborating	Negotiation Agent, DB Agent
Data Requirements	
Data Required:	Negotiation quotation(seller, buyer, price, date)

Table 6 - Detailed Use Case Definition for Transaction Agent

4.1.7 Use Case: DB Agent

Use case for DB Agent is illustrated in following Figure 11

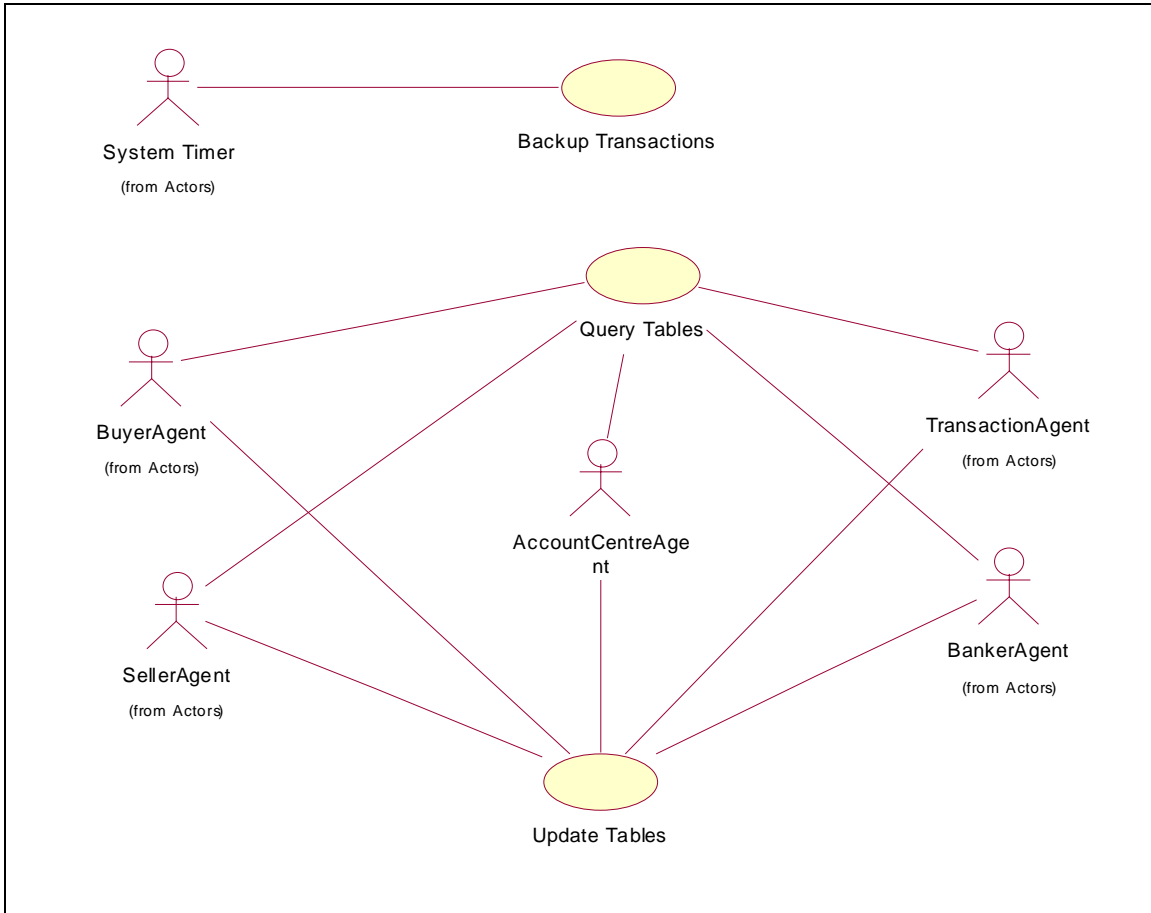


Figure 11 - Use Case Diagram for DB Agent

The detail use case definition for DB Agent is listed in following Table 7:

Brief Description:	Database Agent manipulate ER database (create, update delete tables).
Precondition(s):	ER database has already been created
Post condition(s):	N/A
Process Steps	
1.	The DB agent performs query command (for old user) and update command (for new user) from account center agent.
2.	The DB agent performs query and update operation against database according to buyer, seller and banker agent request to show buyer, seller, banker information or post house sell, buy, mortgage AD.
3.	The DB communicates with transaction agent to keep recording all the transaction in process negotiation.
Exceptions	
	N/A

Relationship	
Initiating	Actor (buyer, seller and banker, transactor and register)
Collaborating	Accountcenter Agent, Buyer Agent, Seller Agent, Banker Agent and Transaction Agent
Data Requirements	
Data Required:	Dataset of the whole ER database

Table 7 - Detailed Use Case Definition for DB Agents

4.2 Class and Sequence Diagrams

4.2.1 Framework

Since each agent will be using a similar mythology to communicate to each other, the “ServiceAgent” class is to be inherited by all the agents being developed in this system. Upon the startup of the agent, this class will automatically register itself to the “FacilitatorAgent” as shown in Figure 12. The “FacilitatorAgent” is responsible for helping agents to locate each other in the network.

