Research Matchmaking Project

Agent-Based Software Engineering
(SENG 696)

(Group 13)
Marko Mijovic (30019553)
Subroto Kumer Deb Nath (30187307)

Detailed Development Document
(Assignment 2)

UNIVERSITY OF
CALGARY

SCHULICH
School of Engineering

Table of Contents

1. Introduction:

Research matchmaking is a web-based app to find individuals or organizations of researchers, contributors, experts, etc. for a specific project or research work with similar interests and experience to complete a certain project or research work as a paid (in most cases) or free (very often) service. This tool is very useful for finding paid or free contributors, aka researchers, for an incomplete research work or project. The similar and matched researchers/contributors help by serving information, data, codes, guidelines, methodologies, research scopes, analysis reports, direct efforts, and sometimes the whole or a part of the research project to complete the client's project or research work. In this project, we'll develop a complete research matchmaking system that will help clients to find service providers and vice-versa.

2. System Specification:

A Graphical User Interface (GUI) will assist both clients and service providers in creating user accounts/signing up and logging into the system in this research matchmaking project. The providers can get verified with their credentials, and the clients will see a verified icon just beside the name of a service provider who has already been verified in the system. Obviously, clients will trust a verified service provider to share their confidential research-related data-set and information. The service providers can share their name, fields of expertise, resume, field-related certifications, resume, portfolios, website links, special keywords, service fee, etc. This information will help the clients select their preferred service providers easily. There will be another feature for the client to login to the system as a guest. During the guest session, a guest client will be able to search for the service providers with a specific keyword or find one or more service providers from the database. But the guest client can't place a bid for any service and can't see the service fee of any service provider. Fully signed-up clients and providers can see the entire system and use the app. The providers can choose either a regular basic plan or a premium plan during the contract with the system. Premium subscribers will appear at the top of the search results, followed by the basic verified service providers when a client searches for service providers for a specific service. Thus, premium service providers will have more

access to the clients, which will help them increase their earnings. The clients can place bids requesting service providers.

The service provider can accept or reject the bid. In the event of a rejection, the client will be notified directly. The system will generate a contract letter and send the letter to both the client and the provider. The provider and the client will get a chance to review the contact letter and accept or reject the contract. Upon acceptance, all types of payment and financial transactions will be handled by the system to make the overall app reliable and secure. The system will receive a 30% fee for any transaction or payment. Both the client and the provider can monitor the progress of a project or research during the timeline. The App can estimate the tentative deadline by analyzing the current progress report and pace. The client and the provider can do live chat during the whole project timeline. The clients can give ratings and feedback to the providers on their quality, time of completion, and expertise, etc. These ratings will be displayed in the provider's profile. These are the features and specifications of the app.

3. Development Methodology:

We've selected the MaSE Methodology for our system design. MaSE is one of the most advanced methodologies for developing multi-agent-based software or systems. MaSE combines several established models into a comprehensive methodology, and it guides the developer in the analysis and design process.

4. Development Environment:

The system was developed in the following environment:

· Java 8 (JDK 11.0.11)

· JADE 4.5.0 revision 6825

· MySQL Database 8.0.22 (For offline testing)

· IDE: IntelliJ IDEA Community V2020.3

5. System Architecture:

In this Research Matchmaking Project, we have six available agents and four databases. The overall high-level graphical system architecture is demonstrated in figure 5.1.
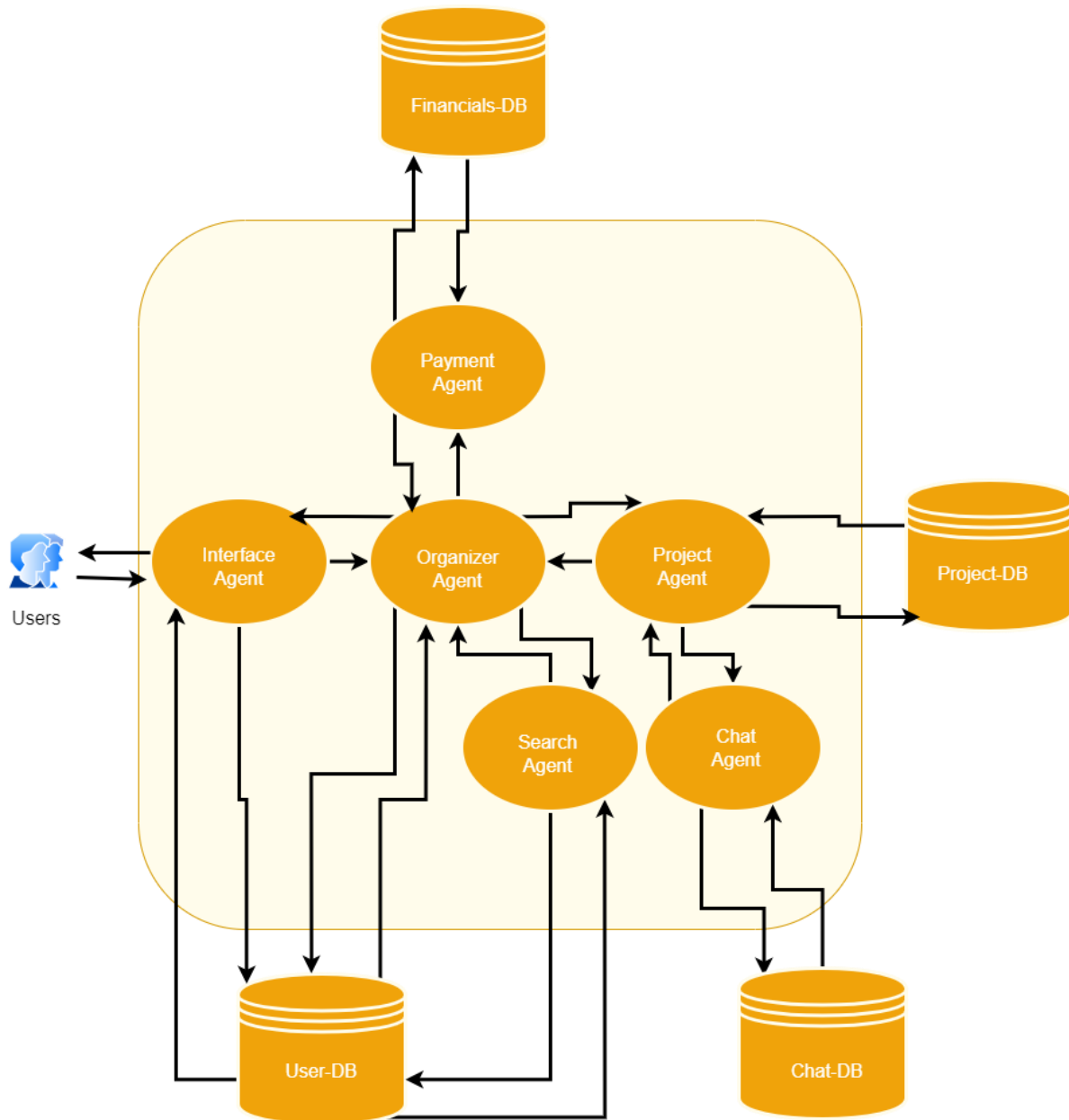

Figure 5.1: High-Level System Architecture

6. Use Case Definition of Agents:

In this Research Matchmaking project, we've six individual agents with one or more roles. We've demonstrated the use case diagram and the use case definition for all the available agents sequentially in this section. The available agents are-

1. Interface Agent
2. Organizer Agent
3. Search Agent
4. Project Agent
5. Payment Agent &
6. Chat Agent

6.1.1 Use Case Diagram of Interface Agent:

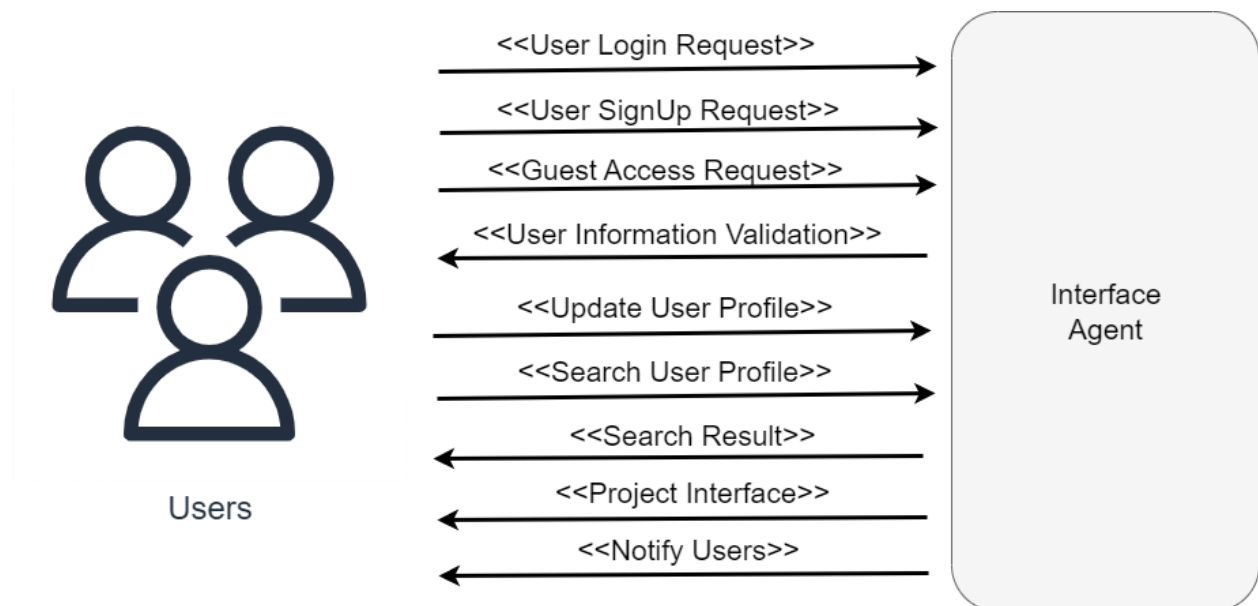The use case diagram of Interface Agent is demonstrated in figure 6.1.1.

