Research Matchmaking Project

Agent-Based Software Engineering
(SEng 696)

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

System Specification & Design Document
(Assignment 1)

UNIVERSITY OF CALGARY

SCHULICH
School of Engineering

Introduction:

Research matchmaking is a web-based app to find individuals or organisations 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.

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 analysing 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.

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. It has two phases.

1) Analysis Phase
2) Design Phase

1. Analysis Phase:

The analysis phase is the primary planning and graphical representation of the entire system. This phase consists of three parts:

A.1 Capturing Goals
A.2 Applying Use Cases
A.3 Refining Roles

<u>A.1 Capturing Goals</u>:

In this section, we demonstrated the list of the goals and the Goal Hierarchy.
List of Goals for Research Matchmaking App:

1.0 Match providers (employees) and clients (employers)

- 1.1 **Create User Network**
  - 1.1.1 Get Sign Up Credentials
  - 1.1.2 Get User and Guest Login Credentials
  - 1.1.3 Manage User Profile
    - 1.1.3.1 Manage Provider Profile Information
    - 1.1.3.2 Manage Client Profile Information
    - 1.1.3.3 Provide Premium Provider Profile Tier
  - 1.1.4 Validate User
- 1.2 **Query User Data**
  - 1.2.1 Sort the Search Results
    - 1.2.1.1 Implement Default Sorting Method
  - 1.2.2 Manage Visible Data based on User Credentials
    - 1.2.2.1 Limit Guest's Data Queries
  - 1.2.3 Search Queries are Dynamic
- 1.3 **Contract Management**
  - 1.3.1 Create Project Proposal
    - 1.3.1.1 Provider can Accept or Reject Proposal
    - 1.3.1.2 View Provider/Client Rating
  - 1.3.2 Request a Provider
    - 1.3.2.1 Accept Specified Provider Price
    - 1.3.2.2 Make a Bid with a New Price
    - 1.3.2.2 Accept or Reject the Request/Bid
  - 1.3.3 Draft a Contract
    - 1.3.3.1 Accept or Reject the Contract
- 1.4 **Money Transfer**
  - 1.4.1 Process Money Payment
  - 1.4.2 Bill a Service Fee Cut
- 1.5 **Track Project Progress**
  - 1.5.1 Visual Indicator
  - 1.5.2 Manage Change Requests
    - 1.5.2.1 Accept/Reject Proposed Change
    - 1.5.2.2 Request a Deadline Change
  - 1.5.3 Provide Feedback on Completion

        - 1.5.3.1 Client Rate the Provider
        - 1.5.3.2 Provider Rate the Client
- 1.6 **Create Project Chat Room**
    - 1.6.1 Make Project Chat Room for Client and Provider

And the Goal Hierarchy is on the following:

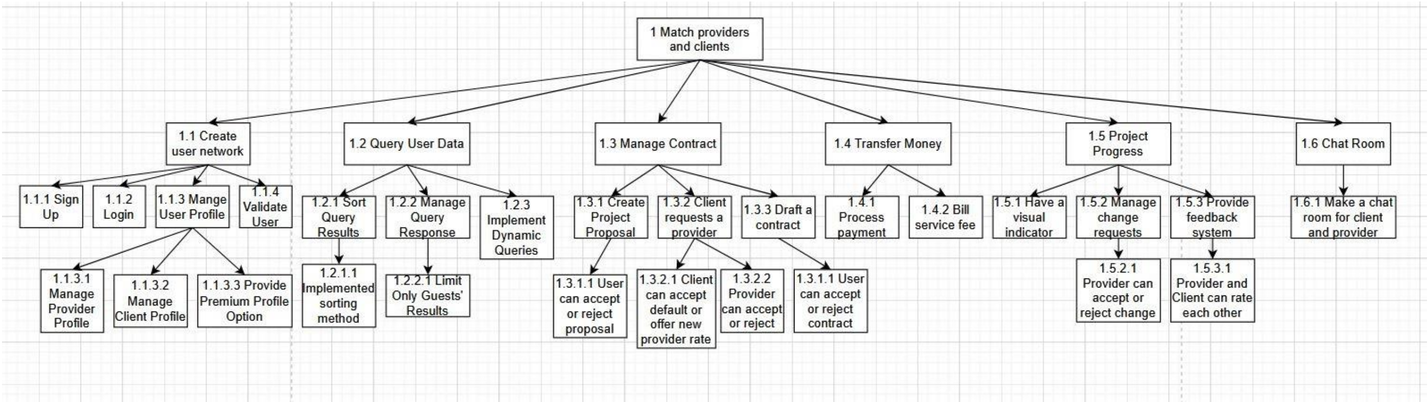
1 Match providers and clients
1.1 Create user network
1.1.1 Sign Up
1.1.2 Login
1.1.3 Mange User Profile
1.1.4 Validate User
1.1.3.1 Manage Provider Profile
1.1.3.2 Manage Client Profile
1.1.3.3 Provide Premium Profile Option
1.2 Query User Data
1.2.1 Sort Query Results
1.2.1.1 Implemented sorting method
1.2.2 Manage Query Response
1.2.2.1 Limit Only Guests' Results
1.2.3 Implement Dynamic Queries
1.3 Manage Contract
1.3.1 Create Project Proposal
1.3.1.1 User can accept or reject proposal
1.3.2 Client requests a provider
1.3.2.1 Client can accept default or offer new provider rate
1.3.2.2 Provider can accept or reject
1.3.3 Draft a contract
1.3.1.1 User can accept or reject contract
1.4 Transfer Money
1.4.1 Process payment
1.4.2 Bill service fee
1.5 Project Progress
1.5.1 Have a visual indicator
1.5.2 Manage change requests
1.5.2.1 Provider can accept or reject change
1.5.3 Provide feedback system
1.5.3.1 Provider and Client can rate each other
1.6 Chat Room
1.6.1 Make a chat room for client and provider

Figure A.1.1: Goal Hierarchy

A.2 Applying Use Cases:

In this section, we demonstrated all the goals in the sequence diagram. For every goal, we've added an individual sequence diagram for better understanding.