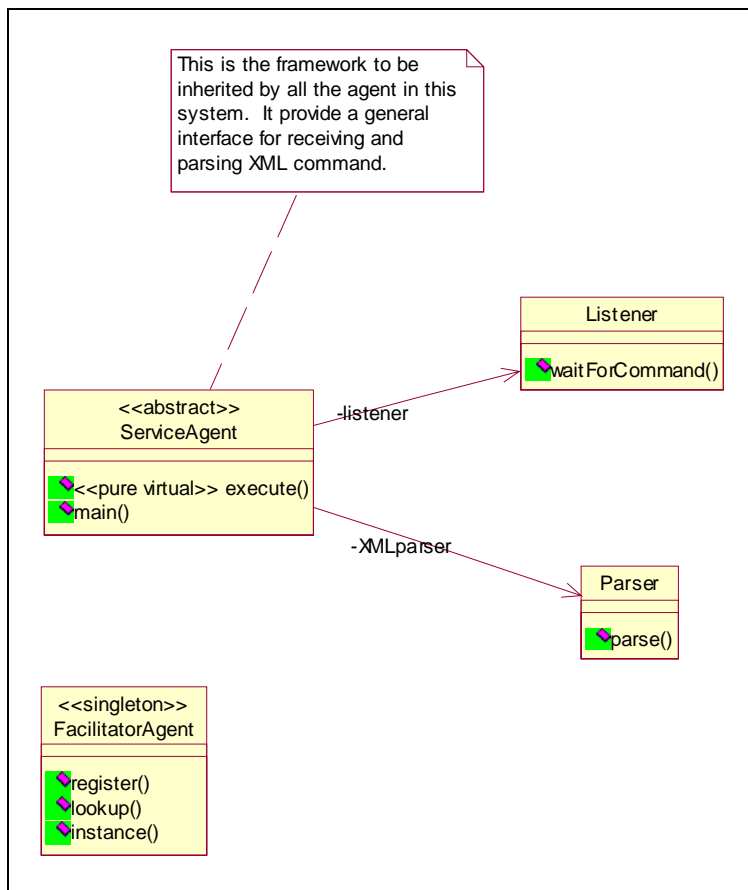


Figure 12 - Framework Class Diagram

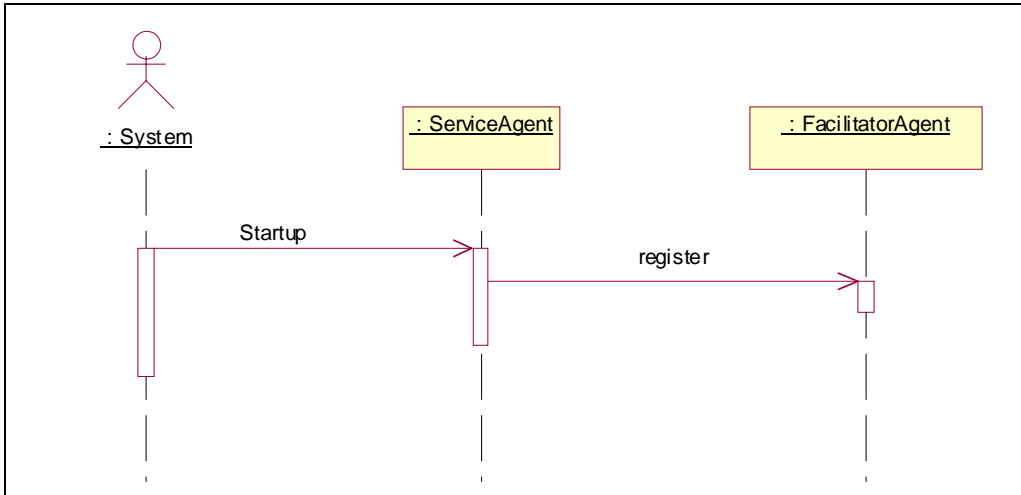


Figure 13 - Agent Bootup Sequence Diagram

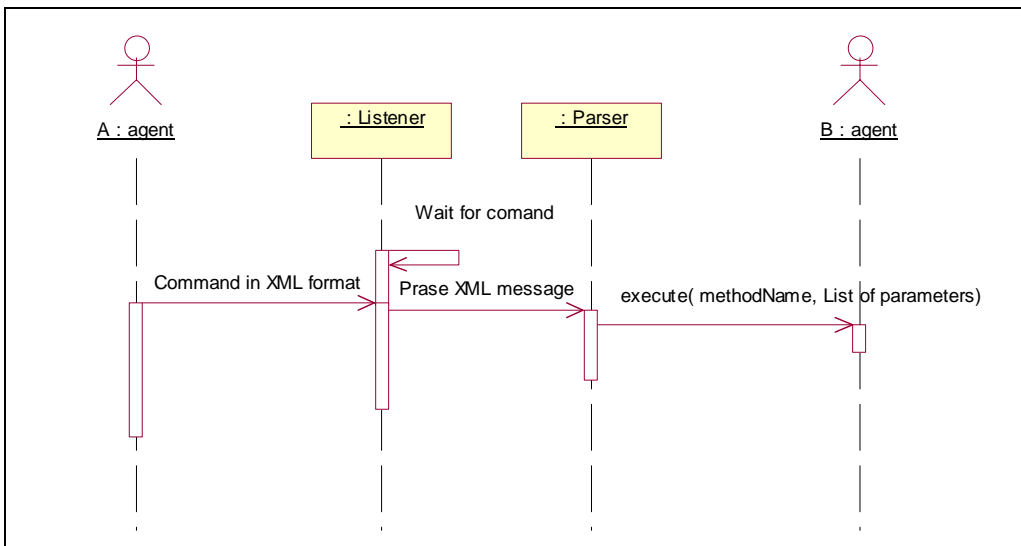


Figure 14 - Process Incoming XML Request Sequence Diagram

4.2.2 Account Center Agent

The class diagram and the sequence diagram for Account Center Agent are illustrated in following Figure 15 and 16 respectively.

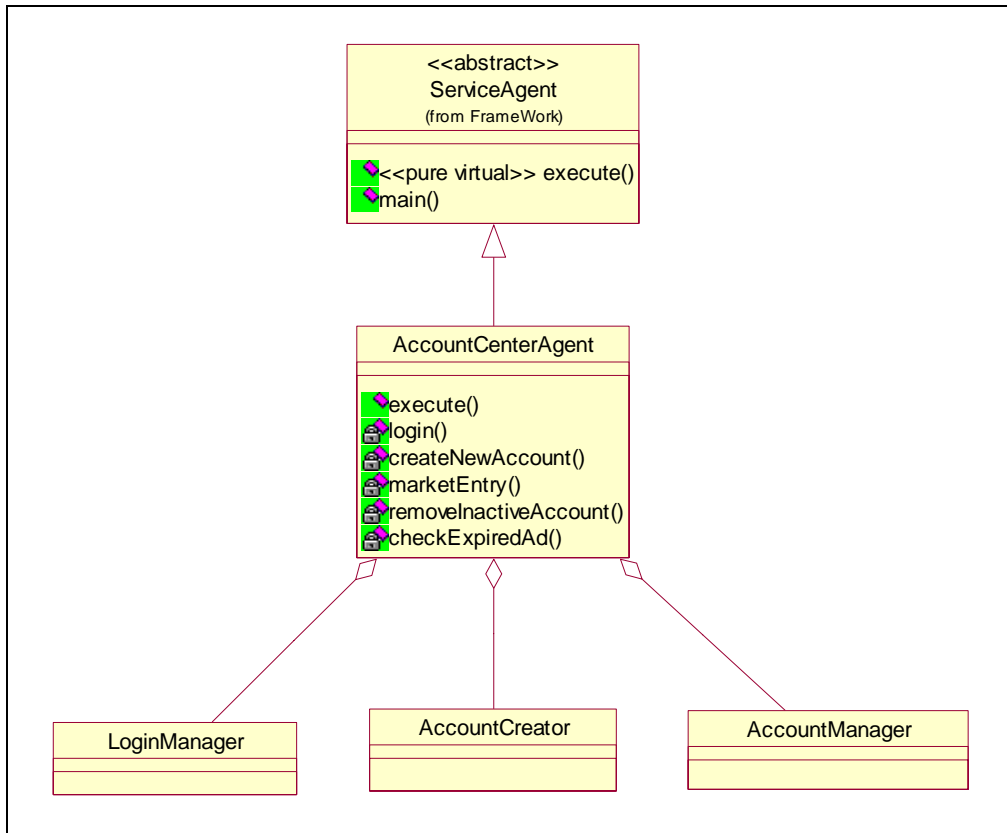


Figure 15 - AccountCenter Agent

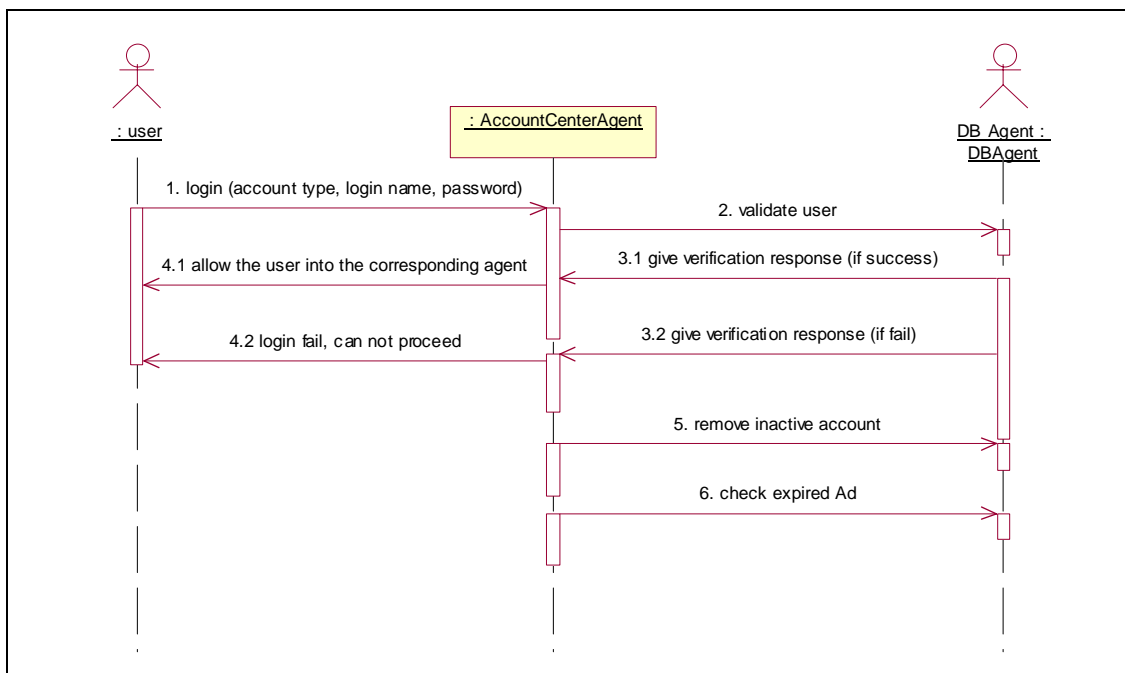


Figure 16 - Login Sequence Diagram

4.2.3 Buyer Agent

As shown in Figure 17, BuyerAgent provides interface to:

1. Query house advertisement,
2. Schedule meeting with seller or banker,
3. Place an offer on a house, and
4. Post advertisement.

