Research Matchmaking Project

Agent-Based Software Engineering
(SENG 696)

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

System Specification & Design Document

UNIVERSITY OF CALGARY

SCHULICH
School of Engineering

Introduction:

Research matchmaking is a web-based app to find individual or organization of researchers, contributors, experts, etc. for a specific project or research work with similar interest 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 to find 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 clients project or research work. In this project we'll develop a complete research matchmaking system which will help clients to find service providers and vice-versa.

System Specification:

In this research matchmaking project, a Graphical User Interface (GUI) will help the both clients and the service providers to create their user account/sign up and login in the system. 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 already got 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 link, special keywords, service fee, etc. This information will help the clients to 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 able to search for the service providers with specific keyword or find one or more service providers from the database. But the guest client can't place bid for any service as well as can't see the service fee of any service provider. The fully signed-up clients and providers can see the entire system and use the app. The providers can choose either regular basic plan or a premium plan during the contract with the system. Premium subscribers will be appeared at the top of a search results, then eventually the basic verified service providers when a client will search for service providers for a specific service. Thus, premium service providers will get more reach to the clients which will help them to increase their earnings. The clients can place bid requesting service providers. The service provider can accept or reject the bid. During a rejection, the client will be notified directly. In accepting a bid, the system will generate a contract letter and send the letter to both client and provider. The provider and the client will get chance to review the contact letter and accept/reject the contract. Upon acceptance, all types of payment and financial transaction will be handled by the system to make the overall app reliable and secured. The system will receive 30% fee of any transaction or payment. Both the client and provider can monitor the progress of a project or research during the timeline. App can estimate the tentative deadline 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 rating and feedback to the providers upon their quality, time of completion and expertise, etc. These ratings will be displayed in the providers profile. These are the features and specification of the app.

Development Methodology:

We've selected MaSE Methodology for our system design. MaSE is one of the most advanced methodology to develop a multi agent-based software or system. 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

Analysis

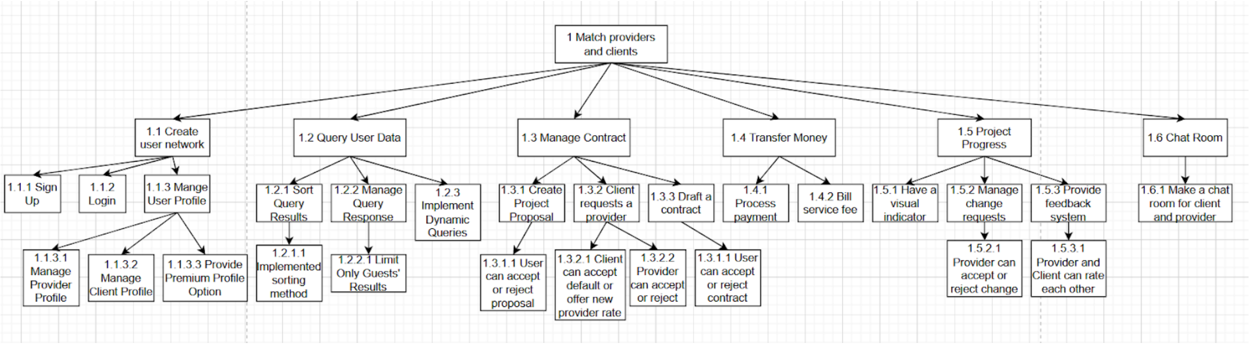
A1) List of Goals and Goal Hierarchy

A1.1) List of Goals

Takes care of reqs: 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15/16/17, 18, 19, 20, 21, 22, 24,

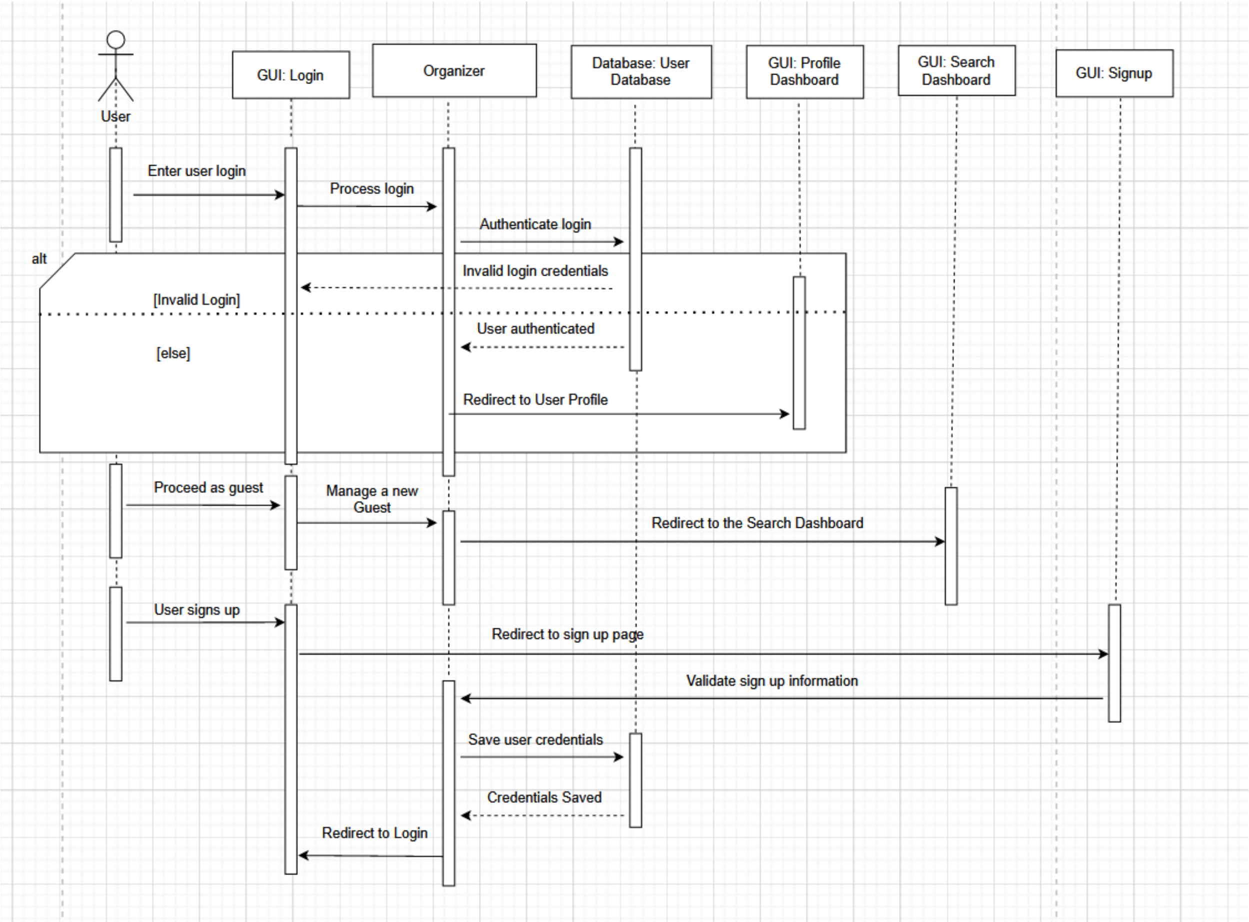
- 1.0 Match providers (employees) and clients (employers)
    - 1.1 Create **User Network**
        - 1.1.1 Get Sign Up Credentials
        - 1.1.2 Get User and Guest Login Credentials
        - 1.1.3 Manage User Profile
            - 1.1.3.1 Manage Provider Profile Information
            - 1.1.3.2 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