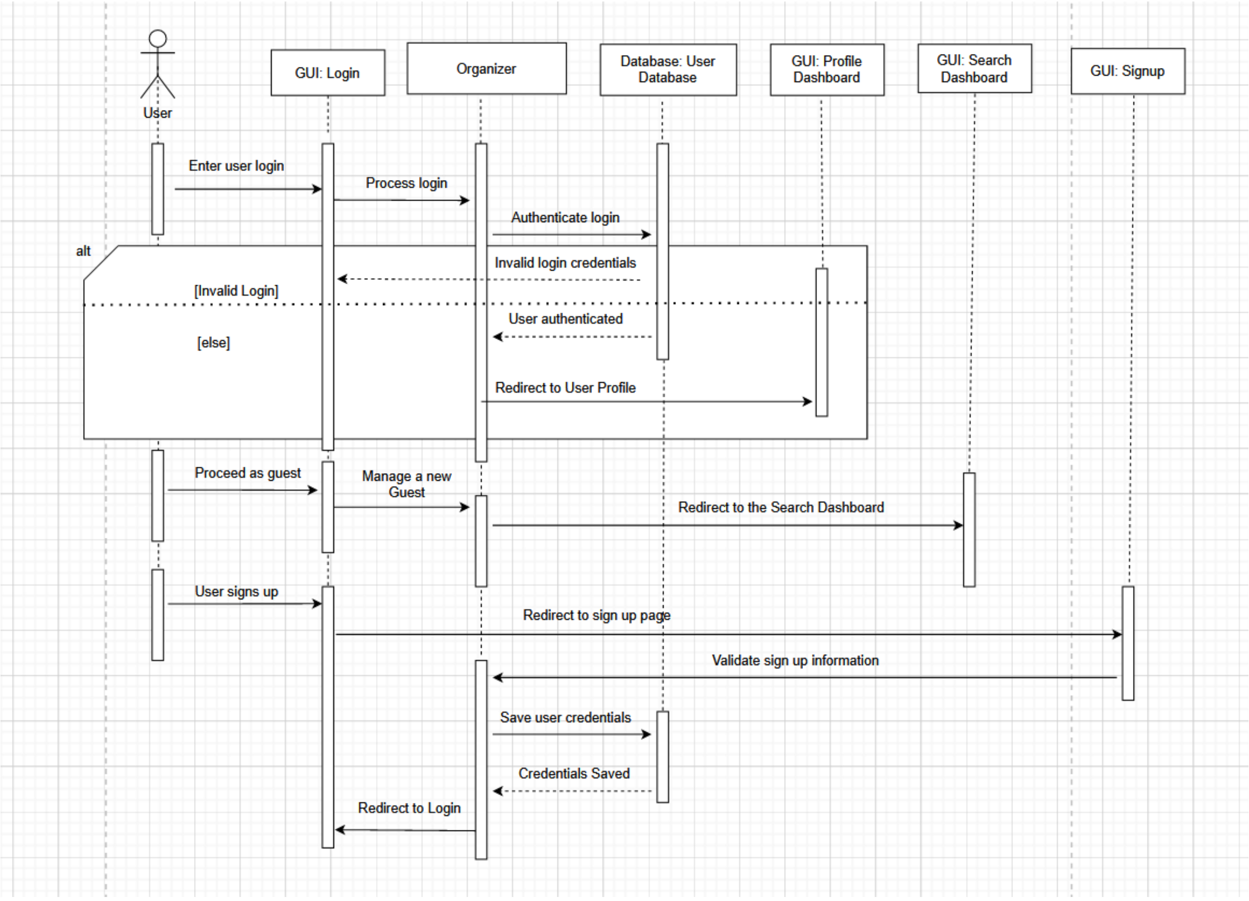

Figure A.2.1: Create User Network (Goal 1)

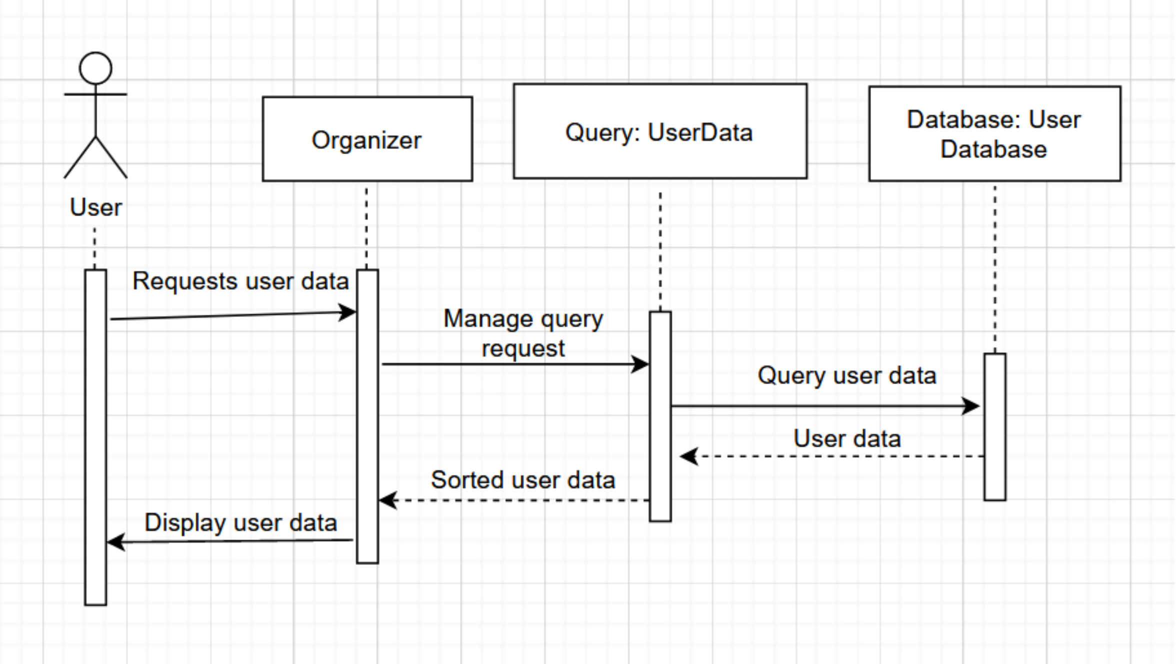

Figure A.2.2: Search User Data (Goal 2)