Figure 6.1.1: Use Case Diagram of Interface Agent

6.1.2 Use Case Definition of Interface Agent:

Brief Description:	The users use this use case to sign-up, login, login as a guest to the system as well as manage their profile, search users, bid for a project, chat and use the overall system with an interface.
Pre-Condition:	Need to signup and login as a verified provider or a verified client to get full access of this system. A guest will get access to search provider profiles only.
Post Condition:	Not Applicable
Process Steps	
1	The users want to login, signup or get access as a guest to the system.
2	For login to the system a pre-verified user has to provide the login related credentials in login interface. For a new user signup, the user has to request for signup interface. For a guest access, the guest has to select proceed as a guest option.
3	In this step login process will be started for login option, the system will redirect the user to signup interface upon requesting signup option in previous step. The system will start managing a new guest if the user selected proceed as a guest in the previous step. In this step, Organizer Agent helps the Interface Agent to perform these operations.
4	In this step the system will authenticate the login information with the help of User-Database. The signup information and credentials are going to be saved at User-Database after validation process. The guest will be redirect to search interface.
5	Upon valid login credentials, the user will be logged in to the system successfully and redirect to his user profile interface. For valid signup information, the signup credentials will be saved and the user will be redirected to login interface.
6	A logged in user can manage his/her profile, update the information, browse the projects, bid for a project(provider), see project progress, search for other profiles, etc.

7	Verified and logged in users (both providers and client) will get notifications related to projects and can chat together using this interface agent.

Exceptions:		
1a	System isn't accessible at this point.	An error message will be generated stating that the system isn't accessible at this point. Use case terminated.
4a	User-Database isn't accessible at this point.	An error message will be generated stating that the user-database isn't accessible at this point. Use case terminated.

Relationships:	
Initiator	Users
Collaborators	Organizer Agent, Project Agent, Search Agent, Chat Agent, User-Database, Project-Database

Data Requirements:	
Data Required:	• Users Information • Login Credentials • Project Information • User Profile Information • Notification Information • Priority Based Search Results

6.2.1 Use Case Diagram of Organizer Agent:

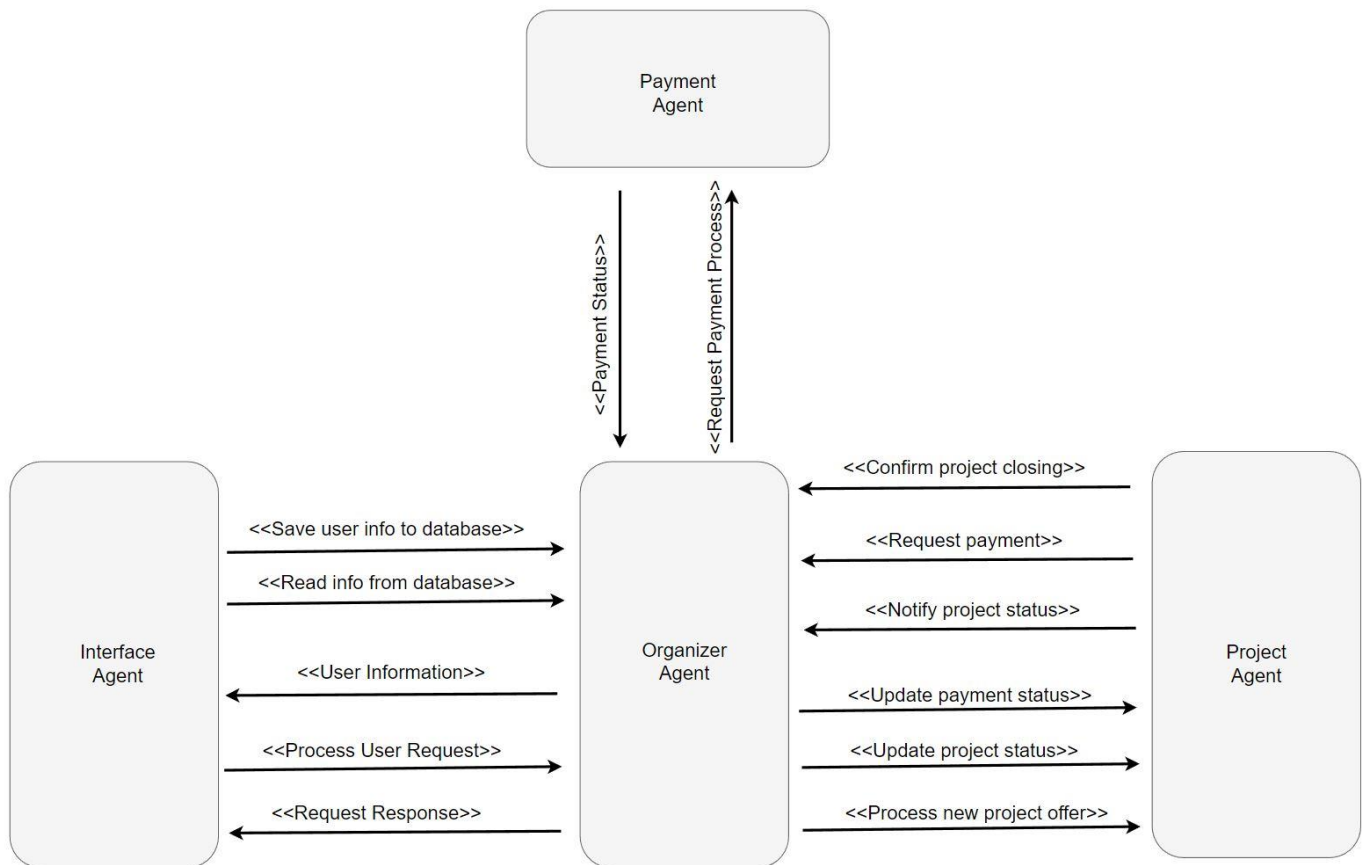The use case diagram of Organizer Agent is demonstrated in figure 6.2.1.

Figure 6.2.1: Use Case Diagram of Organizer Agent

6.2.2 Use Case Definition of Organizer Agent:

Brief Description:	The Organizer Agents handles a lot of things inside the system. It sets basic or premium plan for the provider, manages query responses, controls guest access, manages user profile, initializes new project, updates and validates users' credentials, notifies the both clients and providers and initializes the payment after successful project completion.
Pre-Condition:	The users need to be logged in either as a provider or a client.
Post Condition:	Not Applicable

Process Steps	
1	After successful login, both the client and provider get access to their profile. Guest login is also managed at this step. Guest is redirected to search agent.
2	Providers can update their plan at this step.
3	A client can visit providers profile and see their portfolio, ratings and hourly rate.
4	A client can directly hire a provider or can place a bid.
5	Service provider can bid on the project posting by the client.
6	After a successful consent of both provider and client in a particular job, Organizer Agent requests the Project Agent to draft a contract.
7	A contract is generated to start the project by the provider and client.
8	The contract can be modified by this agent upon the client's or provider's request with the help of Project Agent.
9	The client can track the progress of a project over the time.
10	After a successful project completion, the organizer agent helps the Payment Agent for processing the payment.