Most of these functionalities, however, are delegated to other agents. For instance, as shown in Figure 18, the “placeOffer” function is to be completed by the NegotiationAgent.

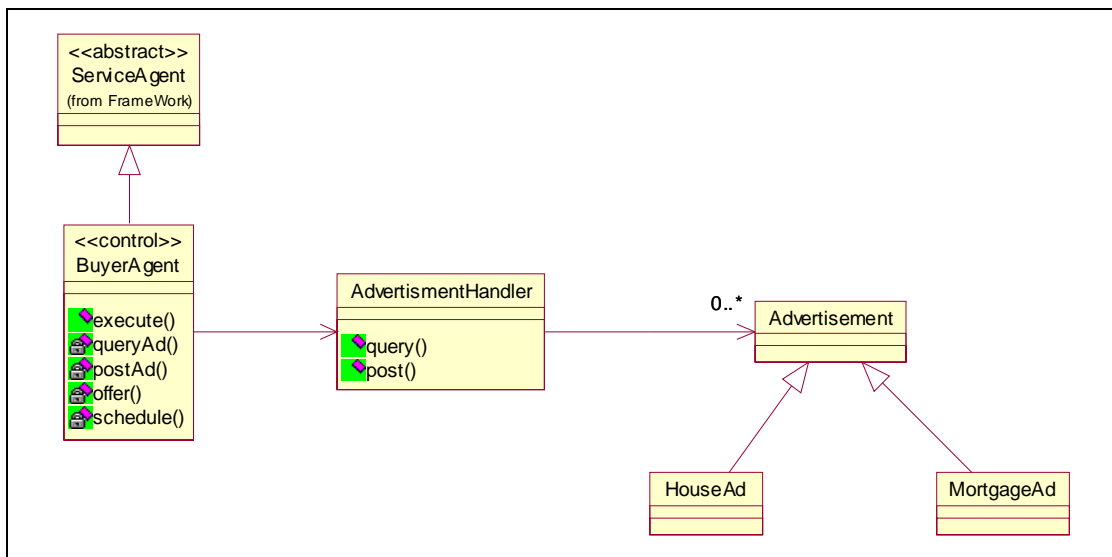


Figure 17 - Buyer Agent Class Diagram

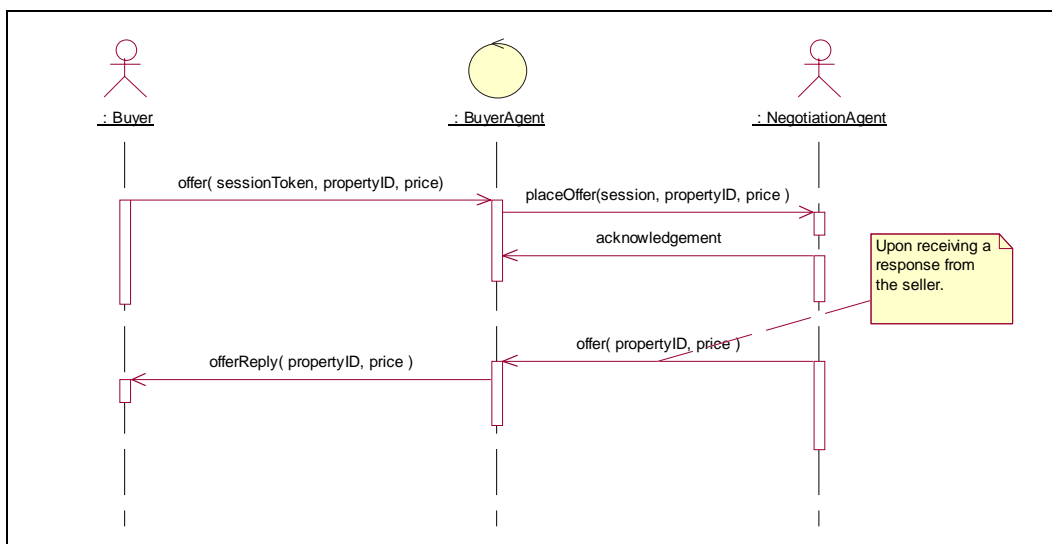


Figure 18 - Place Offer Sequence Diagram

Another two sequence diagrams for BuyerAgent are illustrated in Figure 19 and 20 respectively.

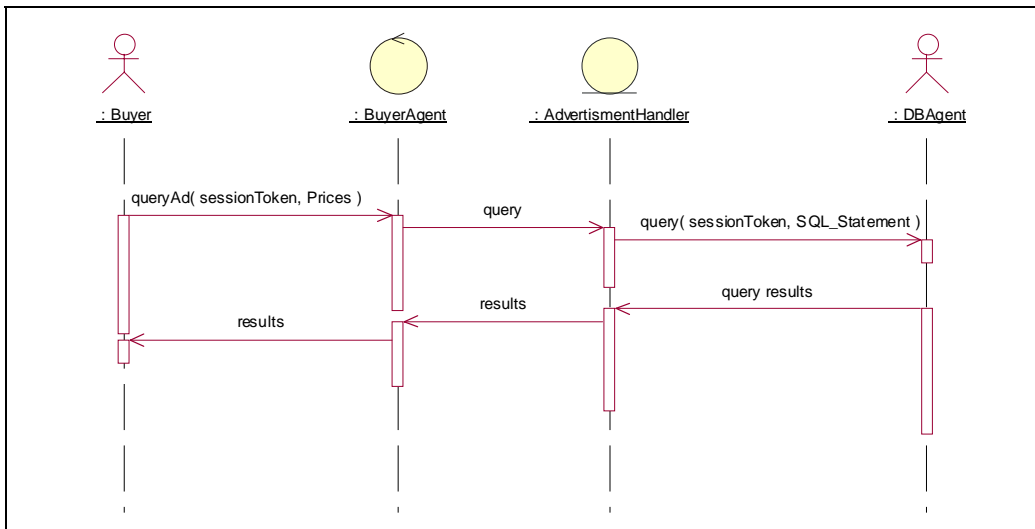


Figure 19 - Query Advertisement Sequence Diagram

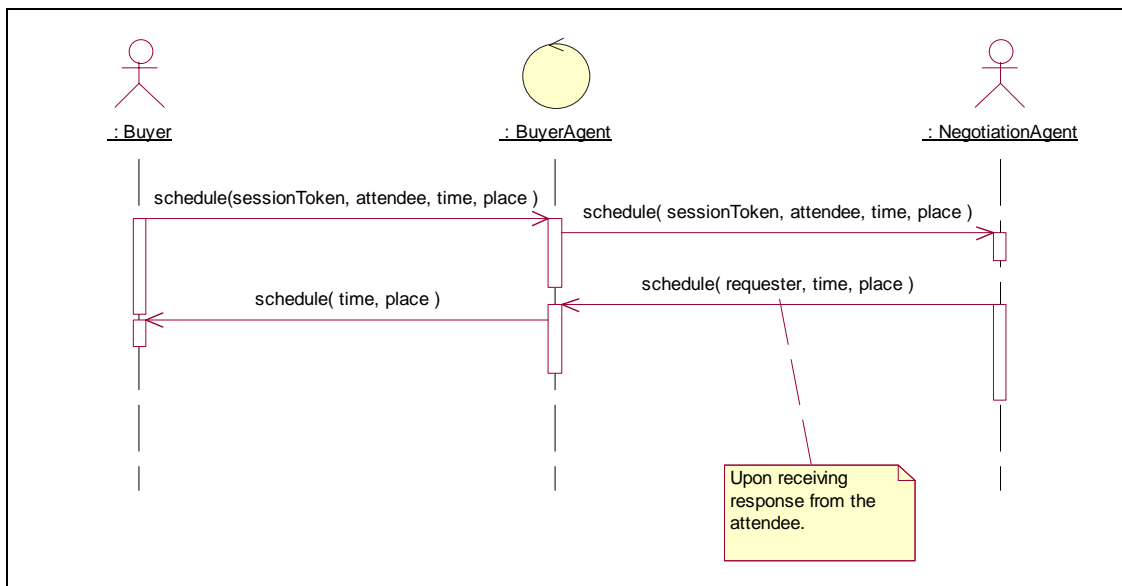


Figure 20 - Schedule Meeting Sequence Diagram

4.2.4 Seller Agent

The class diagram for seller Agent is illustrated in Figure 21. The three sequence diagrams for seller Agent are illustrated in Figure 22, 23 and 24, respectively.