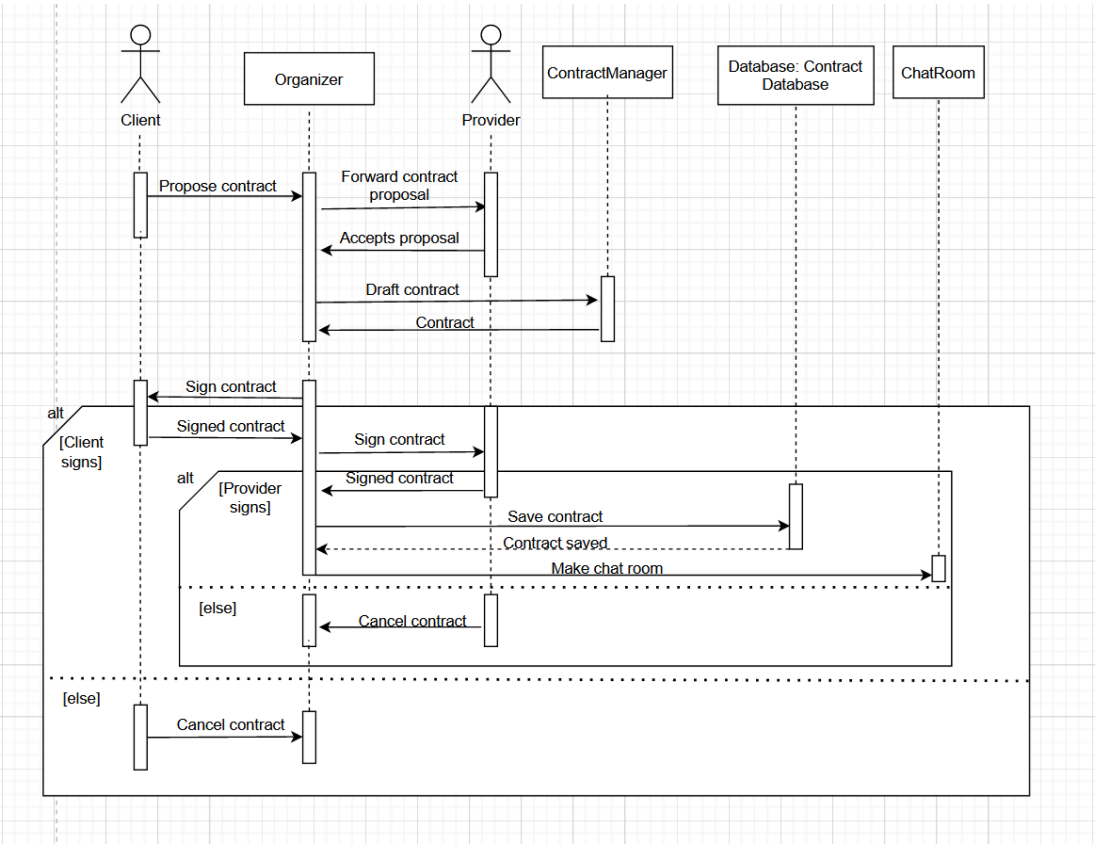

Figure A.2.3: Contract Management and Chatroom (Goal 3 and Goal 6)

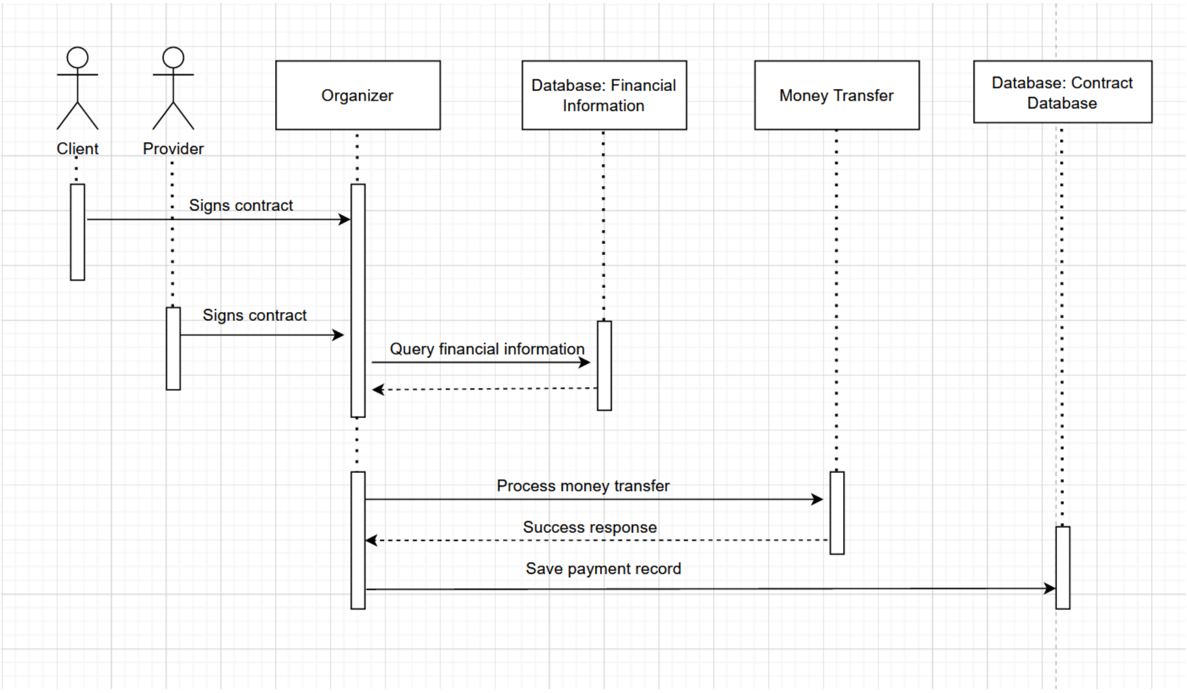

Figure A.2.4: Money Transfer (Goal 4)