A1.2) Goal Hierarchy

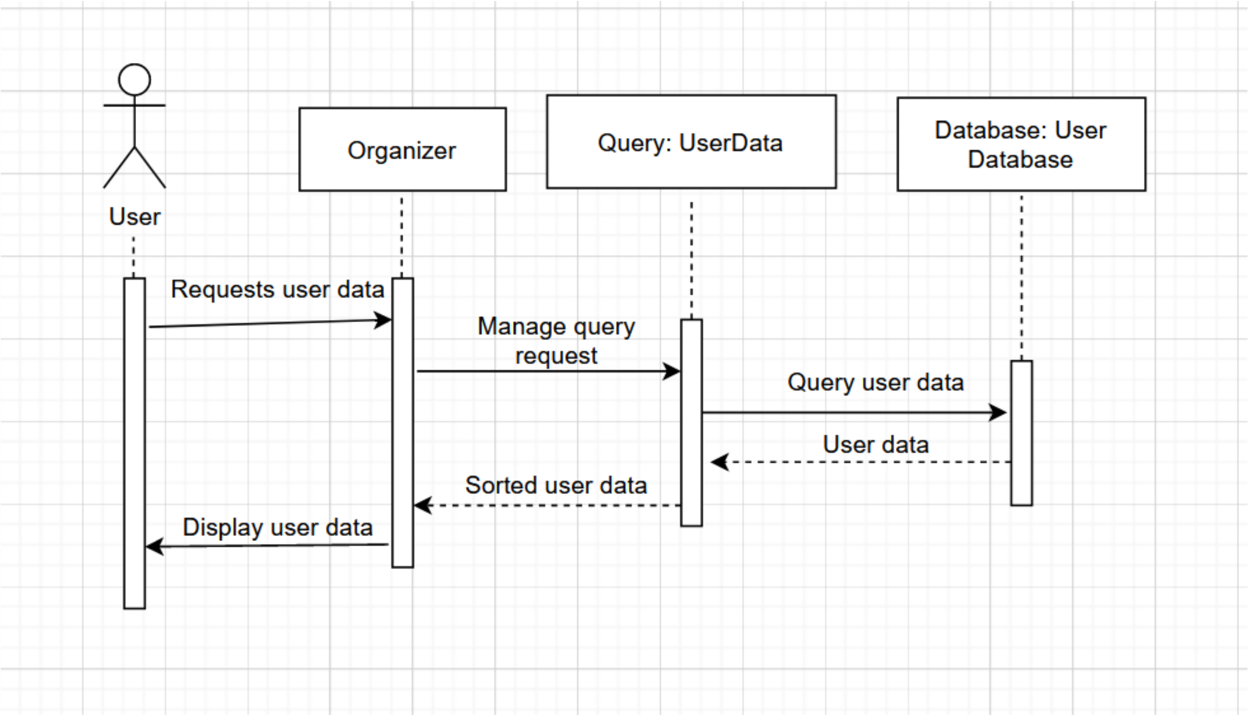

1 Match providers and clients
1.1 Create user network
1.2 Query User Data
1.3 Manage Contract
1.4 Transfer Money
1.5 Project Progress
1.6 Chat Room
1.1.1 Sign Up
1.1.2 Login
1.1.3 Mange User Profile
1.1.3.1 Manage Provider Profile
1.1.3.2 Manage Client Profile
1.1.3.3 Provide Premium Profile Option
1.2.1 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
1.3.1 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
1.4.1 Process payment
1.4.2 Bill service fee
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
1.6.1 Make a chat room for client and provider