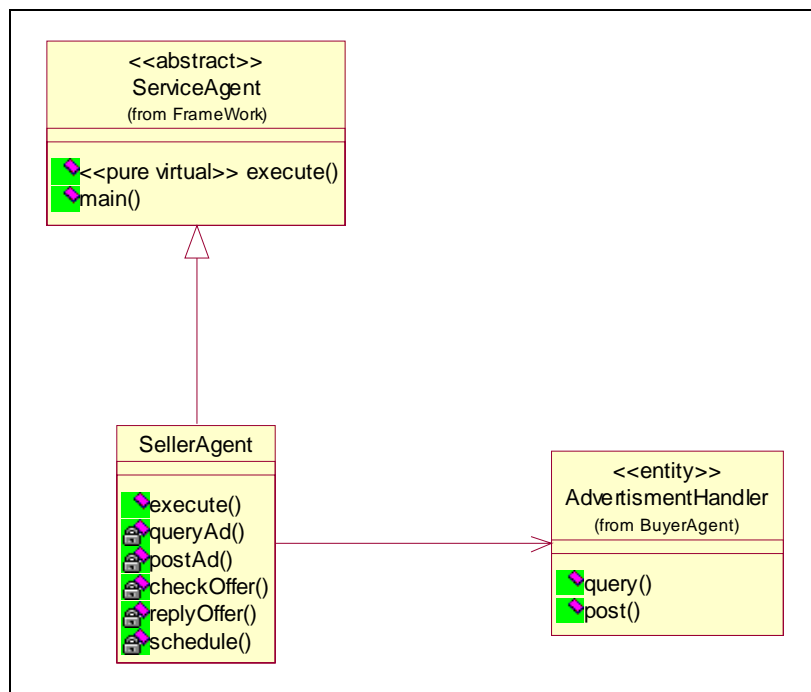


Figure 21 - Seller Agent Class Diagram

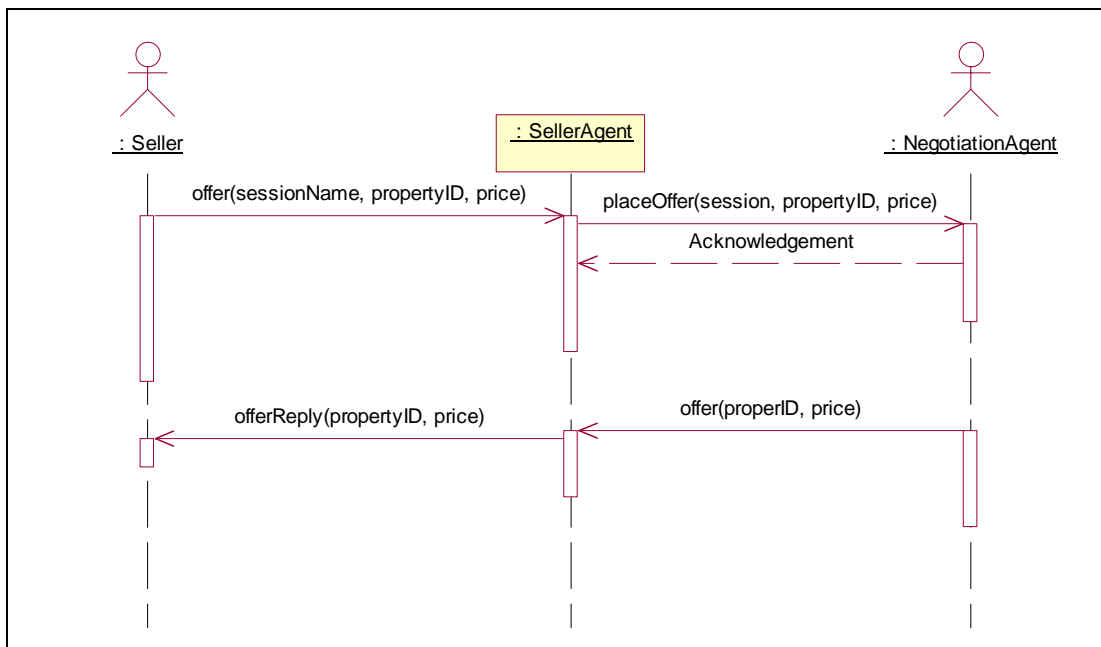


Figure 22 - Place Offer Sequence Diagram

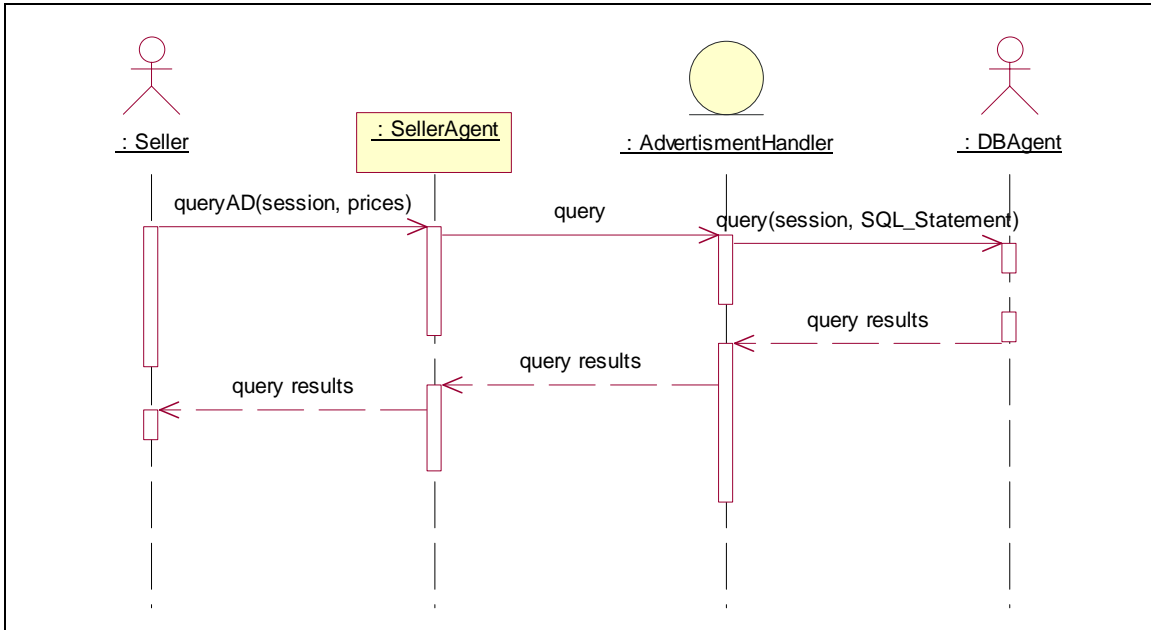


Figure 23 - Query Sequence Diagram

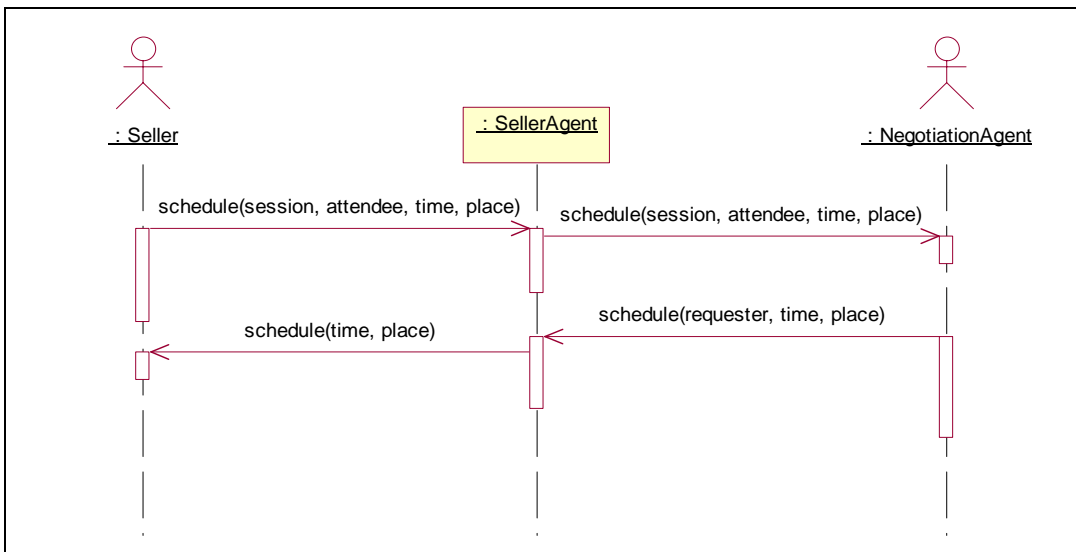


Figure 24 - Schedule Meeting Sequence Diagram

4.2.5 Banker Agent

Banker Agent class diagram and sequence diagram are illustrated in Figure 25 and 26 respectively.