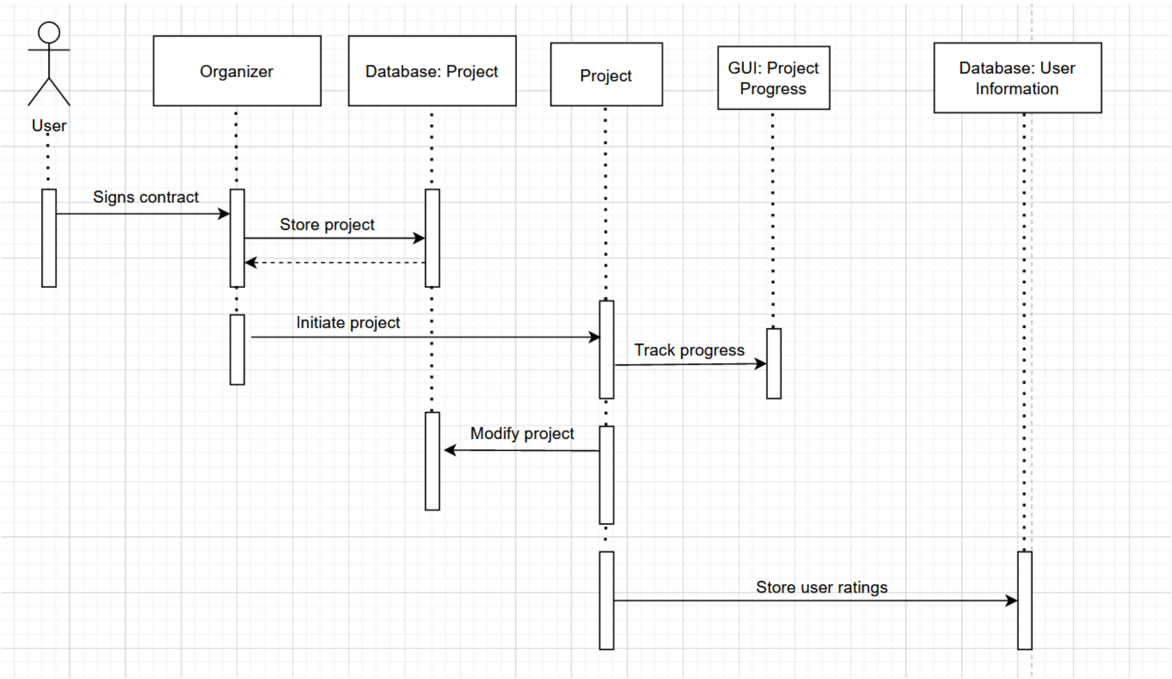

Figure A.2.5: Track Project Progress (Goal 5)

A.3 Refining Roles:

Multiple roles are associated with this project to communicate, manage, contract, organize, pay, receive the money, and, after all, successfully complete a deal. The roles are:

1. Interface
2. Organizer
3. Contract
4. Search
5. Project
6. Chat
7. Payment

We’ve added Role Model Tables with associated tasks lists for each Role on the following:

Role Name: **Interface**	
Sign Up Request	SignUpUser()
Login Request	LoginUser()
Guest Access	AccessAsGuest()
Manage User Profile	ManageUserProfile()
Propose Project for a Provider	ProposeProviderProject()
Visualize Project Progress	VisualizeProjectProgress()

Table A.3.1: Role Model Table with Associated Task Lists of Interface Role

Role Name: **Organizer**	
Set Premium Provider	SetPremiumProvider()
Manage Query Response	ManageQueryResponse()
Guest Access	AccessAsGuest()
Manage User Profile	ManageUserProfile()
Notify Provider	NotifyProvider()
Initialize new Project	InitializeNewProject()
Update User Credentials	UpdateUserCredentials()
Validate User Credentials	ValidateUserCredentials()
Initialize Payment	InitializePayment()

Table A.3.2: Role Model Table with Associated Task Lists of Organizer Role

Role Name: **Contract**	
Draft Contract	DraftNewContract()
Finalize Contract	FinalizeContract()
Propose Project	ProposeProject()
Save Project Record	SaveProjectRecord()

Table A.3.3: Role Model Table with Associated Task Lists of Contract Role

Role Name: **Search**	
Sort Results	SortResults()
Check User Credentials	CheckUserCredentials()
Set Visibility	SetVisibility

Table A.3.4: Role Model Table with Associated Task Lists of Search Role

Role Name: **Project**	
Mange Change	ManageProjectChange()
Track Project Progress	TrackProjectProgress()
Rate Client	RateClient()
Rate Provider	RateProvider()
Propose Deadline Change	ProposeDeadlineChange()

Table A.3.5: Role Model Table with Associated Task Lists of Project Role

Role Name: **Chat**	
Create Chat Room	CreateChatRoom()

Table A.3.6: Role Model Table with Associated Task Lists of Chat Role

Role Name: **Payment**	
Process Payment	ProcessPayment()
Read Billing Info	ReadBillingInfo()
Bill Service Fee	BillServiceFee()

Table A.3.7: Role Model Table with Associated Task Lists of Payment Role

The Concurrent Task Diagram is demonstrated in the figure A.3.8.