Exceptions:		
7a	The project isn't initiated due to no contract signing	An error message will be generated stating that the project will not be initiated due to no contract signing, please sign the current contract or modify the

		contract. Use case terminated.
Relationships:		
Initiator	Interface Agent	
Collaborators	Project Agent, Search Agent, Payment Agent, User-Database, Project-Database, Financials-Database	
Data Requirements:		
Data Required:	- Client and Provider Information - Login Credentials - Project Information - User Profile Information - Notification Information - Priority Based Search Results - Payment Information (Hourly Rate, Payment ID) - Project Deadline	

6.3.1 Use Case Diagram of Search Agent:

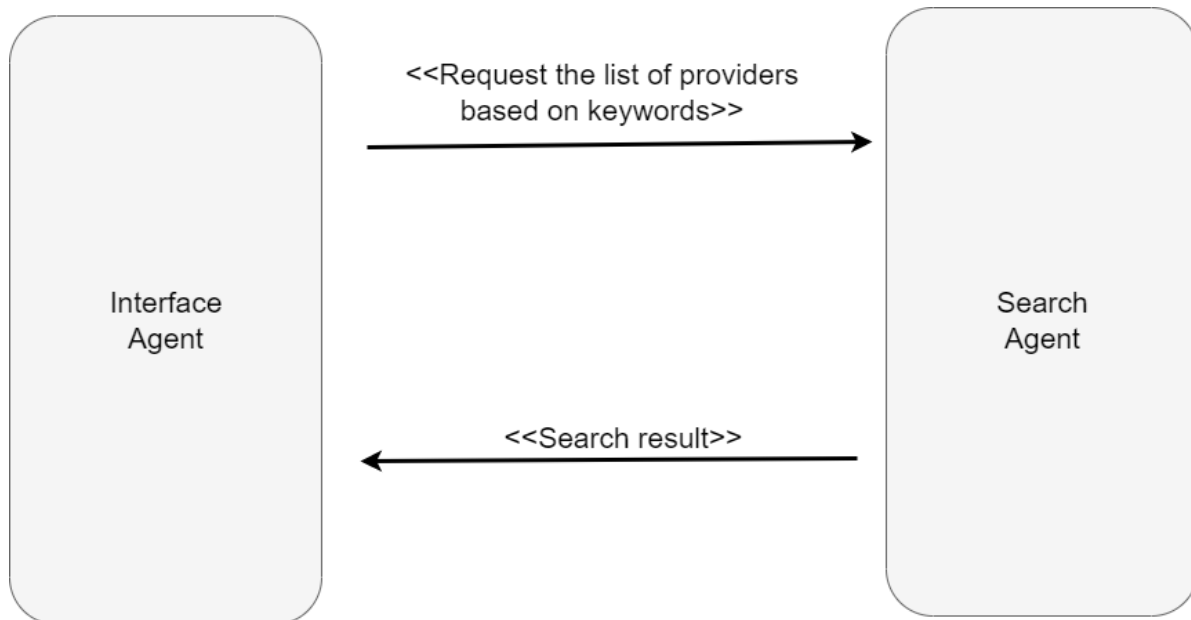The use case diagram of Search Agent is demonstrated in figure 6.3.1.

Figure 6.3.1: Use Case Diagram of Search Agent

6.3.2 Use Case Definition of Search Agent:

Brief Description:	The users use this use case to search for both basic and premium provider profile.
Pre-Condition:	The users need to be logged in either as a client or a guest.
Post Condition:	Not Applicable
Process Steps	
1	The users use the client ID or a guest option to login to the system
2	Users request providers list by entering some keywords.
3	The Interface Agent reads the information and transmits the keywords to the Search Agent.
4	The Search Agent will retrieve information related to providers from the user-database and organizes the provider list according to the premium and basic membership and keywords matching.
5	The Interface Agent will get the provider information (premium provider on the top) from the Search Agent and demonstrate the sorted list to the clients or guests .

Exceptions:		
4a	User-Database isn't accessible at this point.	An error message will be generated stating that the user-database isn't accessible at this point. Use case terminated.

Relationships:	
Initiator	Interface Agent
Collaborators	Users, Organizer Agent, User-Database

Data Requirements:	
Data Required:	• Keywords • List of Providers • Membership Information

6.4.1 Use Case Diagram of Project Agent:

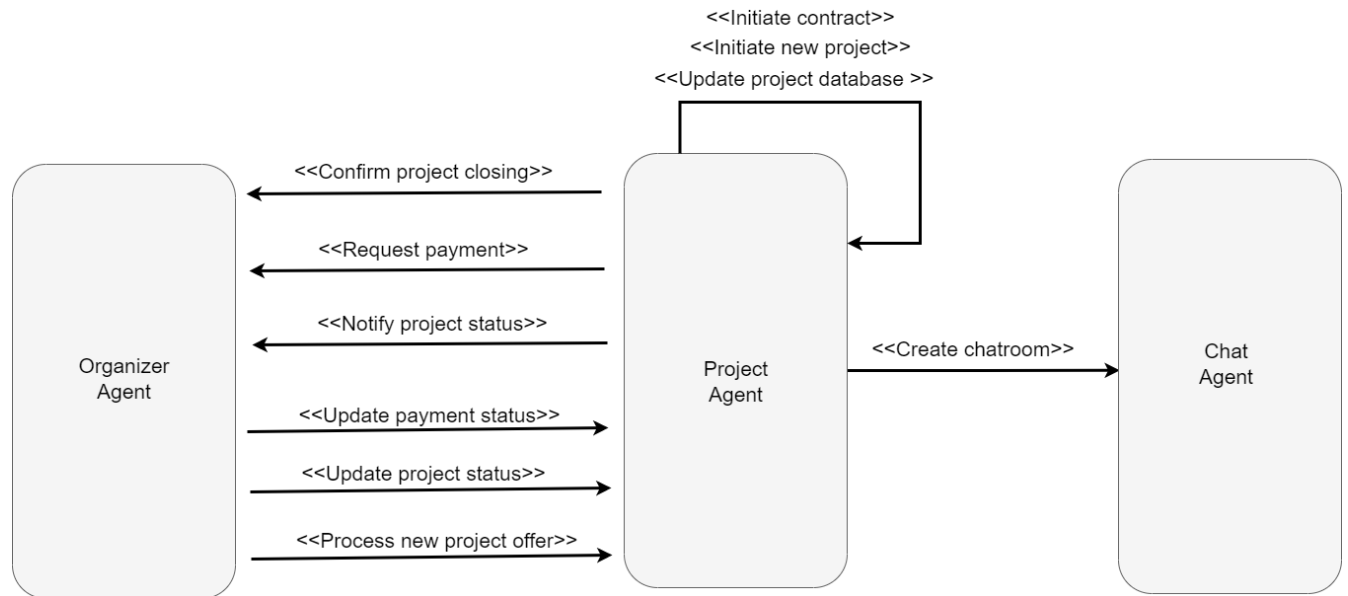The use case diagram of Project Agent is demonstrated in figure 6.4.1.

Figure 6.4.1: Use Case Diagram of Project Agent

6.4.2 Use Case Definition of Project Agent:

Brief Description:	The Organizer Agent uses this use case to initiate, record, process a project and to create contract, deadline for a project. As well as Project Agent helps to create chatroom for the Chat Agent and tracks the overall progress.
Pre-Condition:	Both the Provider and The Client have to be agreed for a project.
Post Condition:	Not Applicable
Process Steps	
1	Upon agreed by both parties, the organizer Agent requests to generate a contract to Project Agent.
2	Project Agent create a draft contract and sends to Organizer Agent.
3	If there's any change request for the contract, the Organizer Agent changes the contract and gets sign of both parties by Interface Agent.
4	Upon signing the contract by both parties, the Project Agent initiates a project and saved the contract and project information to the Project-Database.
5	The Project Agent tracks the progress of the project and informs to Organizer Agent.
6	In this step the Project Agent gets help from the Chat Agent and creates a project-specific chatroom for both provider and the client.
7	The Project Agent helps the Client to give rating for the provider and the vice-versa at successful completion of the project.
8	The Project Agent close the project and updates the Project-Database. In this step, the Project Agent gives the final information about the project completion and the rating information of both client and provider.
9	The Organize Agent updates the User-Database with the rating details of both client and provider and request the Payment Agent to process the payment. And the project finally gets closed.

Exceptions:		
4a	Project-Database isn't accessible at this point.	An error message will be generated stating that the Project-Database isn't accessible at this point. Use case terminated.
8a	Project-Database isn't accessible at this point.	An error message will be generated stating that the Project-Database isn't accessible at this point. Use case terminated.
9a	User-Database isn't accessible at this point.	An error message will be generated stating that the User-Database isn't accessible at this point. Use case terminated.

Relationships:	
Initiator	Organizer Agent
Collaborators	Interface Agent, Chat Agent, Users, User-Database, Project-Database

Data Requirements:	
Data Required:	- Client Information - Provider Information - Hourly Rate - Project Deadline - Contract Information - Payment Information - Rating Information for both parties - Membership Information for the Provider - Project Information

6.5.1 Use Case Diagram of Payment Agent:

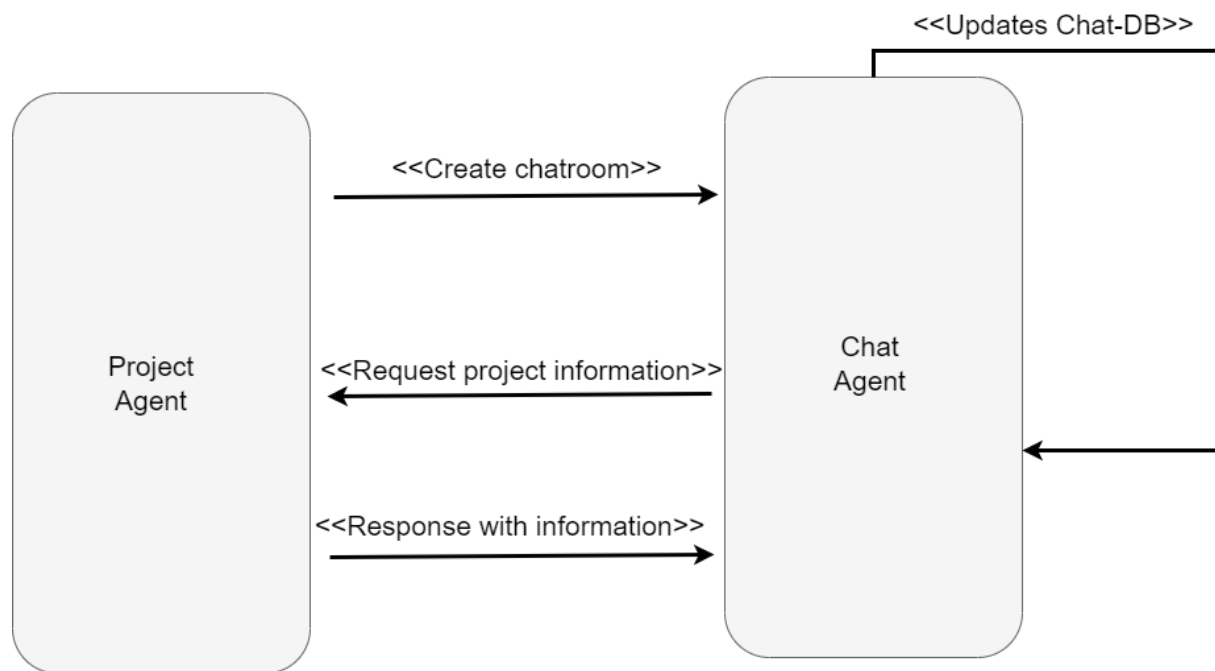The use case diagram of Payment Agent is demonstrated in figure 6.5.1.