A2) Sequence Diagrams

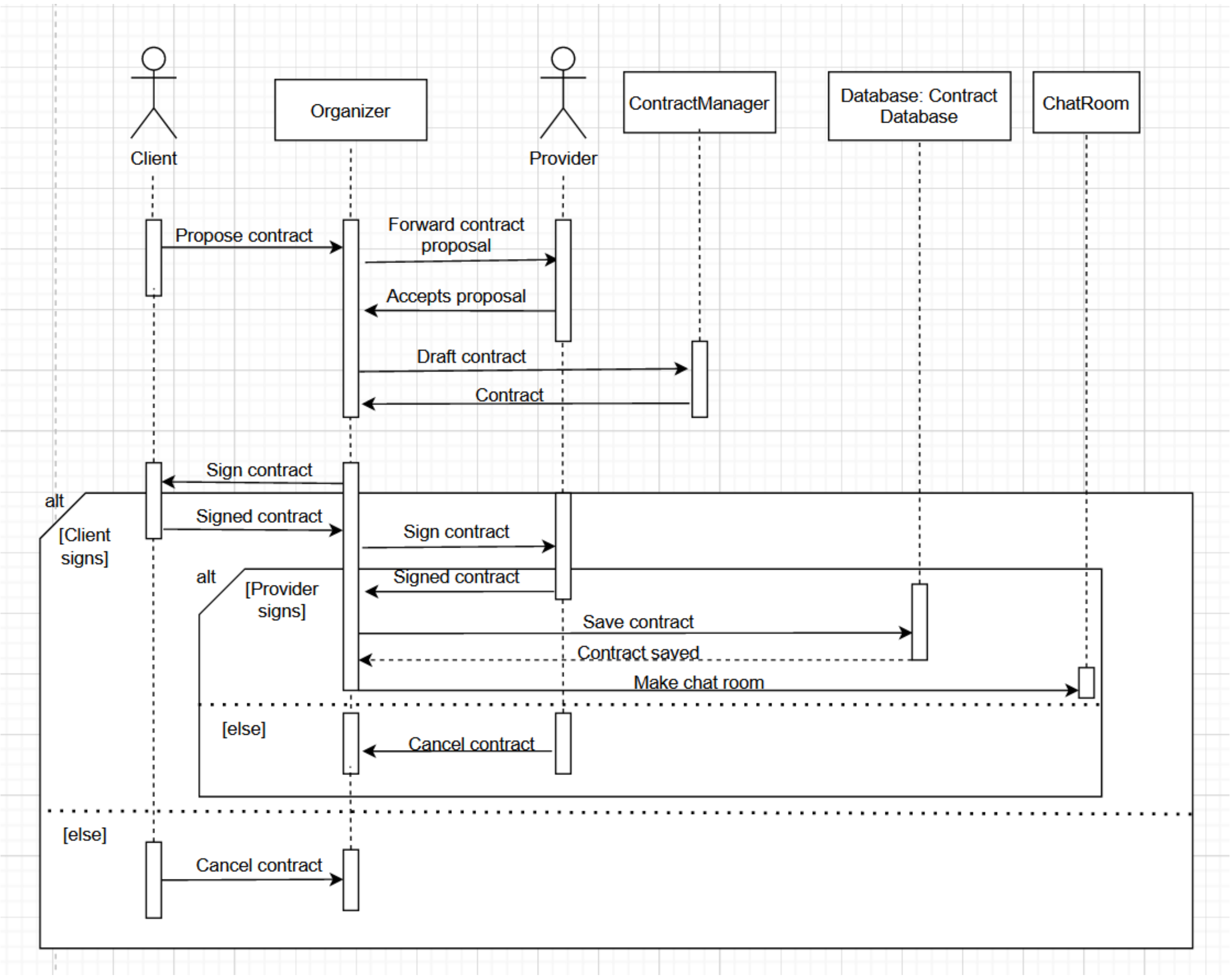
A2.1) Goal 1: Create User Network - Sequence Diagram