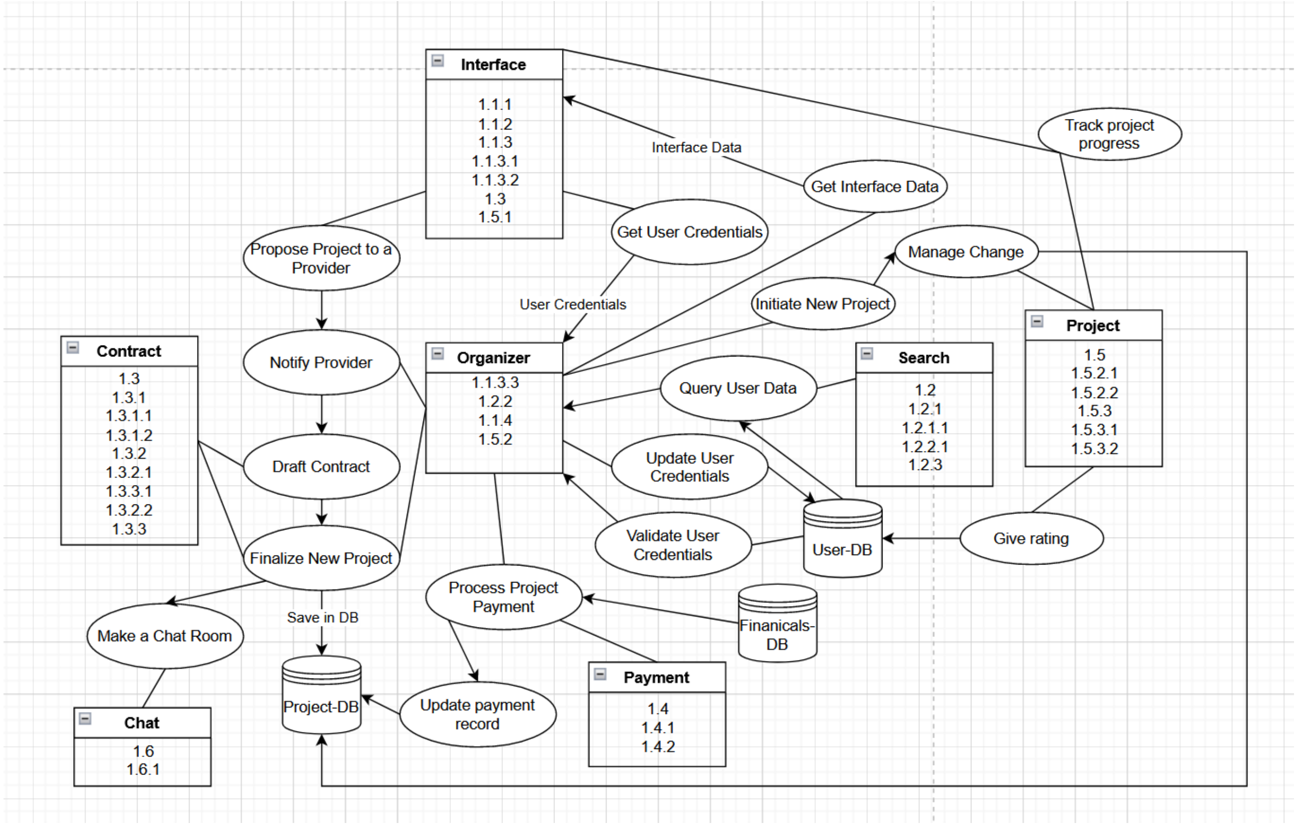

Figure A.3.8: Concurrent Task Diagram

2. Design Phase:

The design phase is the final model and workflow to implement the project in the real world. Typically, the design phase consists of four parts: creating agent classes; constructing conversations; assembling agent classes; and system design. In our project, we discussed and demonstrated the following design sections:

D.1 Agent Classes

D.2 Agents' Architecture

D.3 Deployment Diagram

D.1 Agent Classes:

We've demonstrated the Agent Classes in the Figure D.1.1.

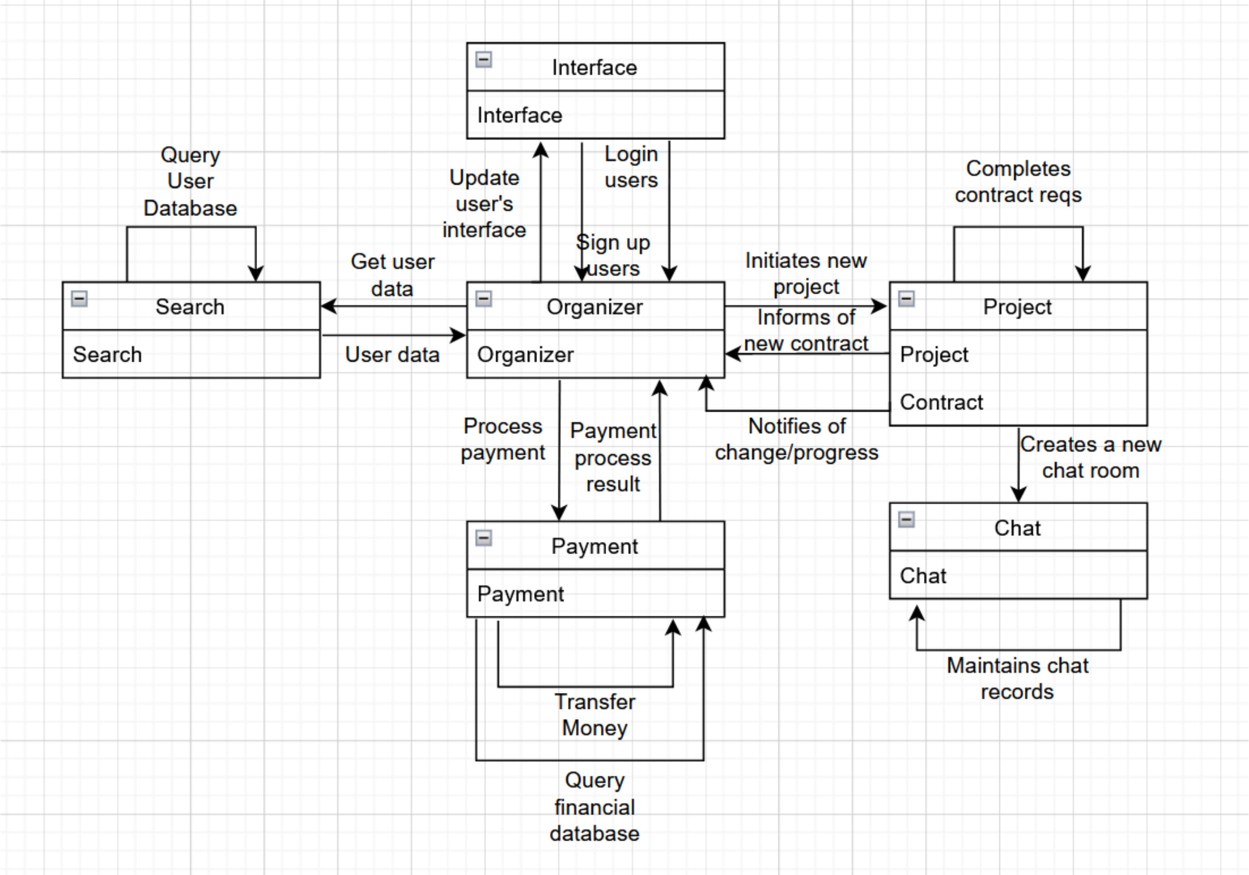

Figure D.1.1: Agent Classes

D.2 Agents' Architecture:

The following Agents' Architecture is the primary assumption based on our design. Please note that, it may be changed during the development process due to optimization and to increase the efficiency and performance of the overall system.

In this section, we demonstrated all the agents' architecture. For every agent, we've added individual figures for better understanding.