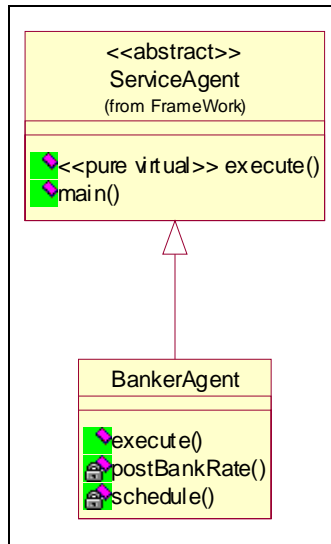


Figure 25 - Banker Agent Class Diagram

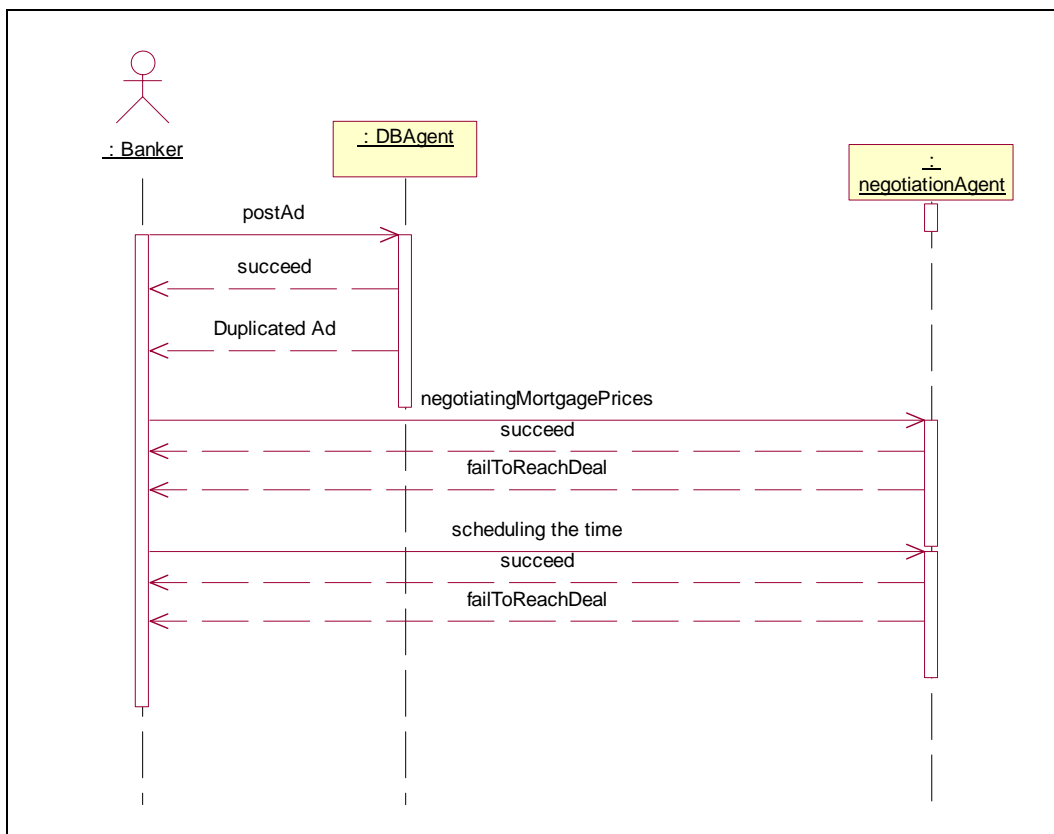


Figure 26 - Banker Agent Sequence Diagram

4.2.6 Negotiation Agent

Negotiation Agent class diagram is illustrated in Figure 27.

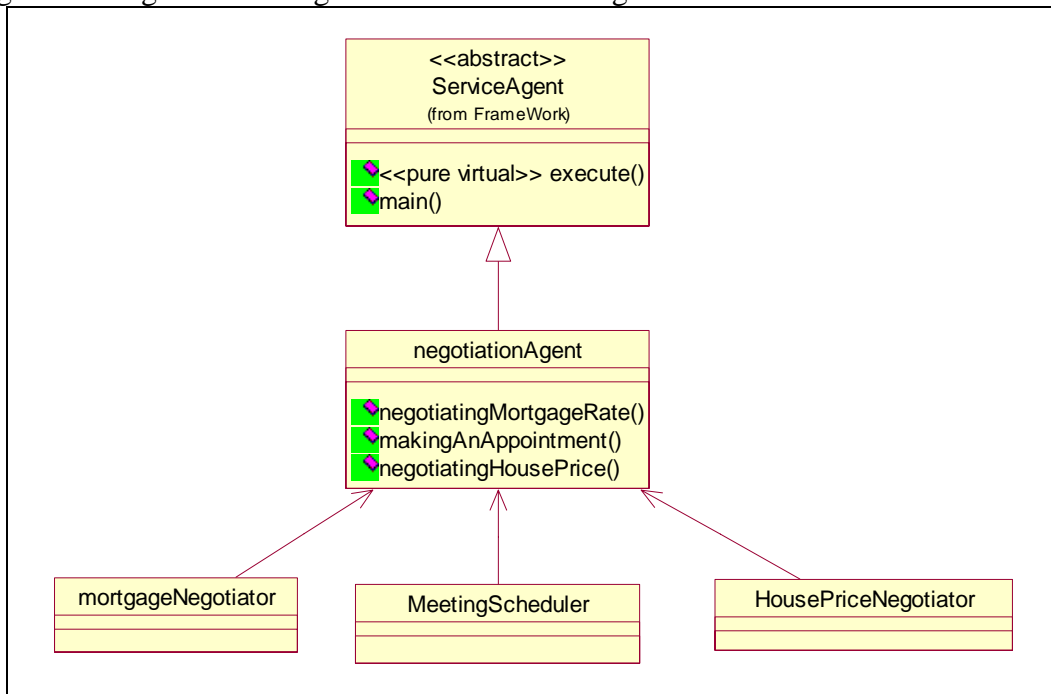


Figure 27 - Negotiation Class Diagram

4.2.7 Transaction Agent

Transaction Agent class and sequence diagrams are illustrated in Fig. 28 and 29 respectively.