Figure 6.5.1: Use Case Diagram of Payment Agent

6.5.2 Use Case Definition of Payment Agent:

Brief Description:	The Organizer Agent uses this use case to calculate, initiate, and process the service fee for the provider upon a successful project completion. This use case also helps the Organizer Agent to deduct the Matchmaking Project Service Charge from the bill paid by the client
Pre-Condition:	A project has to be completed successfully to initiate this.
Post Condition:	Not Applicable
Process Steps	
1	After a successful project the Organizer Agent requests the Payment Agent to process the payment.
2	The Payment Agent gets the financial transaction related data from the Financials-Database for both Client and Provider.

3		The Organizer Agent sends the hourly rate and total work hours related information to the Payment Agent after getting this information from the Project-Database via the Project Agent.
4		The Payment Agent Calculate the total service fee and ask the client to transfer the amount.
5		When the fund from the clients reaches to the Payment Agent, it deducts the Service Charge for the system as per the membership type of the provider.
6		After deduction, the Payment Agent process the payment for the provider and updates everything to its Financials-Database.
7		After sending the fund to the provider, the Payment Agent updates the Organizer Agent.

Exceptions:		
2a	Financial-Database isn't accessible at this point.	An error message will be generated stating that the Financials-Database isn't accessible at this point. Use case terminated.
3a	Project-Database isn't accessible at this point.	An error message will be generated stating that the Project-Database isn't accessible at this point. Use case terminated.
6a	Financials-Database isn't accessible at this point.	An error message will be generated stating that the Financials-Database isn't accessible at this point. Use case terminated.

Relationships:	
Initiator	Organizer Agent
Collaborators	Users, Financials-Database, Project-Database

Data Requirements:	
Data Required:	• Client Information • Provider Information

	• Hourly Rate
	• Payment Information
	• Membership Information for the Provider
	• Project Information

6.6.1 Use Case Diagram of Chat Agent:

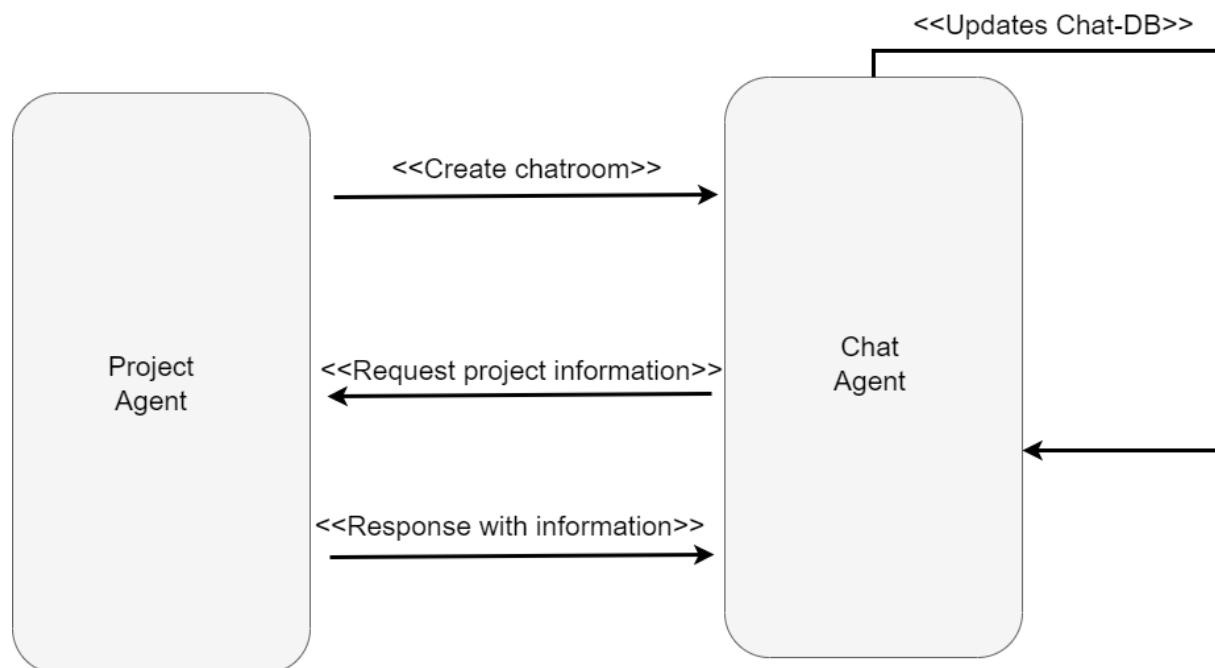The use case diagram of Chat Agent is demonstrated in figure 6.6.1.

Figure 6.6.1: Use Case Diagram of Chat Agent

6.6.2 Use Case Definition of Chat Agent:

Brief Description:	The Project Agent uses this use case to create chatroom for the both Client and Provider after successfully initiating of a project.	
Pre-Condition:	A project has to be initiated successfully to initiate this.	
Post Condition:	Not Applicable	
Process Steps		
1	After a successful initialization of a project the Project Agent requests the Chat Agent to create a chatroom for the both client and provider.	
2	The Chat Agent creates a private chatroom for the both client and provider.	
3	The Chat Agent gets the project related information from the Project-Database via Project Agent.	
4	Bothe the Client and Provider can chat together about their project updates and progress with the help of Interface Agent.	
5	During the project closing the chatroom gets closed and the chat history is saved to Chat-Database.	
Exceptions:		
3a	Project-Database isn't accessible at this point.	An error message will be generated stating that the Project-Database isn't accessible at this point. Use case terminated.
5a	Chat-Database isn't accessible at this point.	An error message will be generated stating that the Chat-Database isn't accessible at this point. Use case terminated.
Relationships:		
Initiator	Project Agent	
Collaborators	Users, Interface Agent, Project-Database, Chat-Database	
Data Requirements:		
Data Required:	- Client Information - Provider Information - Project Information	

7. Detailed Class Diagram:

A high-level view of the class diagram is demonstrated in the figure 7.1. The detailed class diagrams using all the available agents are demonstrated sequentially in the figure 7.2(a), 7.2(b), 7.2(c), 7.2(d), 7.2(e) and 7.2(f) for better view and understanding.
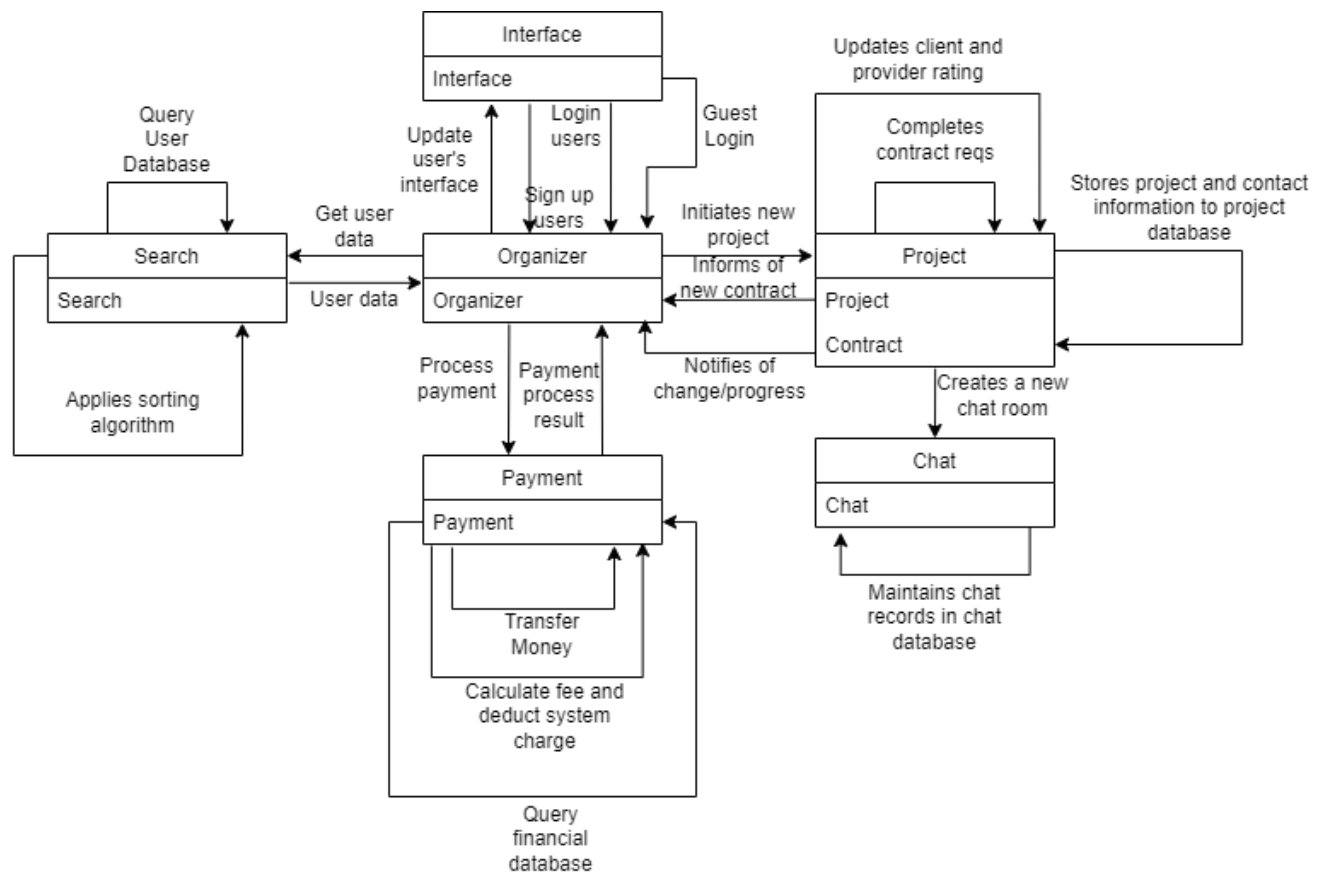