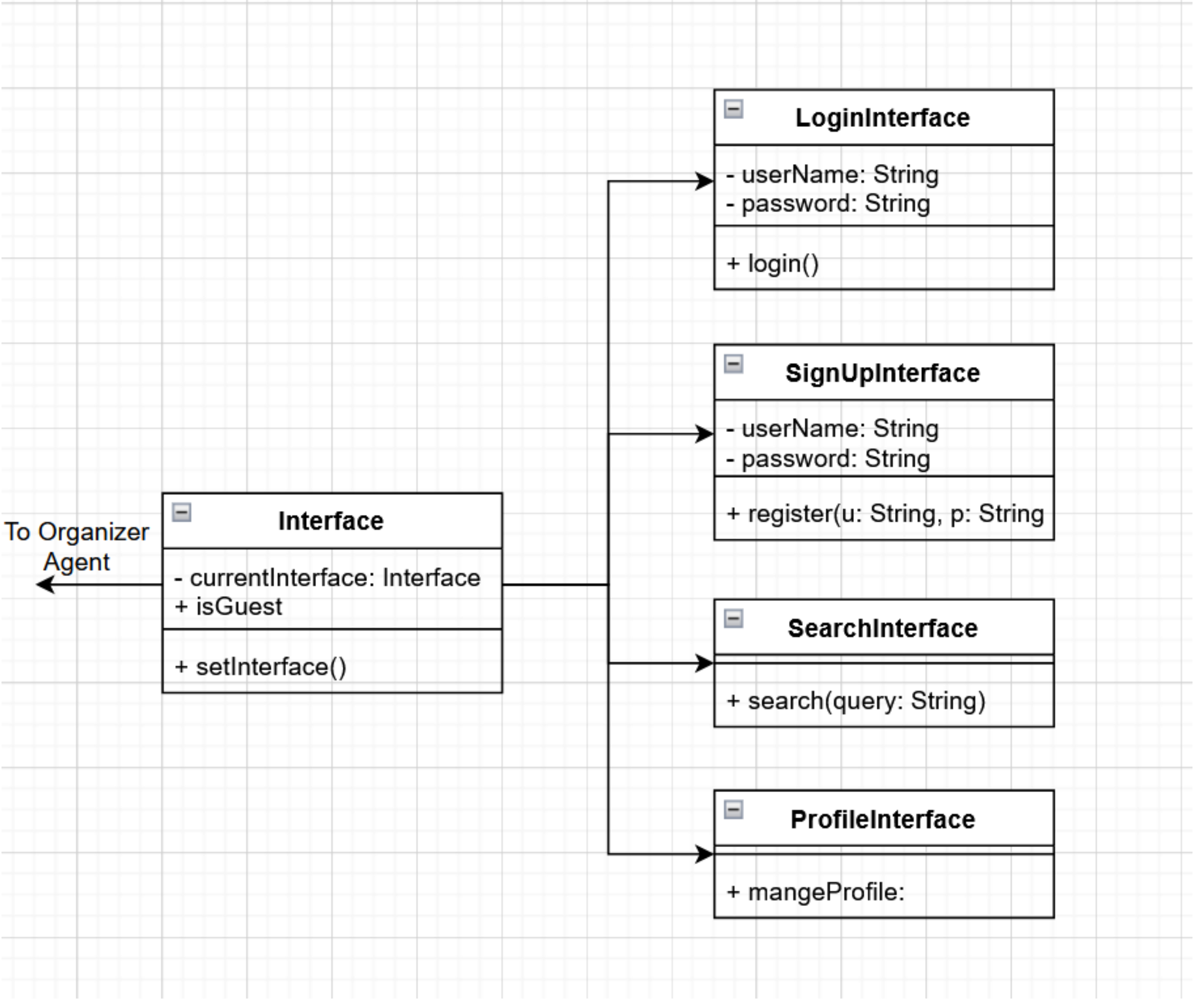

Figure D.2.1: Architecture of Interface Agent

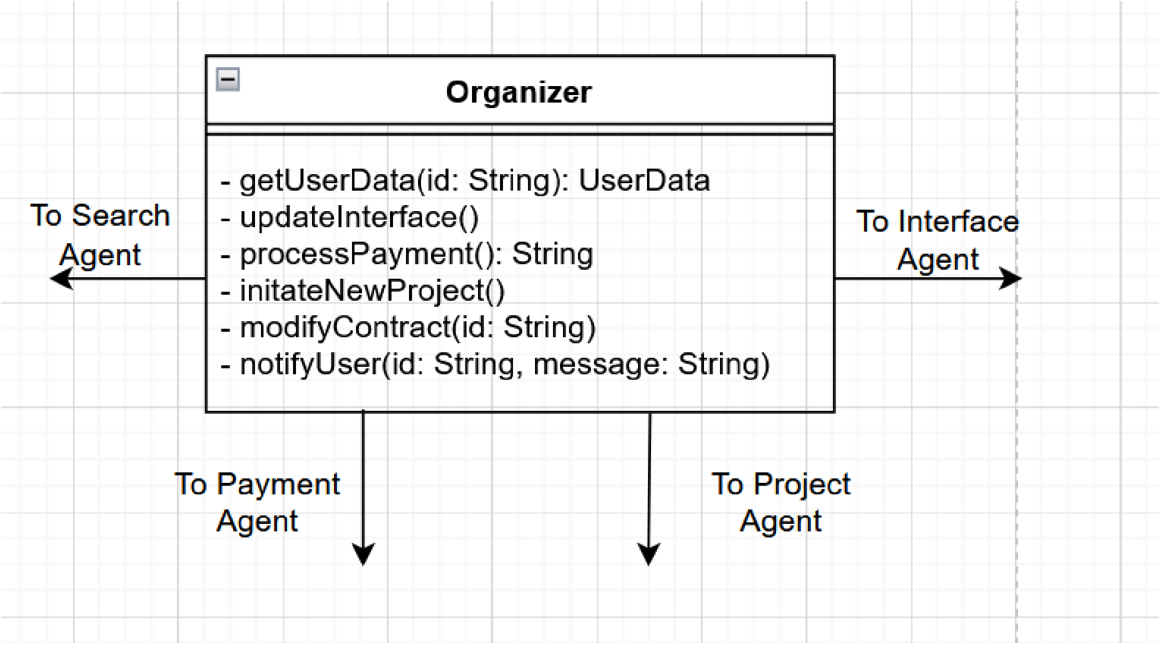
Organizer
- getUserData(id: String): UserData
- updateInterface()
- processPayment(): String
- initateNewProject()
- modifyContract(id: String)
- notifyUser(id: String, message: String)
To Search Agent
To Interface Agent
To Payment Agent
To Project Agent