A2.2) Goal 2: Search User Data - Sequence Diagram

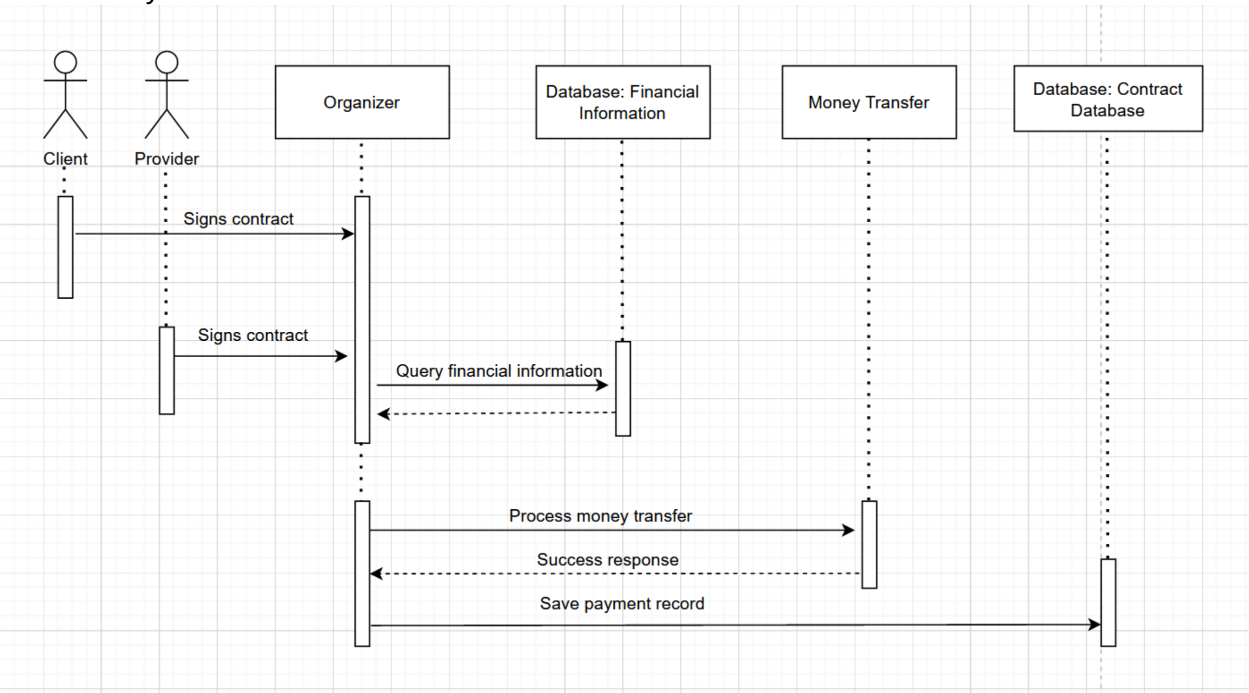


A2.3) Goal 3: Contract Management and Goal 6: Chat Room - Sequence Diagram

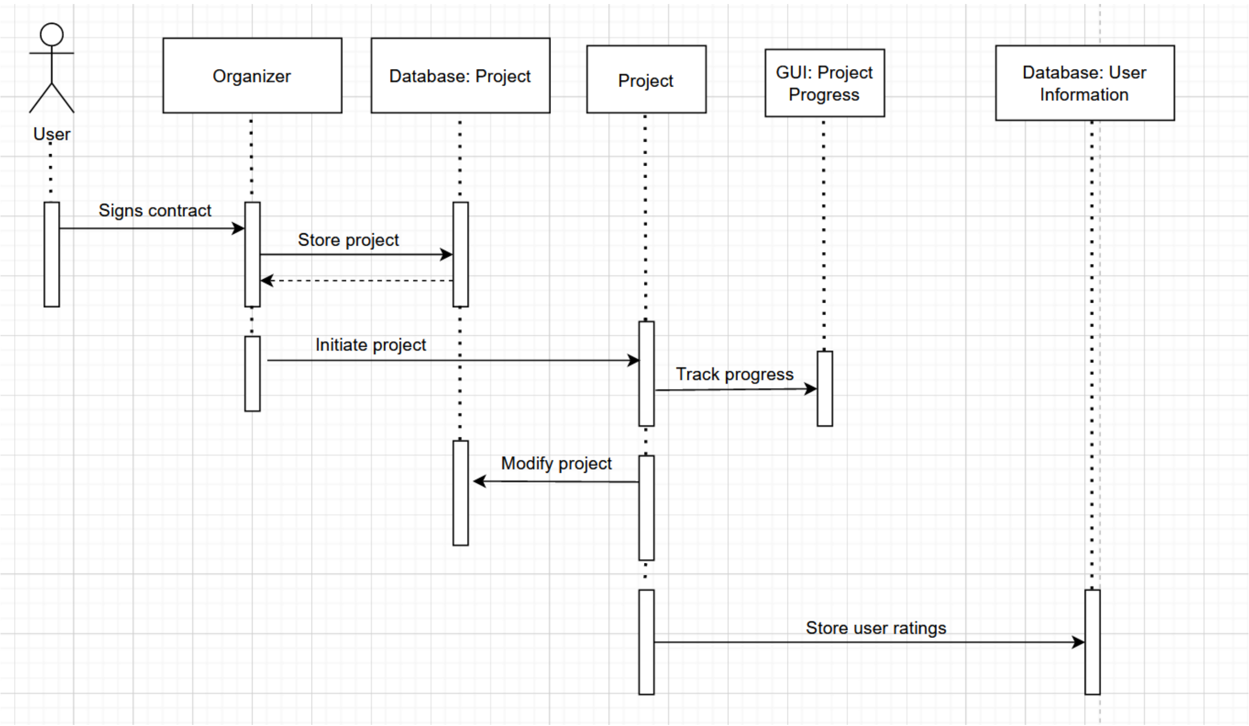


A2.4) Goal 4: Money Transfer

This goal is directly tied to goal 3 above (Contract management) but is shown separately for better clarity.



A2.5) Goal 5: Track Project Progress



A3) Refining Roles

A3.1) Role Model Table

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

Role Name	**Organizer**
Associated Tasks	
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()

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

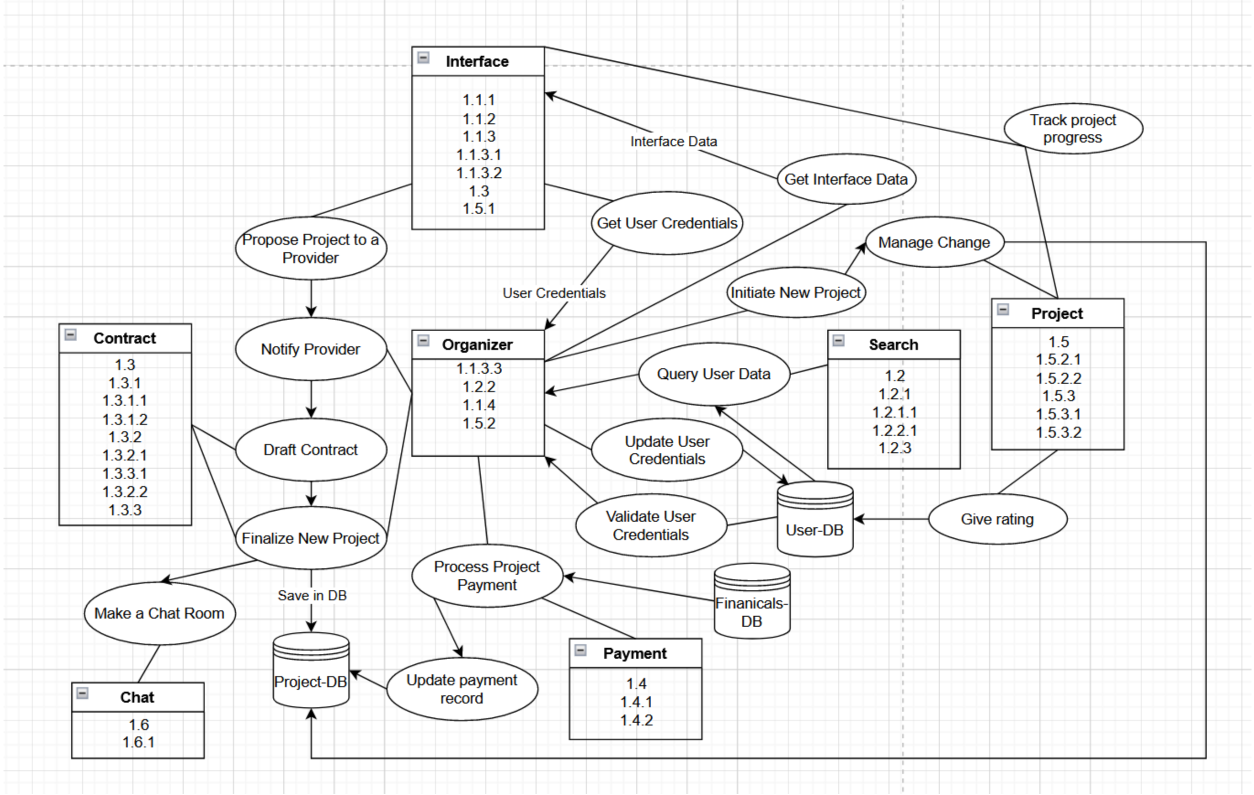
Role Name	**Search**
Associated Tasks	
Sort Results	SortResults()
Check User Credentials	CheckUserCredentials()
Set Visibility	SetVisibility

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

Role Name	**Chat**
Associated Tasks	
Create Chat Room	CreateChatRoom()