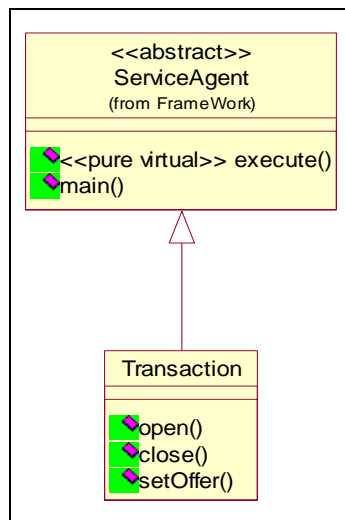


Figure 28 - Transaction Class Diagram

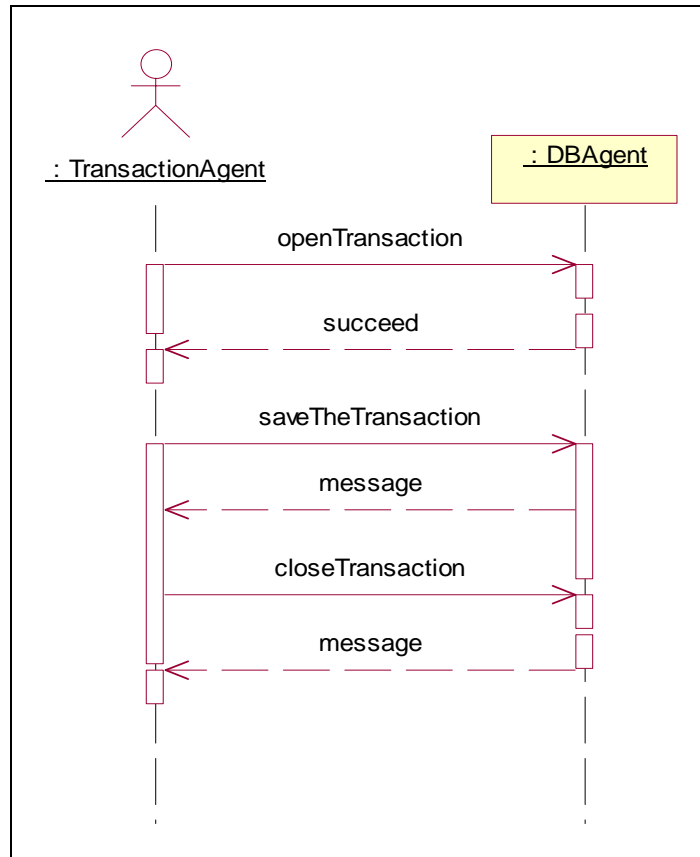


Figure 29 - Transaction Agent Sequence Diagram

4.2.8 DB Agent

DB Agent class diagram and sequence diagram are illustrated in Fig. 30 and 31 respectively.

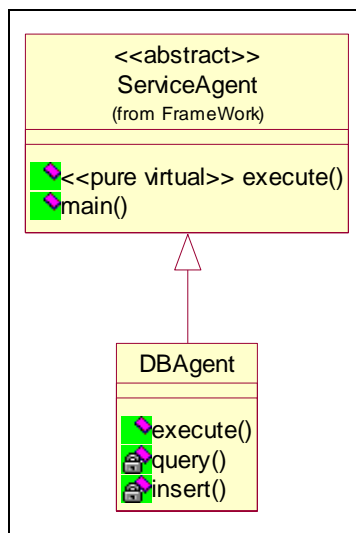


Figure 30 - DB Agent Class Diagram

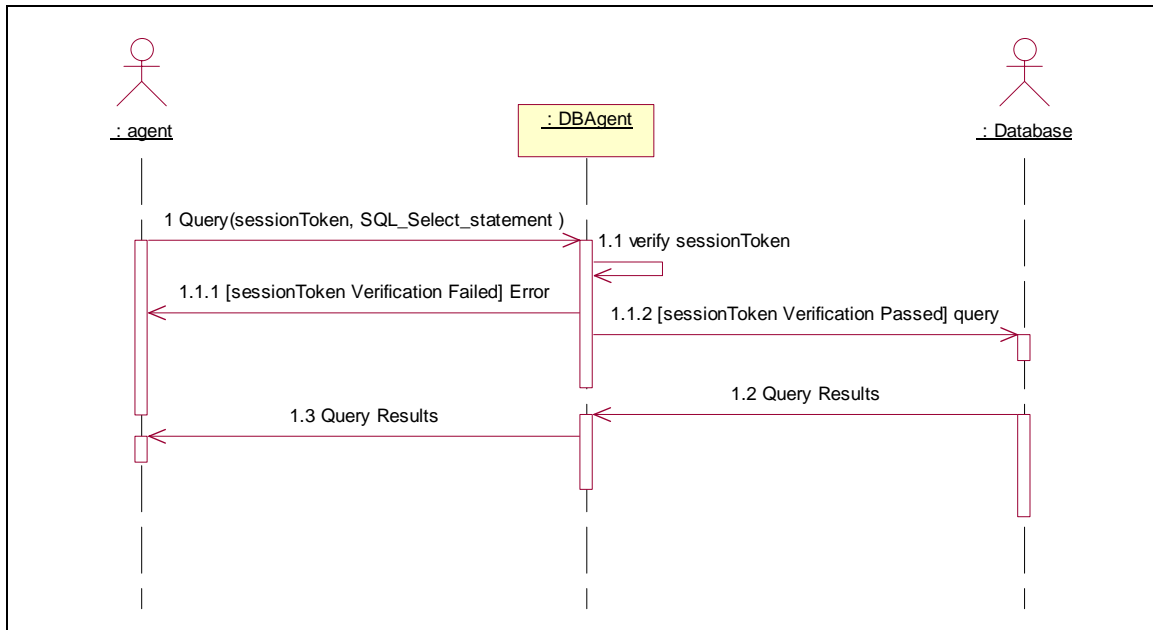


Figure 31 - DB Agent Sequence Diagram

5 Data Specification

In this part, detail data specification is given. It is composed of two parts. The first part is the definition of the database schema. In this schema, we define various tables and their Entity Relationship. In order to give a vivid picture of the system we adopt the data type of ANSI SQL 92, and it is transferable to any other data type when we implement it.

At the second parts, detail data elements are defined.

5.1 Database Schema

Database schema gives a vivid picture of the database structure of the system. In our system, we define seven tables. The data element, primary key and data element type of each are also defined in details. The detailed schema is illustrated in following Fig.32.

5.2 Detail Data Elements Definition

Detail data elements definition can be found from Table 8 to Table 14.