Figure D.2.2: Architecture of Organizer Agent

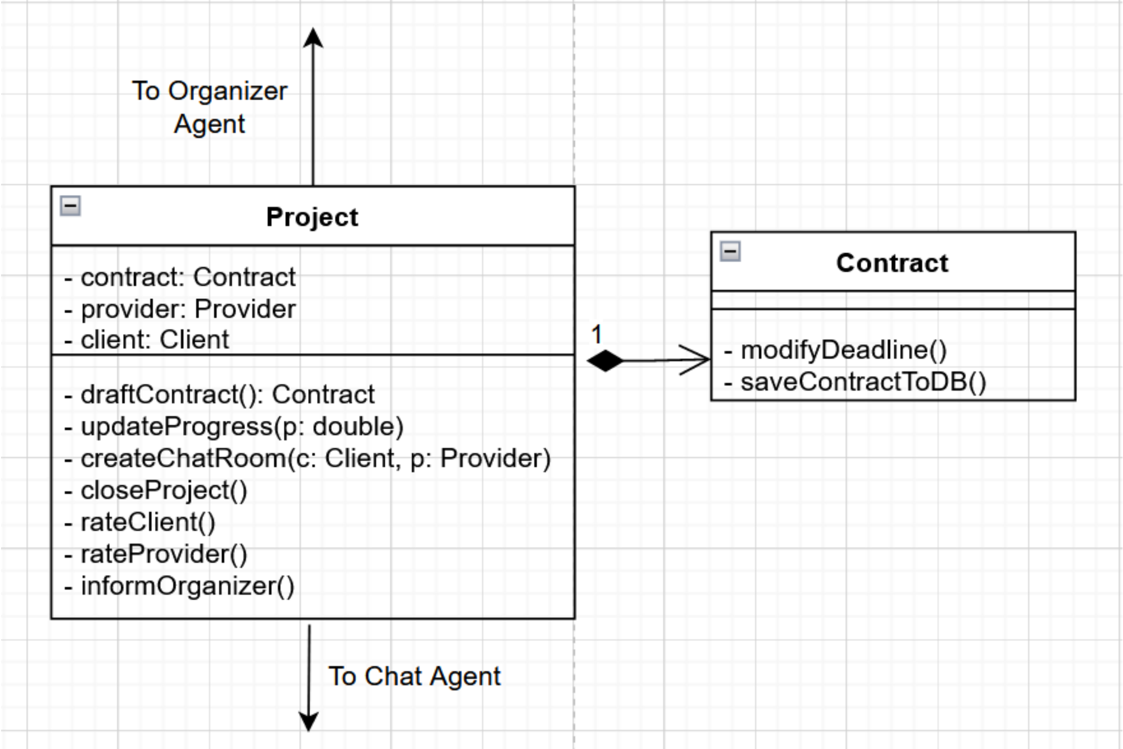

Figure D.2.3: Architecture of Project Agent

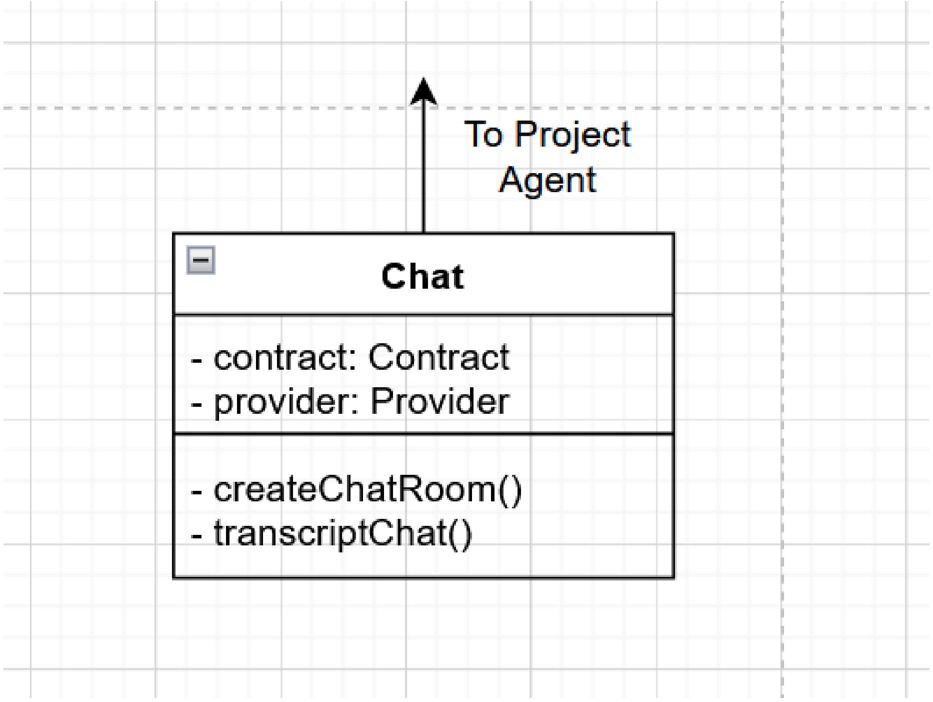

Figure D.2.4: Architecture of Chat Agent

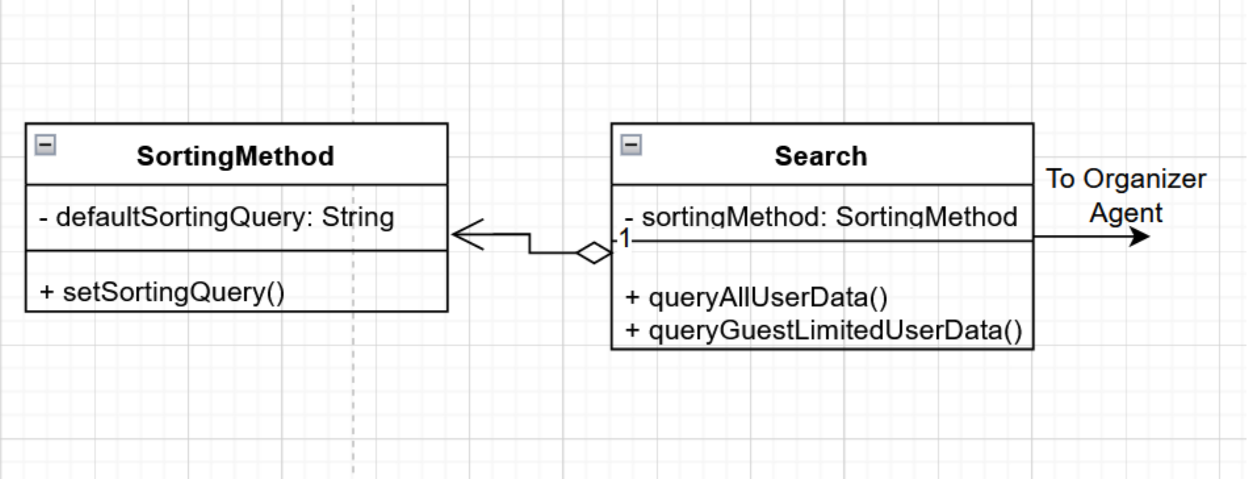
SortingMethod
- defaultSortingQuery: String
+ setSortingQuery()
Search
- sortingMethod: SortingMethod
1
+ queryAllUserData()
+ queryGuestLimitedUserData()
To Organizer Agent