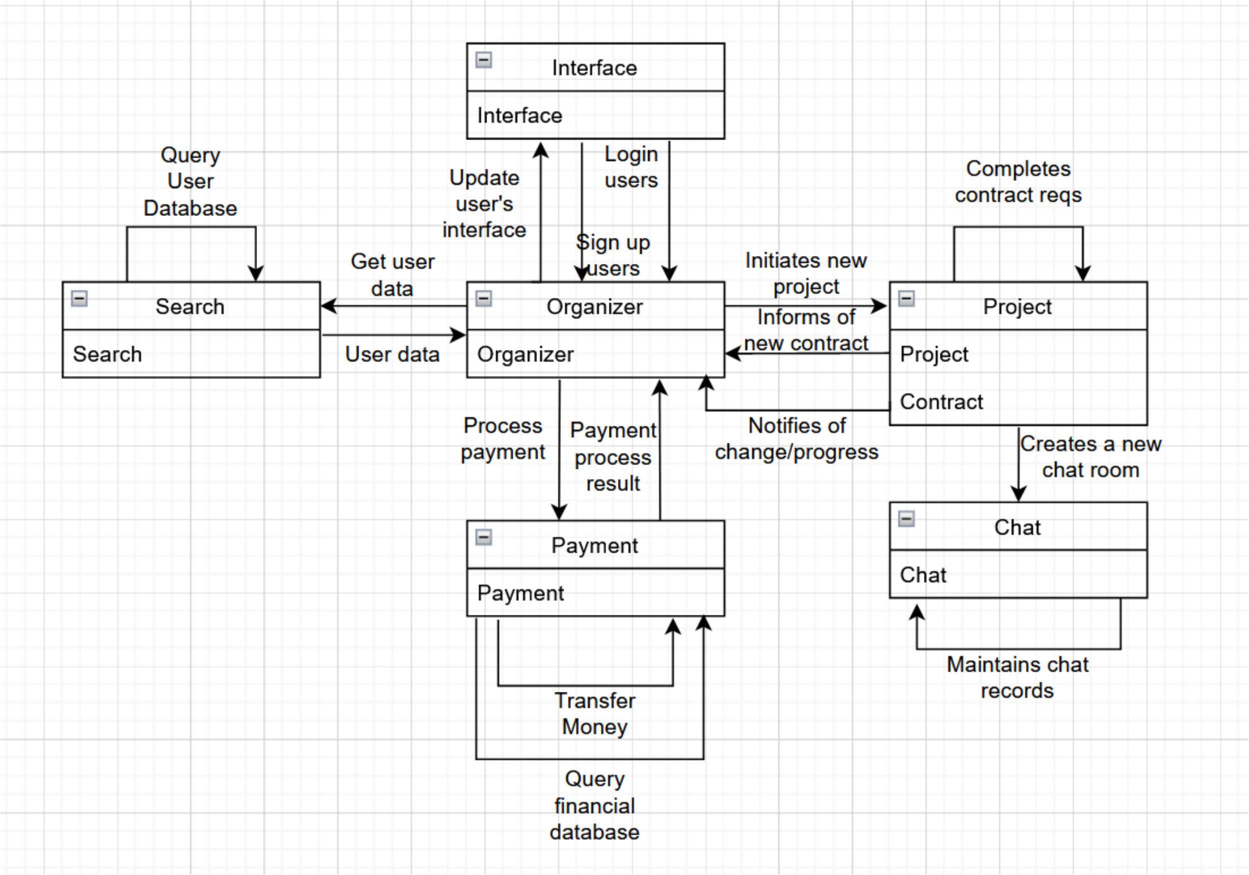
Role Name	**Payment**
Associated Tasks	
Process Payment	ProcessPayment()
Read Billing Info	ReadBillingInfo()
Bill Service Fee	BillServiceFee()