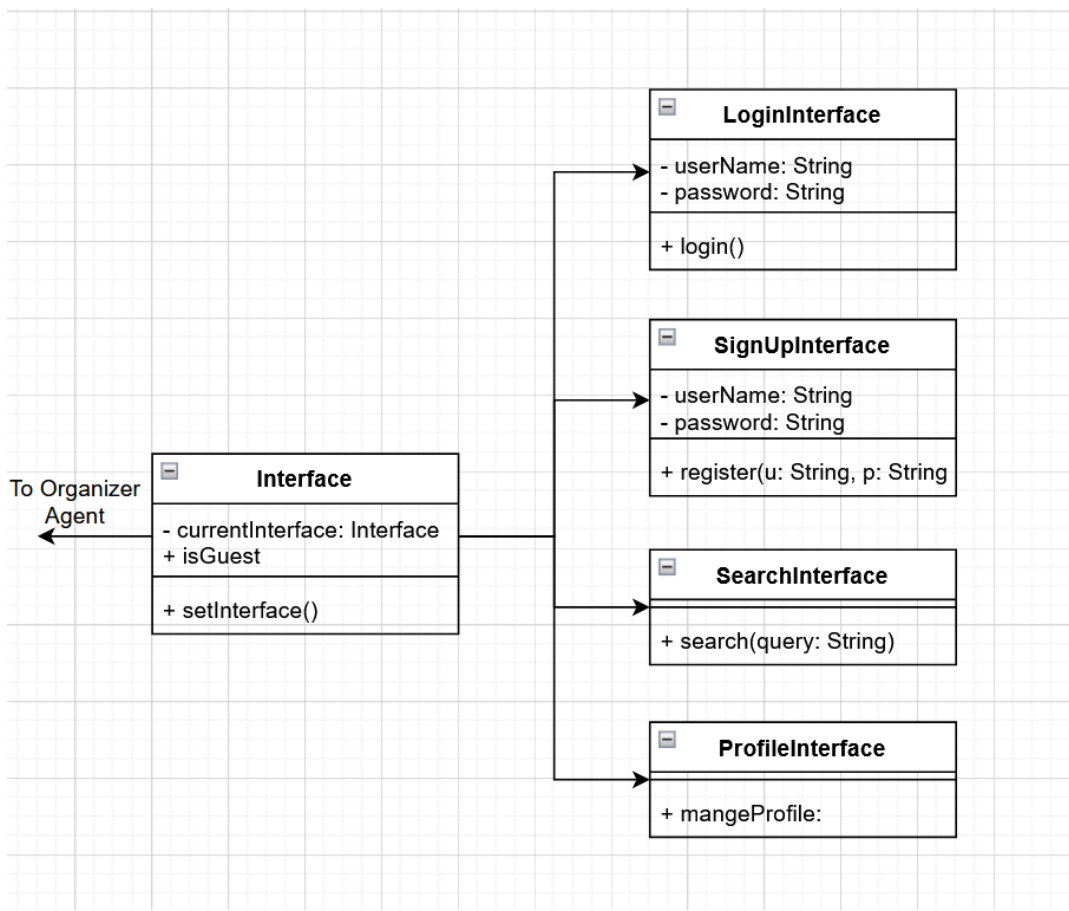Figure 7.1: High-Level View of Class Diagram of Research Matchmaking Project

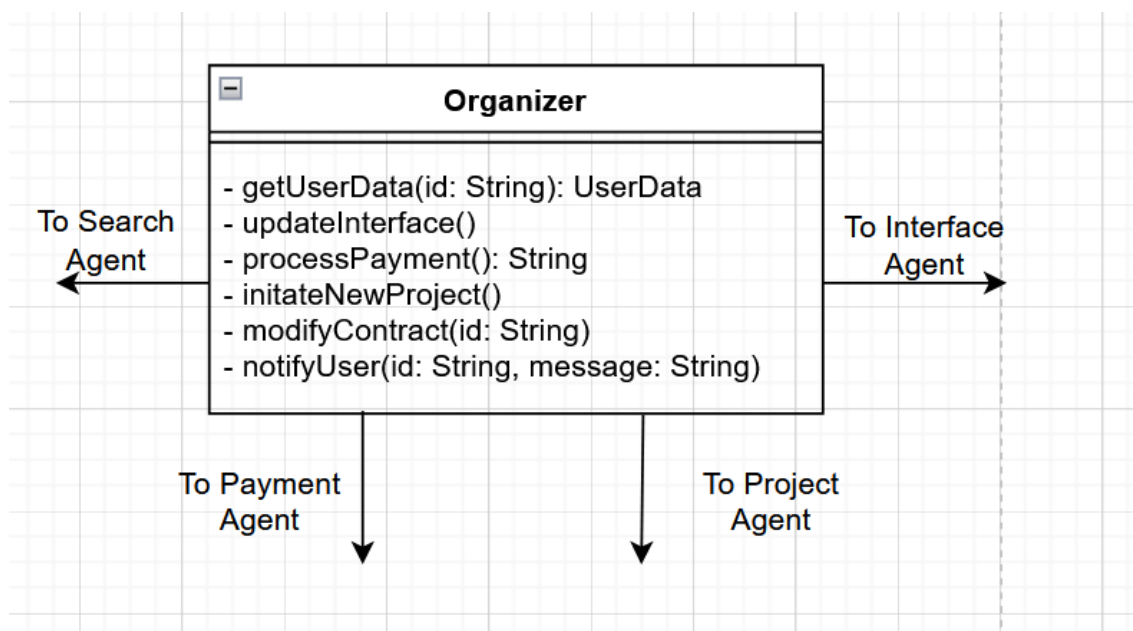Figure 7.2(a): Detailed Class Diagram of Interface Agent
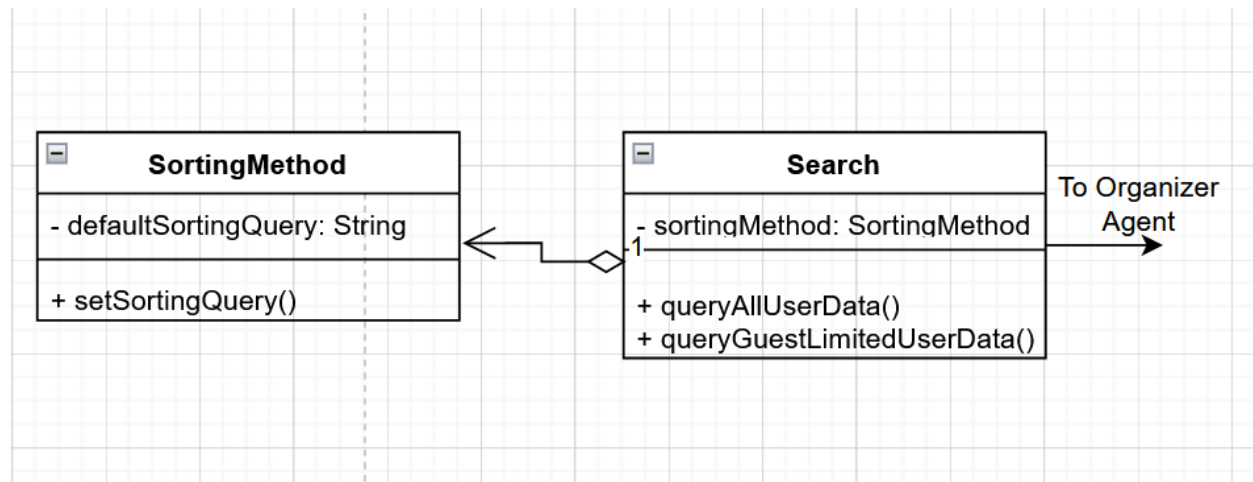