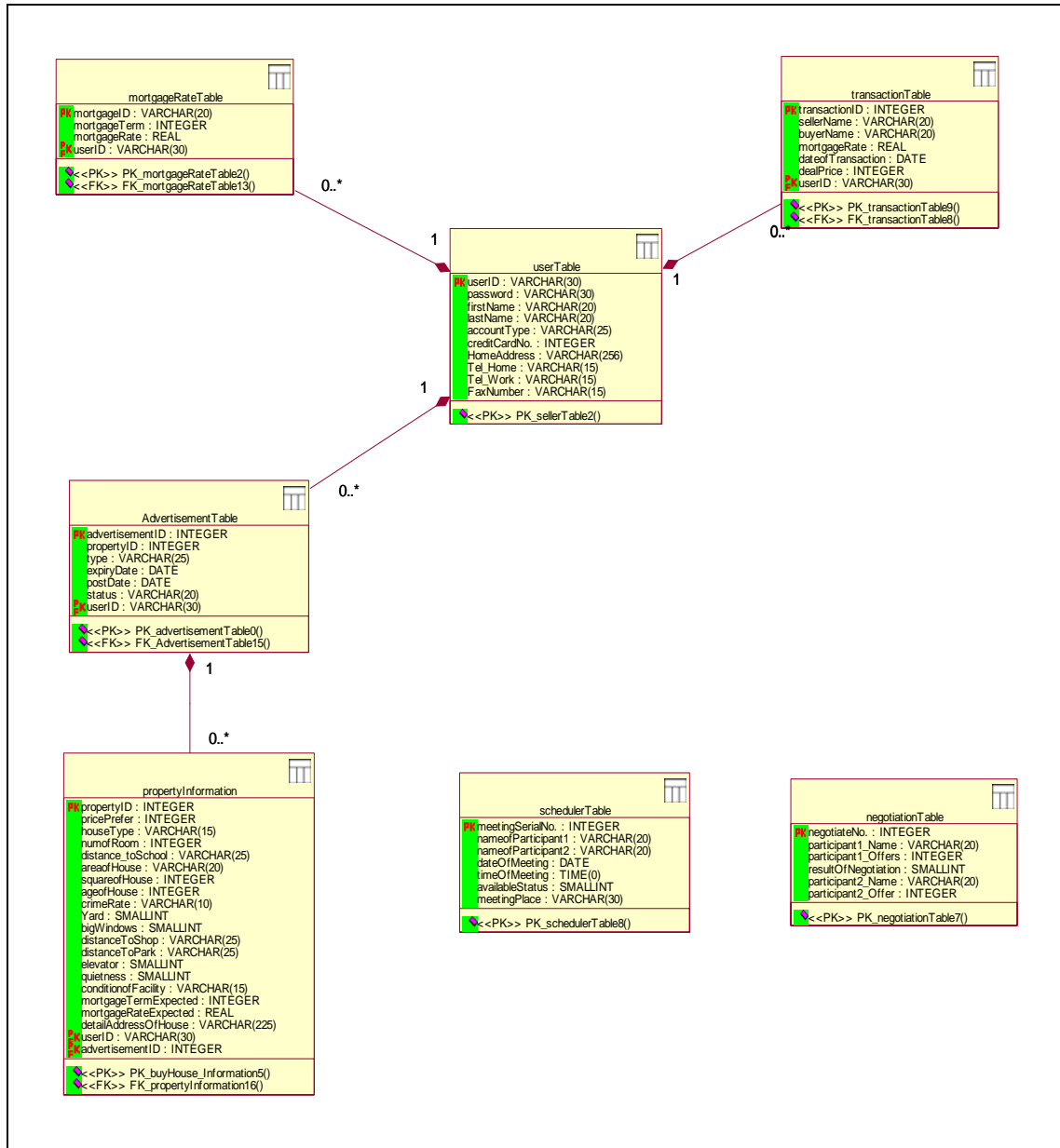


Figure 32 - Database Schema

Data Element	Description	Type
user_ID	Unique ID, actually it is user login name in our case, it is primary key	Varchar(30)
password	User password	Varchar(255)
FirstName	User first name	Varchar(20)
LastName	User last name	Varchar(20)
accountType	accountType has three kinds: seller, buyer and banker	Varchar(25)
creditCardNo	User credit card Number For banker, this element value set to zero	integer
homeAddress	User home address	Varchar(256)
Tel_home	This is the user home telephone number	Varchar(15)
Tel_work	This is the user work telephone number	Varchar(15)
faxNumber	This is the user fax number	Varchar(15)

Table 8 - User Profile Table

Data Element	Description	Type
advertisementID	User advertisement number, it is unique number for each advertisement. This element is the primary key of the table.	integer
propertyID	This is the property ID number	integer
type	Advertisement type includes sell type, buy type, mortgage type	Varchar(20)
expiryDate	expiryDate means when the advertisement will expire	DATE
postDate	postDate means the advertisement post date	DATE
Status	Status refers to the status of the advertisement, it could be valid, expired	
user_ID	Unique ID, actually it is user login name in our case, it is primary key	Varchar(30)

Table 9 - Advertisement Table

Data Element	Description	Type
mortgage_ID	Unique ID, this is the mortgage ID number. Which is posted by banker	Varchar(30)
mortgageTerm	It indicates how long the mortgage will last such as 89 months	integer
mortgageRate	It is the mortgage rate offered by the bank	real
Unique ID	actually it is user login name in our case it is a foreign key at here	Varchar(30)

Table 10 - Mortgage Rate Table

Data Element	Description	Type
propertyID	Unique ID, it is the number of the property assigned by the system	integer
pricePrefer	The favorite price for the property provided by the user who want to sell the property or who want to buy the property.	integer
houseType	This is the type of house. The main type of the house is identified as following type: ("Bungalow", "Townhouse", "Condo", "Duplex" "Fourplex", "Multiplex", "Mobile Home")	Varchar(20)
numofRoom	Number of room	Integer
Distance_toSchool	It represents the distance to school. We define its value among following categories: ("very near", "walking distance", "far")	Varchar(25)
areaofHouse	It represents the general location of the property. Its values are defined as : ("NorthWest", "NorthEast", "SorthEast", "SorthWest", "CityCenterArea")	integer
squareofHouse	It represents that how many square meters of the house	Integer
ageofHouse	It represents the age of the house	Integer
crimeRate	Represents the crime rate of the house location	Varchar(10)
Yard	True means that the property has Yard, false means that it has not.	SMALLINT (Boolean)
bigWindows	True means that the property has big windows, false means that it has not.	SMALLINT (Boolean)
distanceToShop	Represents the distance to shop of the property. We define it has following value: ("very near", "walking distance", "far")	Varchar(25)
DistanceToPark	Represents the distance to park, we define it with following value: ("very near", "walking distance", "far")	Varchar(10)
Elevator	True means that the property has elevator, false means that it has not.	SMALLINT (Boolean)
Quietness	True means that the property is quiet, false means that it is not.	SMALLINT (Boolean)
ConditionofFacility	Represents the condition of facility: we define it has following	Varchar(15)

	values: :("Basic", "Well-facility", "Less", "Good")	
mortgageTermExpected	This is the elements only valid for the user who want to buy the house. It represents the expectation of the user for mortgage term, the unit is “month”	varchar(10)
mortgageRateExpected	This is the elements only valid for the user who want to buy the house. It represents the expectation of the user for mortgage rate	Real
detailAddressOfHouse	Represents the detail address of the house	Varchar(225)
user_ID	Unique ID, actually it is user login name in our case it is a foreign key at here	Varchar(30)
advertisementID	Represents the advertisement ID,	Integer

Table 11 - Property Information Table

Data Element	Description	Type
NegotiateNo	It is the series number of negotiation. It is the primary key for the table.	integer
Participant1_name	It is the one of the participant for the negotiation	Varchar(255)
Participant2_name	It is the other participant for the negotiation	Varchar(20)
Participant1_Offers	This is the negotiation price provide from the first participant	integer
Participant2_Offers	This is the negotiation price provide from the second participant	integer
resultOfNegotiation	This is the result of negotiation, it could be two values one is true which mean reach the deal, another is false which means not reach the deal.	SMALLINT (Boolean)

Table 12 - Negotiation Table

Data Element	Description	Type
meetingSerialNo.	It is the series number for the meeting scheduling. It is the primary key for the table.	integer
nameofParticipant1	It is the name for the first participant for the meeting	Varchar(20)
nameofParticipant2	It is the other name of participant for the meeting	Varchar(20)
dateOfMeeting	This is the date for the meeting	data
timeOfMeeting	This is the time for the meeting	time
availableStatus	This indicates that whether the time for meeting is final made	Varchar(20)

Table 13 - Scheduling Table

Data Element	Description	Type
transactionID	It is the series number of advertisement. It is the primary key for the table.	integer
sellerName	sellerName is the seller's name for the property	Varchar(20)
buyerName	buyerName is the buyer's name for the property	Varchar(20)
mortgageRate	This element is for the mortgage Rate that the user used for buying the house.	Real
dealPrice	Deal price is the last price which the buyer and seller reach the deal.	integer
user_ID	Unique ID, actually it is user login name in our case it is a foreign key at here	Varchar(30)

Table 14 - Transaction Table

6 Electronic Realtor Inter-Agents Messages

As discussed in the previous document, the Inter-Agents message communication is implemented using SOAP. SOAP uses XML as the communication language. For simplicity, the detail content of the SOAP is left out and only the core input and output parts used in the system is presented.

Due to the time constraints, we cannot enumerate all functions' input and output of the system. We only show the input and output of the main function definitions of the system in the following part of the documents.

6.1 Login

This part shows the input and output message of the Login operation. It involves two agents: AccountCenter Agent and DB Agent.