A3.2) Concurrent Task Diagram



Design

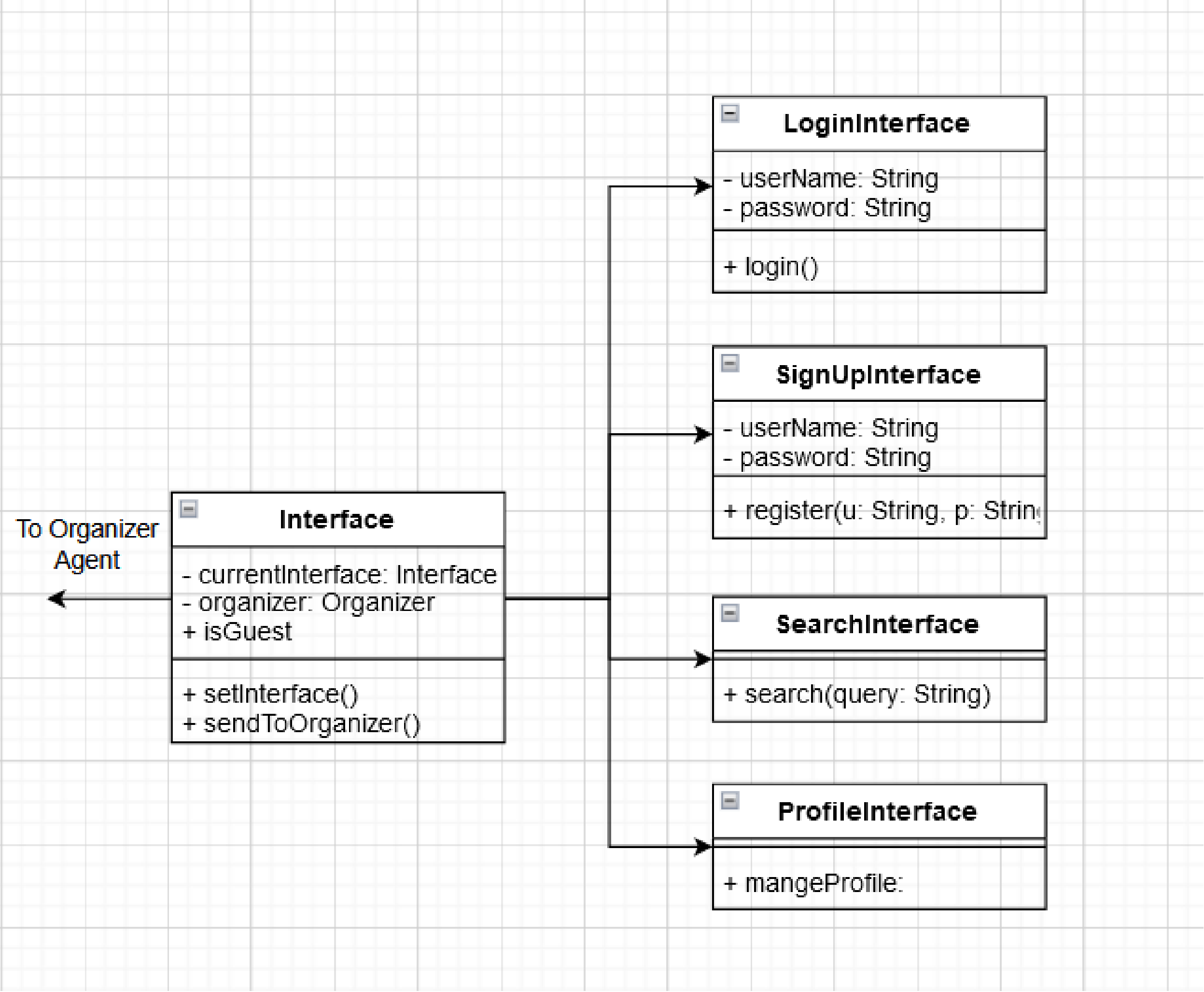
D1) Agent Classes



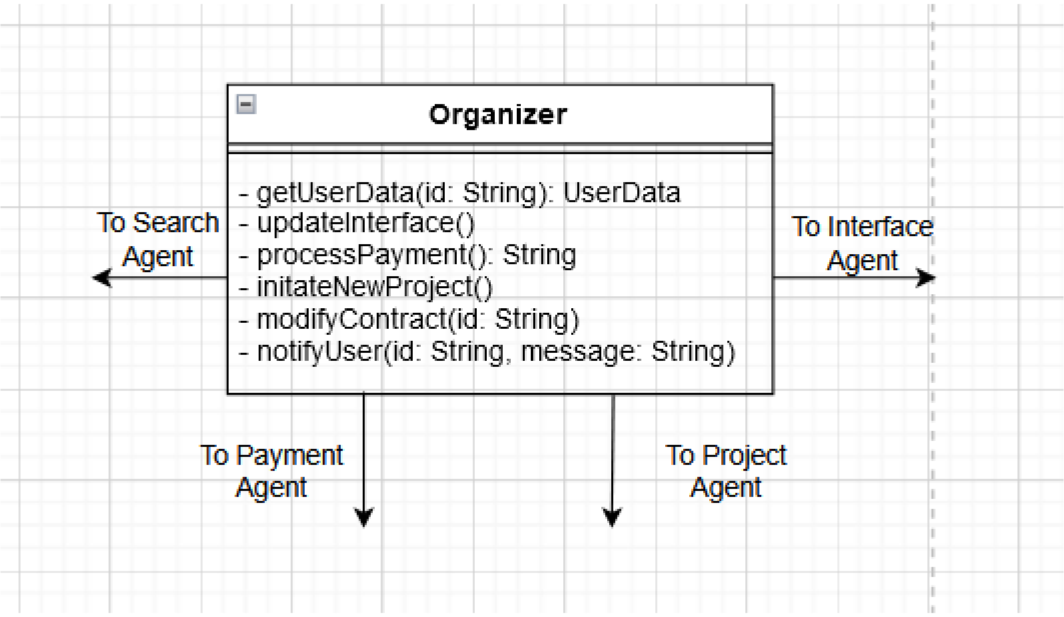
D2) Agent Architecture

Note that the agent architecture shown below is subject to change and will most definitely change as it becomes more detailed and optimized throughout the development progress.