Figure 7.2(b): Detailed Class Diagram of Organizer Agent

Figure 7.2(c): Detailed Class Diagram of Search Agent
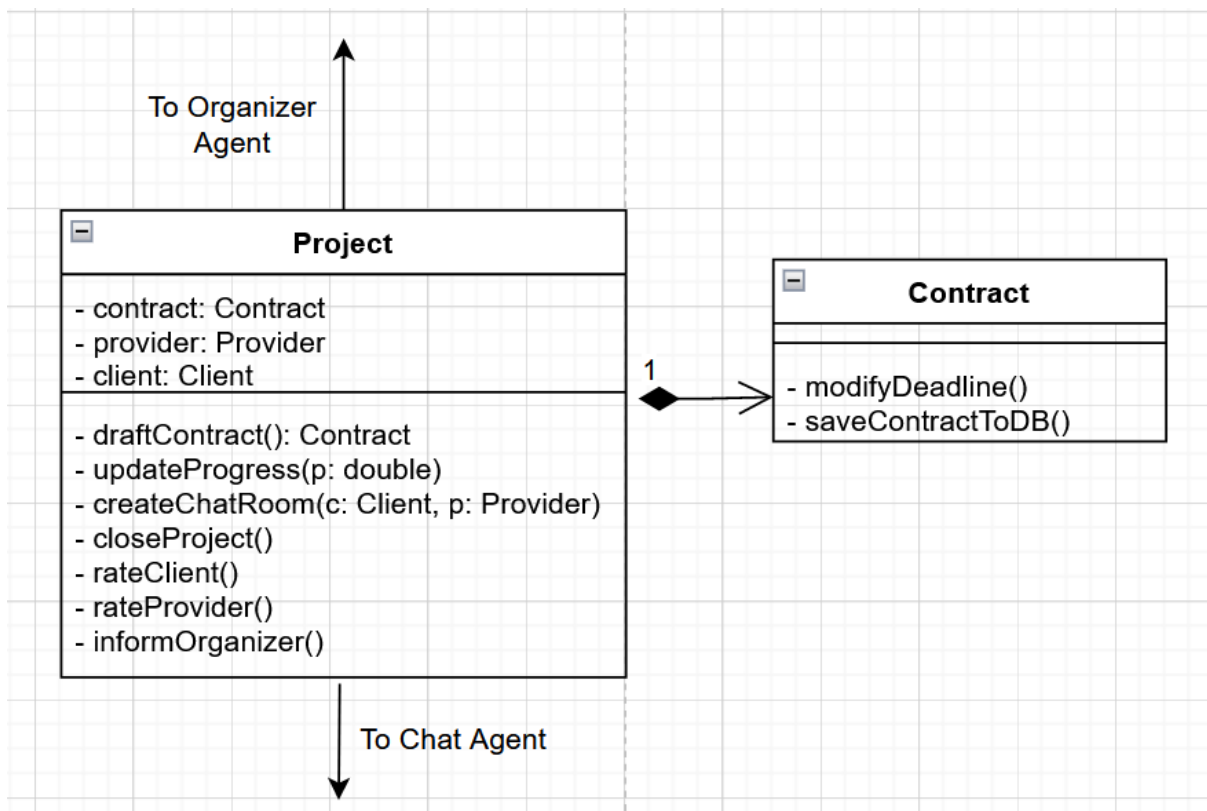
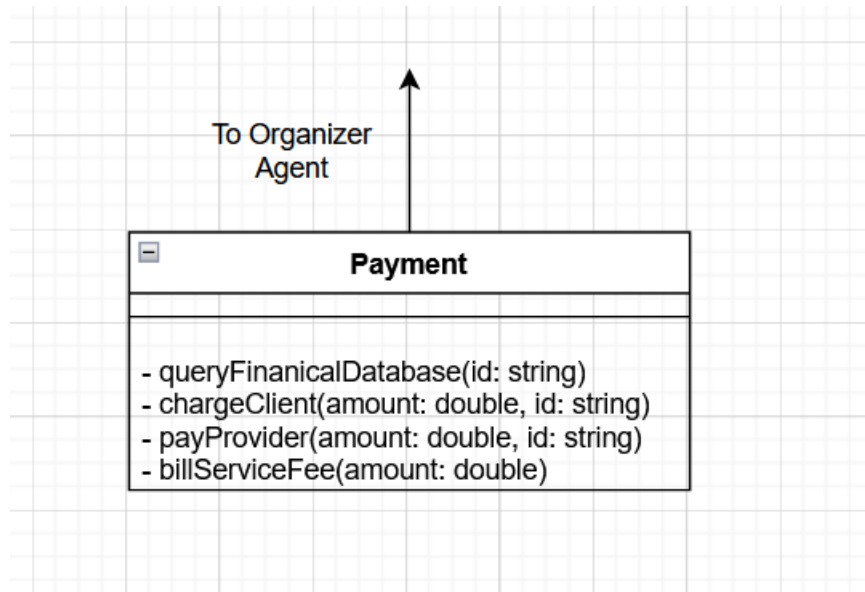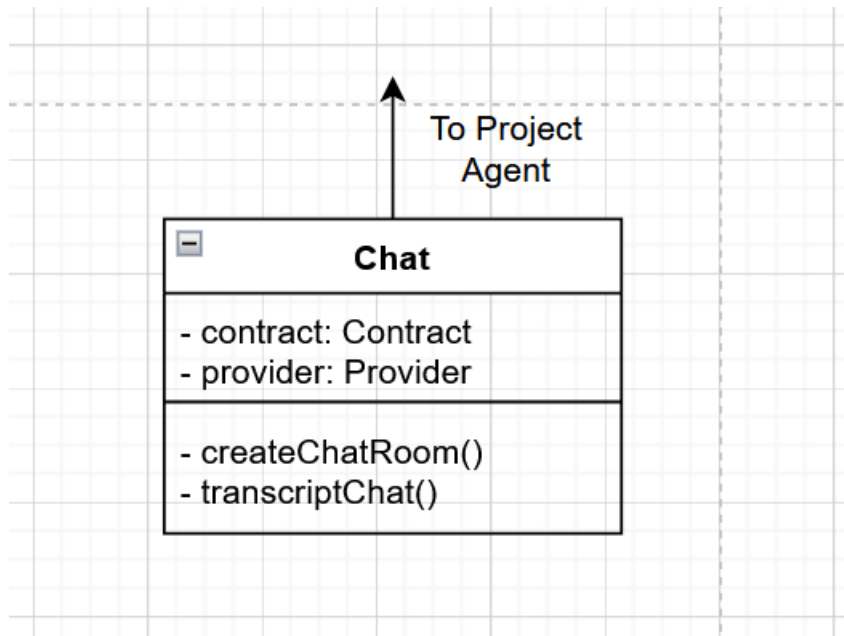Figure 7.2(d): Detailed Class Diagram of Project Agent

To Organizer
Agent

Payment

- queryFinanicalDatabase(id: string)
- chargeClient(amount: double, id: string)
- payProvider(amount: double, id: string)
- billServiceFee(amount: double)

Figure 7.2(e): Detailed Class Diagram of Payment Agent

To Project
Agent

Chat

- contract: Contract
- provider: Provider

- createChatRoom()
- transcriptChat()