Figure D.2.5: Architecture of Search Agent

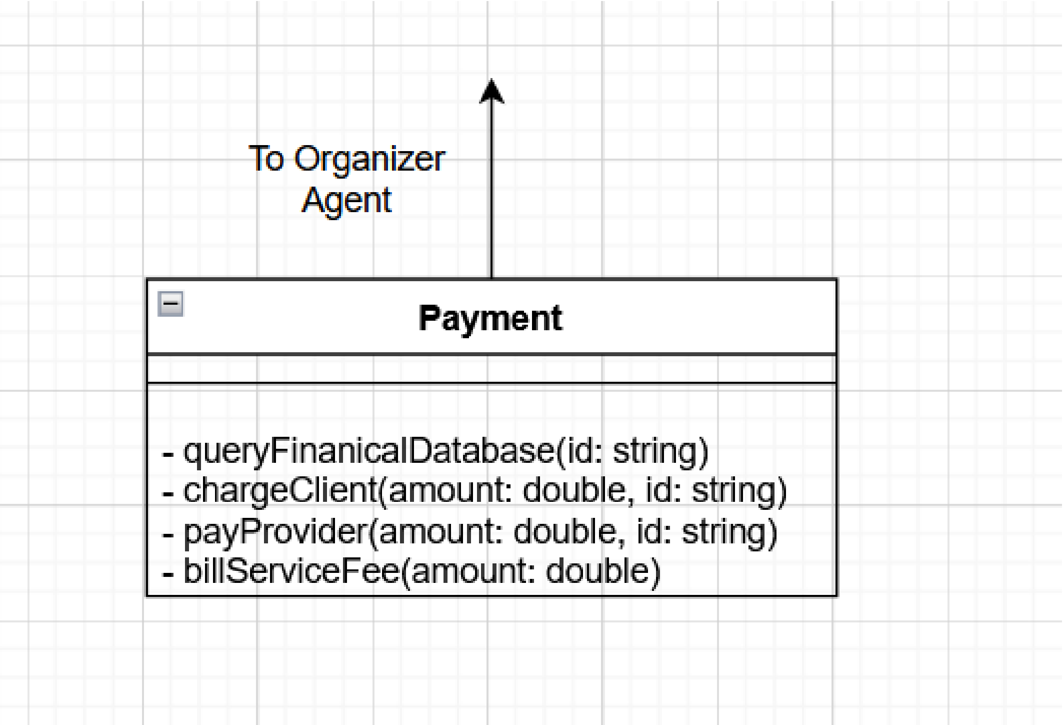
To Organizer
Agent
Payment
- queryFinanicalDatabase(id: string)
- chargeClient(amount: double, id: string)
- payProvider(amount: double, id: string)
- billServiceFee(amount: double)

Figure D.2.6: Architecture of Payment Agent

D.3 Deployment Diagram:

Here is the overall primary deployment diagram of our project. We'll modify this diagram if any changes are needed due to optimization and increase in efficiency and performance of the final project. We'll update this part later if any changes occur. Please find the deployment diagram in figure D.3.1.

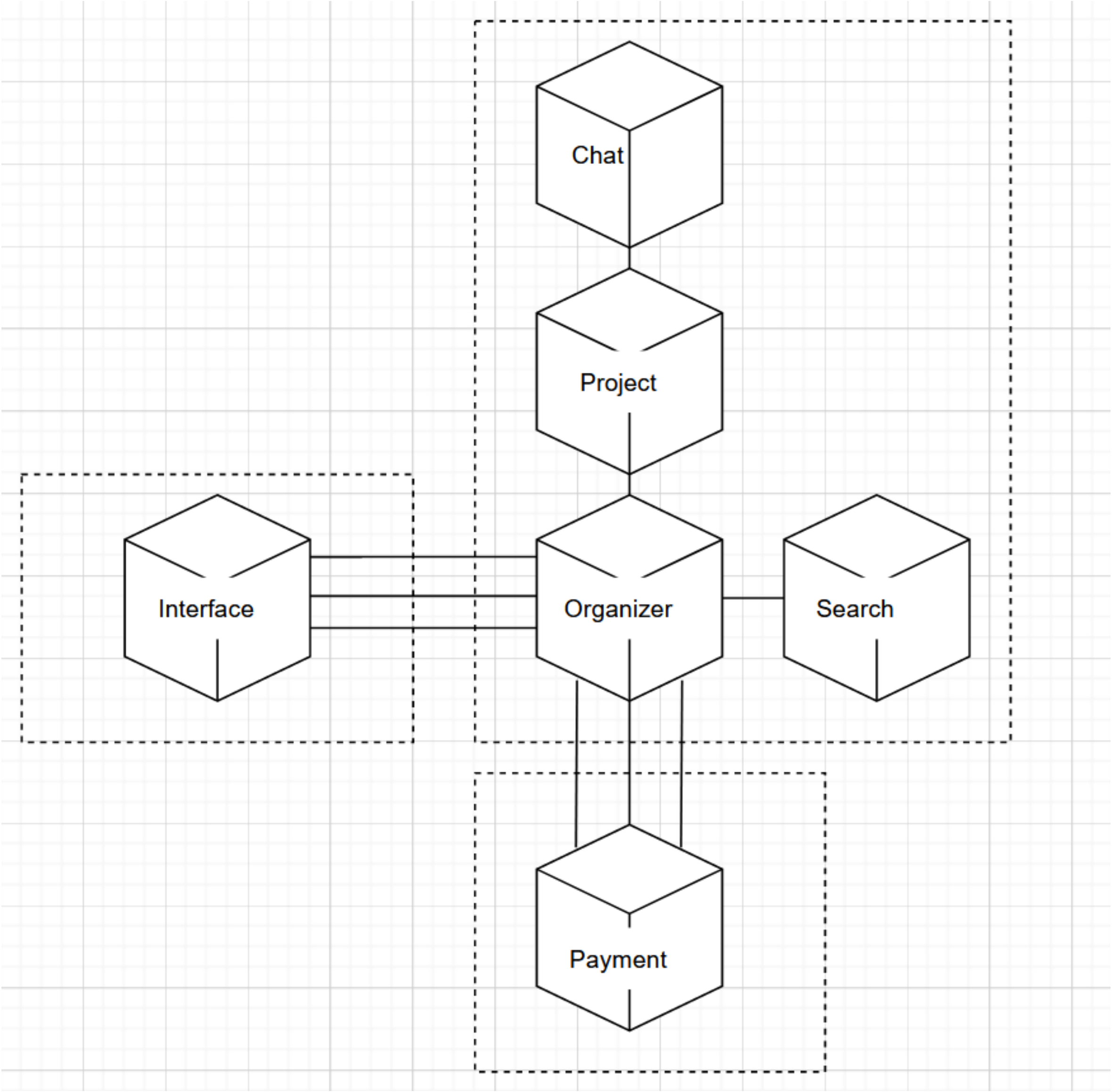

Figure D.3.1: Agent System Design - Deployment Diagram

Summary of Research Matchmaking Project:

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

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.

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.