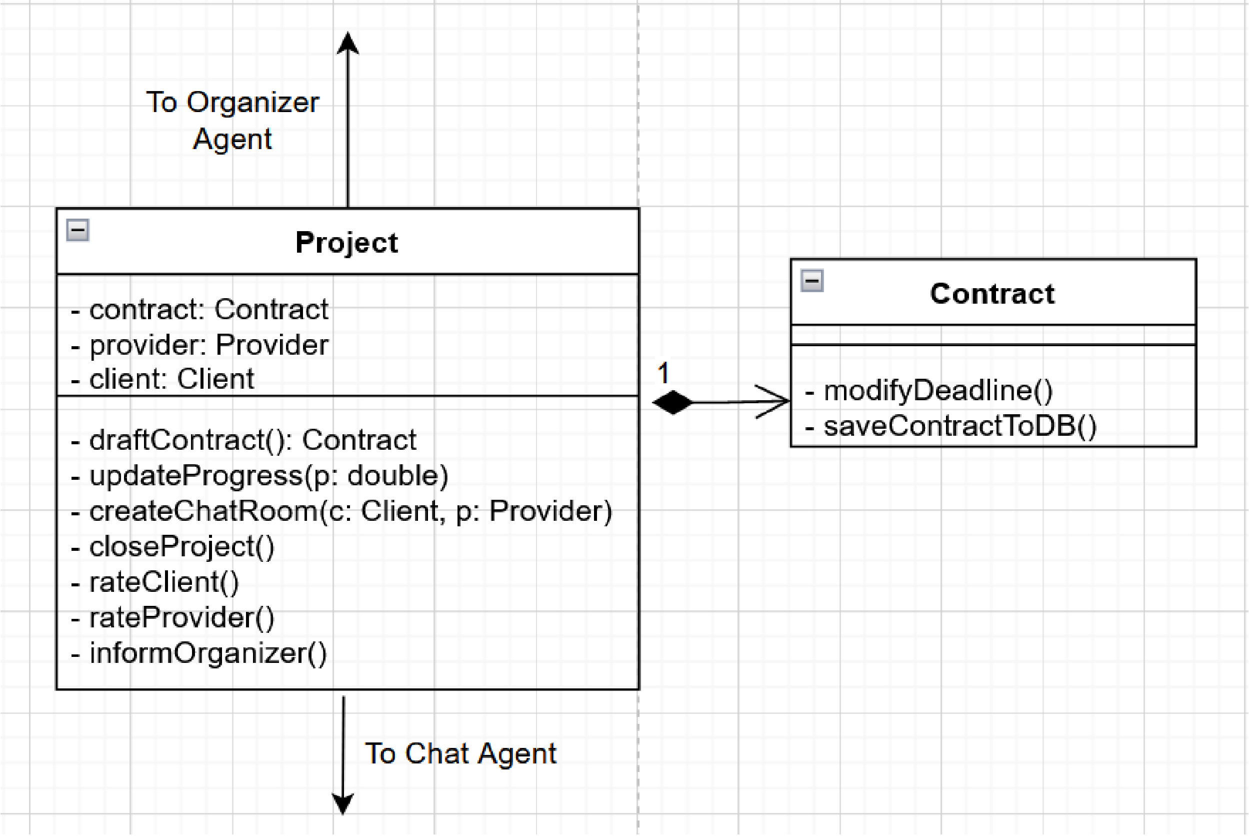
D2.1) Interface Agent Architecture



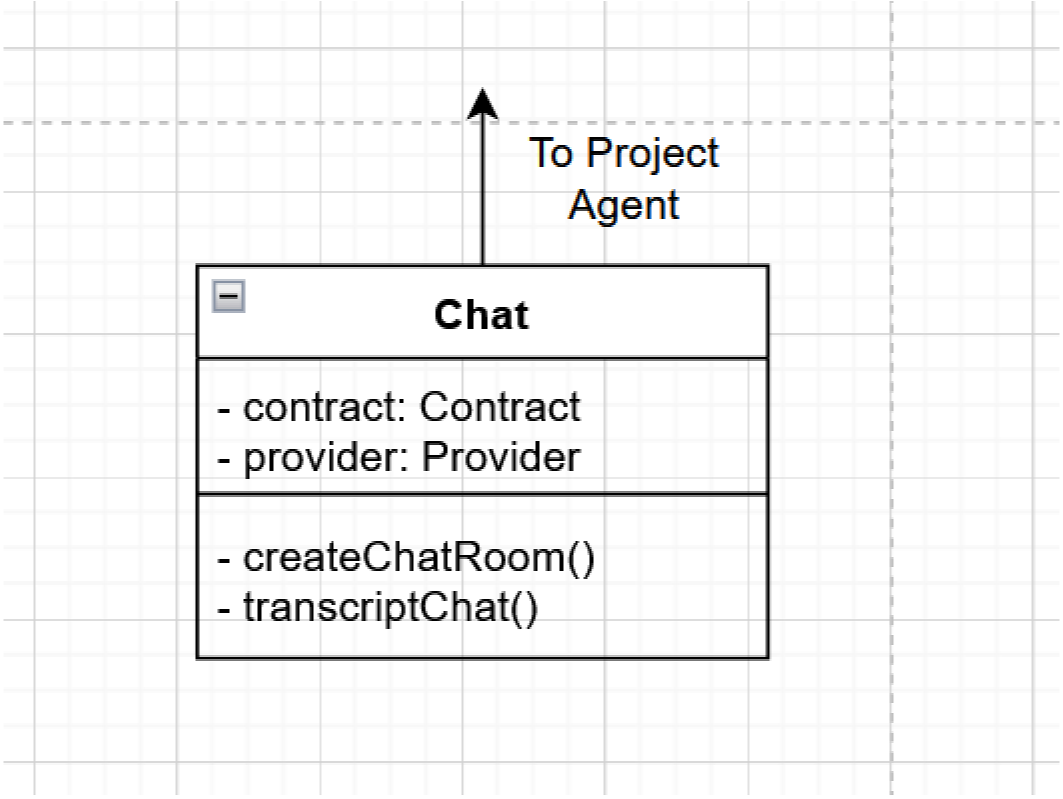
D2.2) Organizer Agent Architecture


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

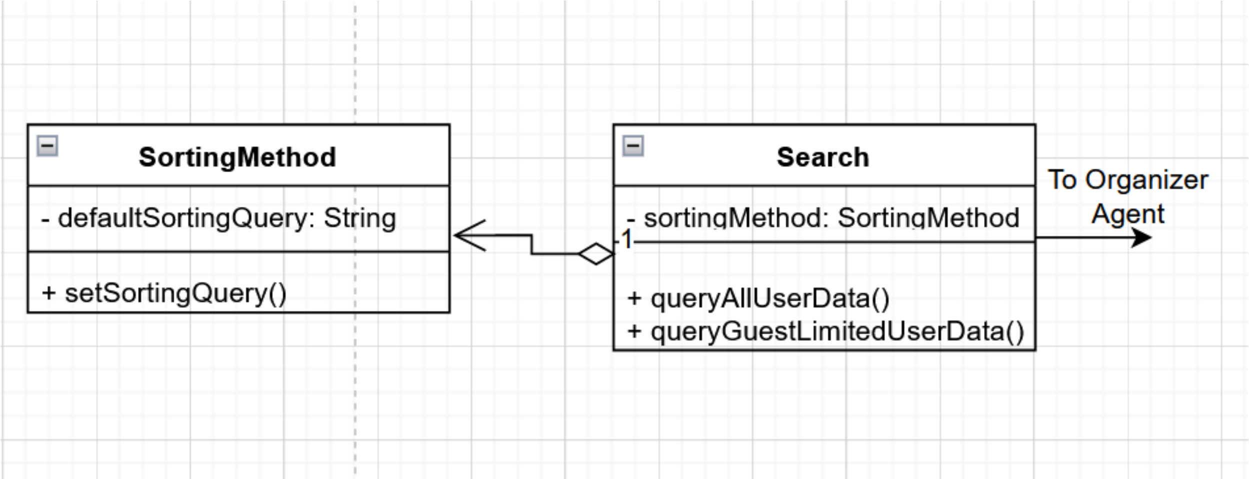
D2.3) Project Agent Architecture



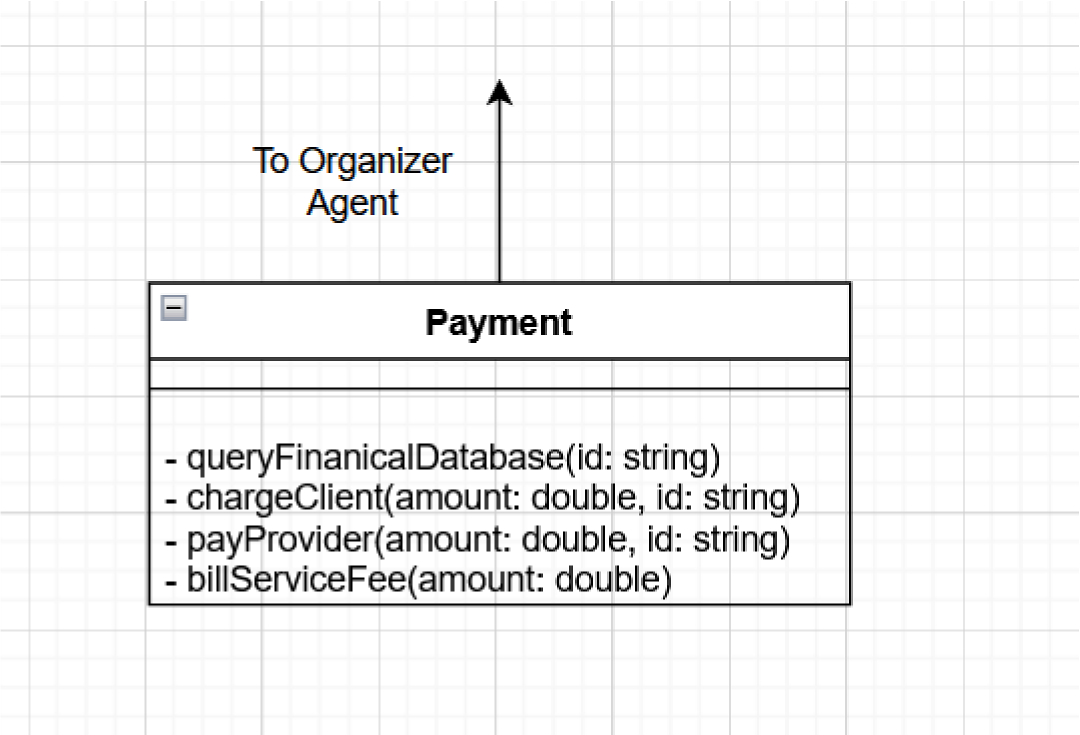