Figure 7.2(f): Detailed Class Diagram of Chat Agent

8. Message Sequence Chart:

In this section, we demonstrated message sequence chart. For better view and understanding, we divided the message sequence chart in five individual parts. We demonstrated the total message sequence chart sequentially in figure 8(a), 8(b), 8(c), 8(d), and 8(e).
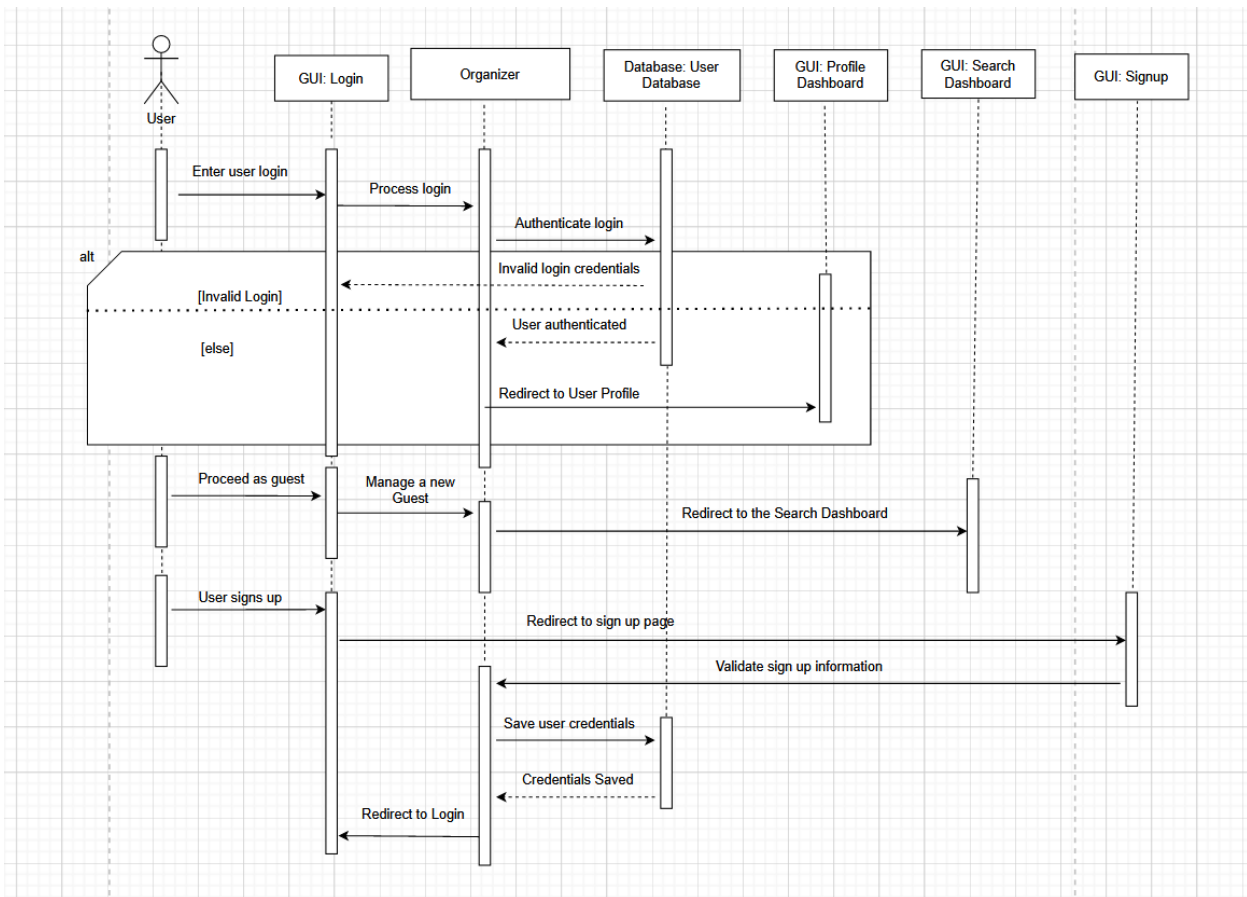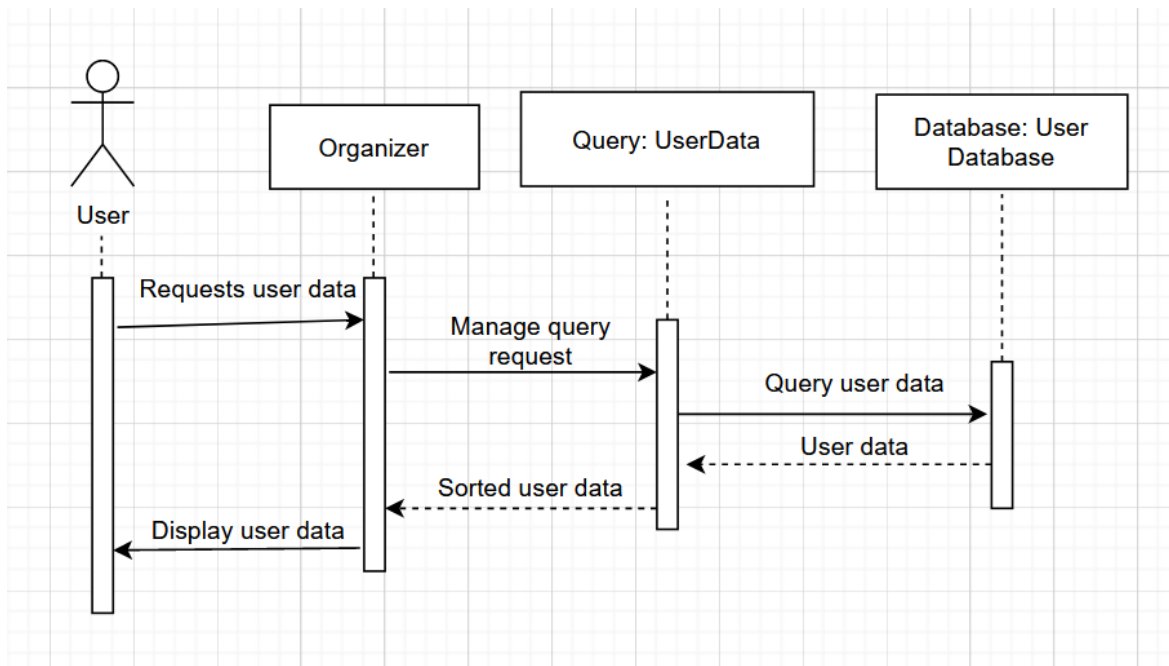

Figure 8(a): Message Sequence Chart Part 1

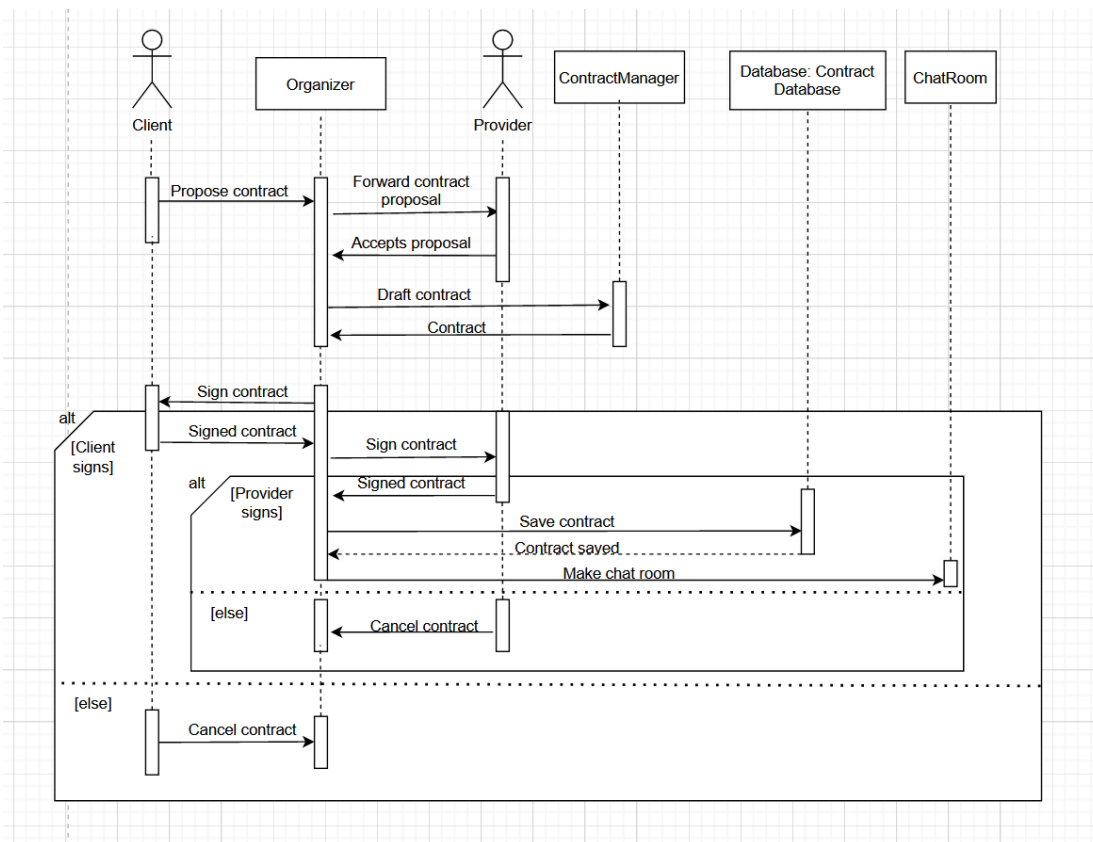Figure 8(b): Message Sequence Chart Part 2