Request:
<pre><login> <userID> string </userID> <password> string </password> </login></pre>
Response:
<pre><login> <validationResult> string </ validationResult > </login></pre>

6.2 CreateNewAccount

This part shows the input and output message of the CreateNewAccount operation. It involves two agents: AccountCenter Agent and DB Agent.

Request:
<pre><CreateNewAccount> <firstName > string </firstName> <lastName> string </lastName> <accountType> string </accountType> <creditCardNo> integer </creditCardNo> <HomeAddress> string </HomeAddress> <Tel_Home> string </Tel_home> <Tel_Work> string </Tel_Work> <FaxNumber> string </FaxNumber> </CreateNewAccount></pre>
Response:
<pre><CreateNewAccount> <userID> string </userID> <password> string </password> </CreateNewAccount></pre>

6.3 checkExpiredAd

This part demonstrates the input and output message of the checkExpiredAd operation. It involves two agents: AccountCenter Agent and DB Agent.

Request:
<pre><checkExpiredAd> <userID> string </userID> <currentDate> date </currentDate> </checkExpiredAd></pre>
Response :
<pre><checkExpiredAd> <checkResult> string </ checkResult > </checkExpiredAd></pre>

6.4 queryAd for house

This part demonstrates the input and output message of the queryAd operation. It involves three agents: Buyer Agent, Seller Agent and DB Agent.

Request:
<pre><queryAd > <queryConstraint> <houseType> string </houseType> <squareofHouse> string </squareofHouse> <areaofHouse> string </areaofHouse> <ageofHouse> integer </ageofHouse> </queryConstraint> </queryAd></pre>
Response :
<pre><queryAd> <advertisementID> integer </advertisementID> <propertyID> integer </propertyID> <houseType> string </houseType> <areaofHouse> string </areaofHouse> <squareofHouse> integer </squareofHouse> <ageofHouse> integer </ageofHouse> <nameofAdPoster> string </nameofAdPoster> <Tel_contact> string </Tel_contact> </queryAd ></pre>

6.5 postAd for house

This part demonstrates the input and output message of the postAd operation. It involves two agents: Sell Agent and DB Agent.

Request:
<pre><postAd> <propertyID> integer </propertyID> <houseType> string </houseType> <areaofHouse> string </areaofHouse> <squareofHouse> integer </squareofHouse> <ageofHouse> integer </ageofHouse> <nameofAdPoster> string </nameofAdPoster> <Tel_contact> string </Tel_contact> </postAd></pre>
Response :
<pre><postAd status="bool"> <advertisementID> integer </advertisementID> </postAd></pre>

6.6 Offer for house

This part demonstrates the input and output message of the offer a price for buying house operation. It involves three agents: Buyer Agent, Negotiation Agent and Seller Agent.

Request:
<pre><offer> <buyerID> string </buyerID> <Tel_contact> string </Tel_contact> <priceprefer> integer </priceprefer> <propertyID> integer </propertyID> <sellerID> string </sellerID> </offer></pre>
Response :
<pre><offer> <feedback> string </feedback> <otherChoice> string </otherChoice> </offer></pre>

6.7 Schedule for meeting

This part demonstrates the input and output message of the scheduling meeting operation. It involves four agents: Buyer Agent, Negotiation Agent, Seller Agent and Banker Agent.

Request:
<pre><schedule> <nameofParticipant1> string </nmeofParticipant1> <nameofparticipant2> string </nameofParticipant2> <dateOfMeeting> DATE </dateOfMeeting> <timeOfMeeting> TIME(0) </timeofMeeting> <meetingPlace> string </meetingPlace> </schedule></pre>
Response :
<pre><schedule></pre>

```

    <feedback> string </feedback>
    <otherChoice> string </otherChoice>
</schedule>

```

6.8 postBankRate

This part demonstrates the input and output message of the post Bank Rate operation. It involves two agents: Banker Agent and DB Agent.

Request:
<pre> <postBankRate> <bankName> string </nmeofParticipant1> <mortgageTerm> integer </nameofParticipant2> <mortgageRate> integer </dateOfMeeting> </postBankRate> </pre>
Response:
<pre> <postBankRate> <mortgageID> string </mortgageID> </postBankRate> </pre>

6.9 OpenTransaction

This part demonstrates the input and output message of the opening Transaction operation. It involves two agents: Negotiation Agent and Transaction Agent.

Request:
<pre> <openTransaction propertyID="ID"> <sellerName> string </sellerName> <buyername> string </buyerName> <quotation> <dateofTransaction> DATE </dateOfTransaction> <dealPrice> integer </dealPrice> . . . </quotation> </openTransaction> </pre>
Response:
<pre> <openTransaction status="string"> <transactionID> string </transactionID> </openTransaction> </pre>

6.10 closeTransaction

This part demonstrates the input and output message of the closing Transaction operation. It involves two agents: Transaction Agent and DB Agent.

Request:
<pre><closeTransaction transactionID="string"> <sellerName> string </sellerName> <buyername> string </buyerName> <quotation> <dateofTransaction> DATE </dateOfTransaction> <dealPrice> integer </dealPrice> <mortgageRate> float </morgageRate> . . </quotation> </closeTransaction></pre>
Response :
<pre><closeTransaction> <status> string </status> </closeTransaction></pre>

7 Conclusions

7.1 Experience of the project

From the design of this project, the team has gained a significant experience in applying the agent based software design concepts learned in class to a project design. The experience that the team obtained from this project is summarized as follows:

- ❑ In the multi-agent system, system analysis is a very important step in an agent-based project development.
- ❑ System analysis based on “role” modeling is much effective than other methodologies.
- ❑ Identifying enough and right communication and interaction among different agents are very important part for the agent based system design.
- ❑ Sequence diagram can help us better understanding the relationship among different agents.
- ❑ Clearly classification of various types of the interactions among all the agents involved is very important for the successful analysis and design.
- ❑ Better understanding the ontology of the domain is very helpful for the construction of the knowledge base.

7.2 Discussion

Because of the limitation of the time frame given, the design presented in this document can not been implemented. This more or less made us has less confidence to say it is a successful design. Nevertheless, the following limitations have been identified:

- ❑ The requirements might only partially reflect the real situation of the business. During the requirements elicitation process, we mimic the role of customers and try our best to get as many requirements as possible so that we can have a reasonable design.
- ❑ Typically, design should be reviewed and revised based on the user feedback and new requirements added before starting the implementation phase. Due to the time constraints, the team has not thoroughly gone over all the design details against the list of requirements specified.
- ❑ This is the first project we did based on the Agent methodology. We have to acknowledge that it did take us a lot of time and it had made us go back and forth several times due to lacking enough experience for the Agent system design. A lot of discussion and group meetings have been made during the past four weeks. We believe that it might still exist some problems in the document even though we try our best to reduce it to the minimum amount.

Overall, the whole analysis and design process for the Multi-Agent System for Electronic Realtor is really a success and learning process for the team. The concepts, knowledge gained from course and experience learned from the project have made us have a better understanding of the agent system and its analysis and design methodology.

References

- [1] M. Wooldridge, N. R. Jennings, and D. Kinny: A Methodology for Agent-Oriented Analysis and Design. In O. Etzioni, J. P. Muller, and J. Bradshaw, editors: Agents '99: Proceedings of the Third International Conference on Autonomous Agents Seattle, WA, May 1998.
- [2] J. Lind.: The MASSIVE Development Method for Multiagent Systems. In Proceedings of the Fifth International Conference on the Practical Application of Intelligent Agents and Multi-Agents (PAAM2000), Manchester, 2000.
- [3] D.L. Martin, A.J. Cheyer, D.B. Moran, The Open Agent Architecture: A Framework for building Distributed Software Systems, Applied Artificial Intelligence: An International Journal. Volume 13, Number 1-2, January-March 1999. pp 91-128.
- [4] Tim Finin, Yannis Labrou, and James Mayfield , KQML as an agent communication language in Jeff Bradshaw (Ed.), Software Agents, MIT Press (1997).
- [5] Dr. B. H. Far, Course slides for SENG 609.22 Agent-based Software Engineering