D2.4) Chat Agent Architecture



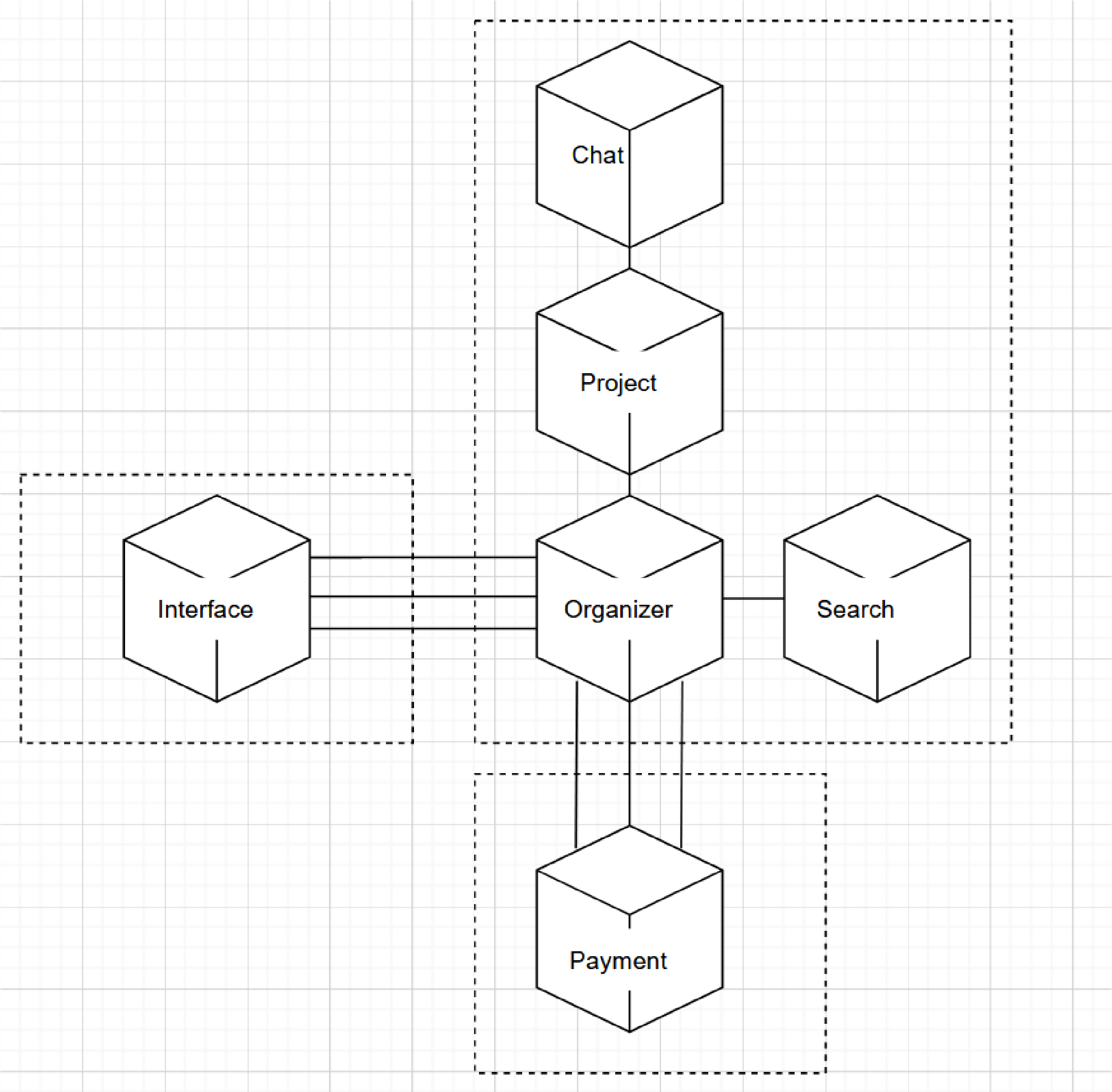
D2.5) Search Agent Architecture


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

D2.6) Payment Agent Architecture


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

D3) Agent System Design - Deployment Diagram


Chat
Project
Interface
Organizer
Search
Payment