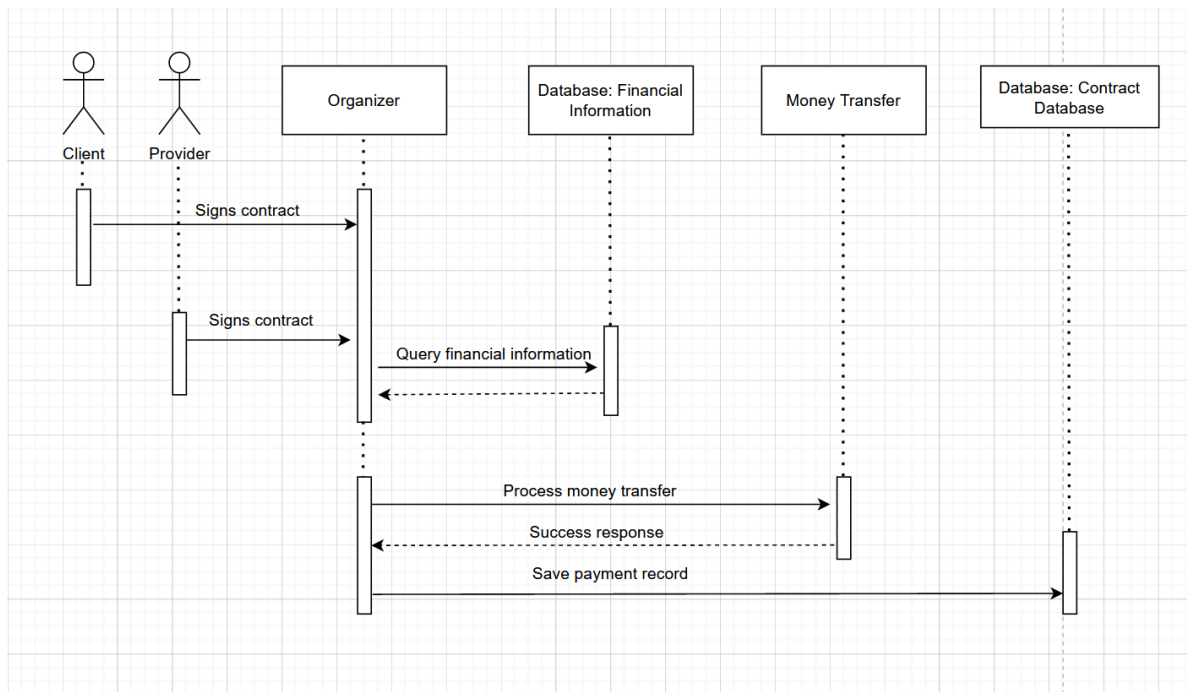Figure 8(c): Message Sequence Chart Part 3
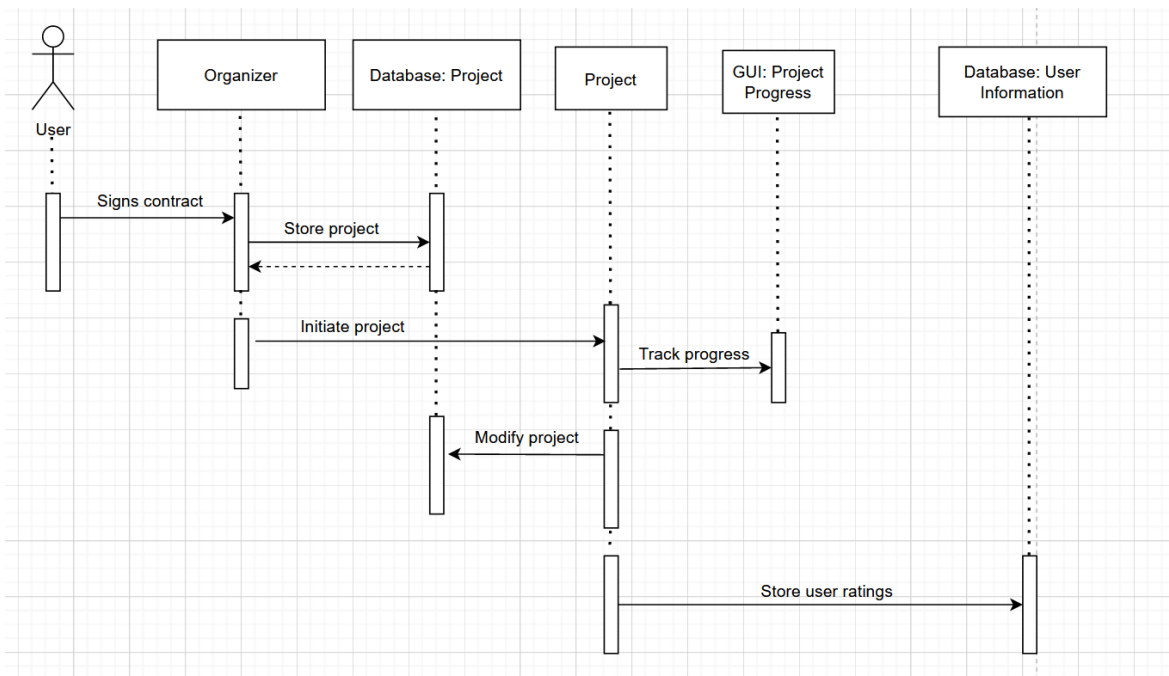
Figure 8(d): Message Sequence Chart Part 4

Figure 8(e): Message Sequence Chart Part 5

9. Data Specification:

In this part, we demonstrated the Entity-Relationship (E-R) Diagram of the entire database in figure 9.1 and the data definition in table 9.1.
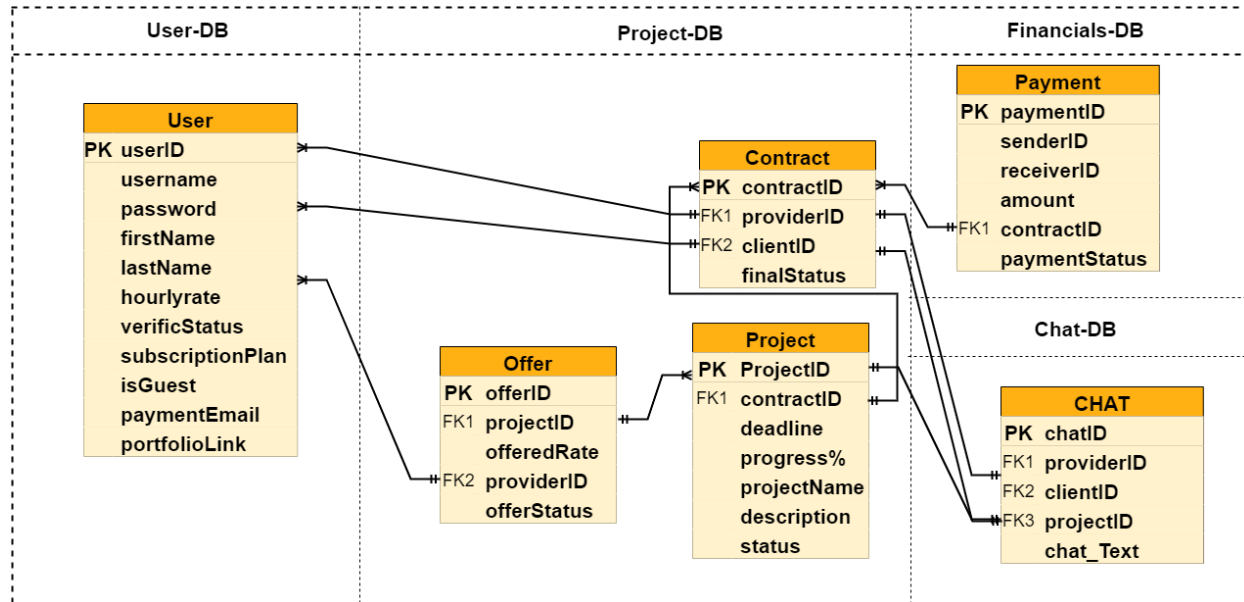

Figure 9.1: E-R Diagram of the Database

Database Name	Attribute	Data Type
User-DB	userID	INT
	username	VARCHAR
	password	VARCHAR
	firstName	VARCHAR
	lastName	VARCHAR
	hourlyRate	DOUBLE
	verificStatus	BOOL
	subscriptionPlan	VARCHAR
	isGuest	BOOL
	paymentEmail	VARCHAR
	portfolioLink	VARCHAR
Chat-DB	chatID	INT
	providerID	INT
	ClientID	INT
	projectID	INT
	Chat_Text	VARCHAR

	projectID	INT
	contractID	INT
	deadline	VARCHAR
	progress%	DOUBLE
	projectName	VARCHAR
	description	VARCHAR
Project-DB	status	VARCHAR
	providerID	INT
	clientID	INT
	finalStatus	VARCHAR
	offerID	INT
	offerStatus	VARCHAR
	offeredRate	DOUBLE
	paymentID	INT
	senderID	INT
Financials-DB	receiverID	INT
	amount	DOUBLE
	contractID	INT
	paymentStatus	VARCHAR

Table 9.1: Data Definition of the Entire database

10. Summary:

The Research Matchmaking Project consists of most of the features that are in the requirements list. The future work scopes for the most advanced features are described in the Future Scopes of Work section of this document. We tried to keep our system easy to understand and maximize the performance by planning to implement this with a minimum number of agents and a simple architecture. The overall system consists of the following elements:

- 6 Units
- 6 Agents
- 6 Goals
- 7 Roles
- 31 Tasks

11. Future Scopes of Work:

Though our project has a lot of conventional features, we want to add the following features in our future work.

- Global Payment Gateway Implementation for Visa, Mastercard, Amex, GPay, AliPay, Paypal, Discover, UnionPay, and etc.
- Artificial Intelligence (AI) based advanced search engine for the clients.
- Integrated cloud-based meeting platform for the Clients, Service Providers and all the contributors related to the project or research work.
- We provide an Artificial Intelligence (AI) based Personal Assistant bot for both Clients and Service Providers to help them maintain work timelines, meeting schedules, work schedules, reminders, etc.

12. Conclusion:

This document is the entire overview of our project, from the planning phase to the design and development phase. We've tried to describe and demonstrate the entire project in different small building blocks using the MaSE methodology. This Research Matchmaking project aims to cover up all the typical features. As we'll get a very short time to develop such a big project like this, some features are not included in our current development plan. We kept those features aside for future work.