Agent Based Software Engineering Project

Phase 1

Research Matchmaking

Group 2

Ehsan Mashhadi

Mohammad Amin Abedi

Ahmad Haji Mohammadkhani

Yashkumar Anantkumar Patel

Fall 2020

Dr. Behrouz Homayoun Far

1. Introduction

1.1 Description

There are many situations where people need to find each other to collaborate on specific projects or ideas, so there would be a platform that makes this process easier and efficient. The research matchmaking is a sample of this idea to help providers and clients work together to do a project by following some rules. The provided resources in this document are also available on https://github.com/aminabedi/Matchmaking-ABSE.

1.2 Specification

This system has three types of users namely provider, client, and guest. One needs to register to find or provide the services one is looking for. Clients can search for providers using different kinds of filters. On the other hand, the Provider needs to choose between two types of accounts either premium or basic. Once the client is able to find a suitable provider, the client can communicate via chat and make a deal with the provider. Both parties have the power to accept or reject the offers. After accepting the common term both need to sign an online contract and afterward both parties can communicate over the secure chat and can see the progress of the project.

Moreover, there are several things each member has the power to do. Like any change request by a client after accepting the contract can be accepted and rejected by the provider. Besides these there are several things that are decided by the agent like hourly compensation does not affect the ratings from the past clients, every payment should be gone through the payment agent and the system will receive 30% of the total payments.

The system will allow the contract negotiation process to be completed quickly and easily, with little need for user input (form either team owners or players) beyond the initial configuration of the software clients according to personal preferences.

1.3 Methodology

The idea behind this project is to implement the proposed topic by using agent-based methodologies in three phases. In the first part, which is a tutorial part, we are investigating the MAS requirements and design by using GAIA methodology which is a popular methodology in this area.

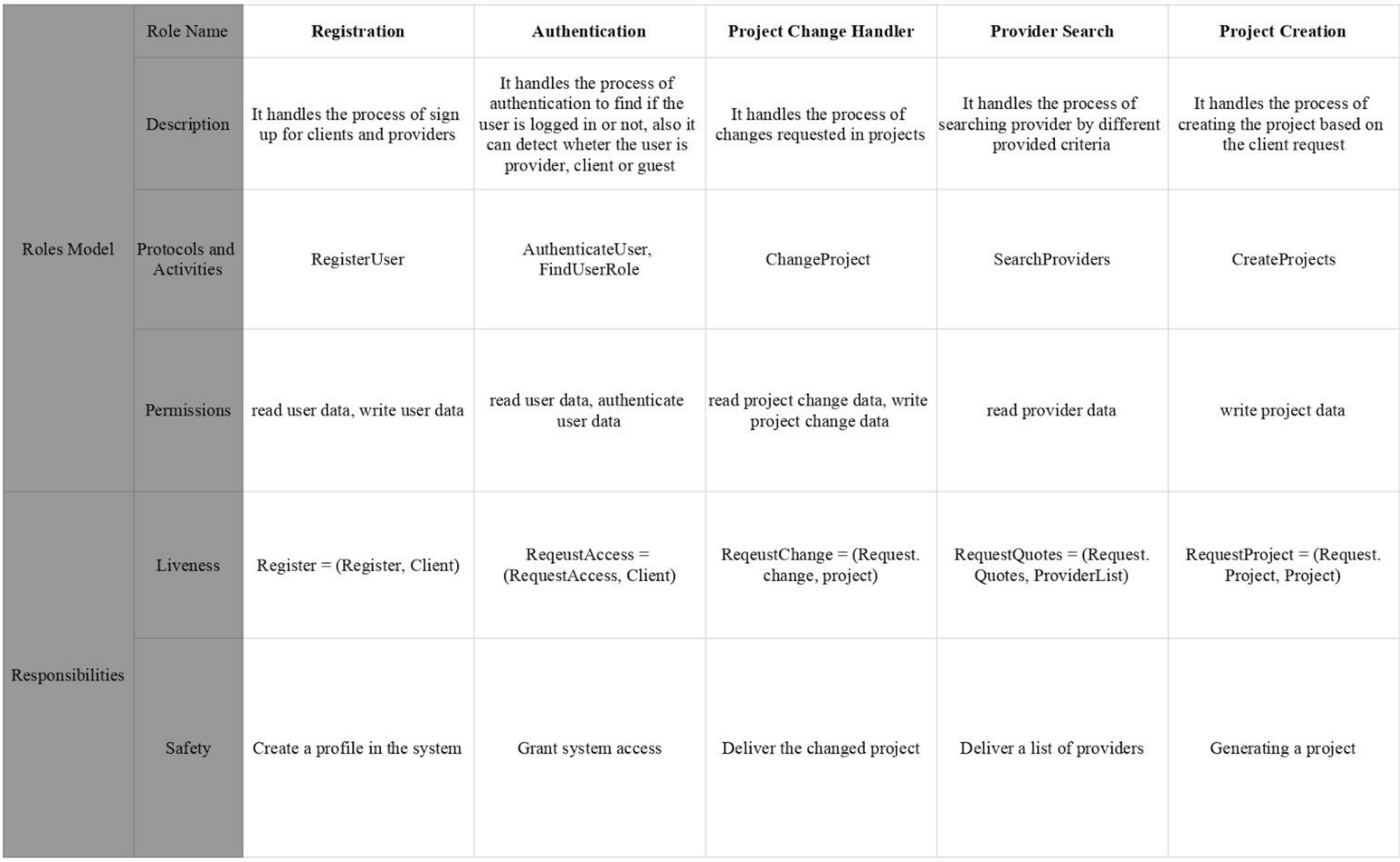
First, we will present the analysis part by defining Role Models by following the standard format of GAIA methodology. In the next part, we will document the Interaction Model and its related graph. After that, we will present the Design part by defining the Agent Model, Service Model, and Acquaintance model.

2. Analysis

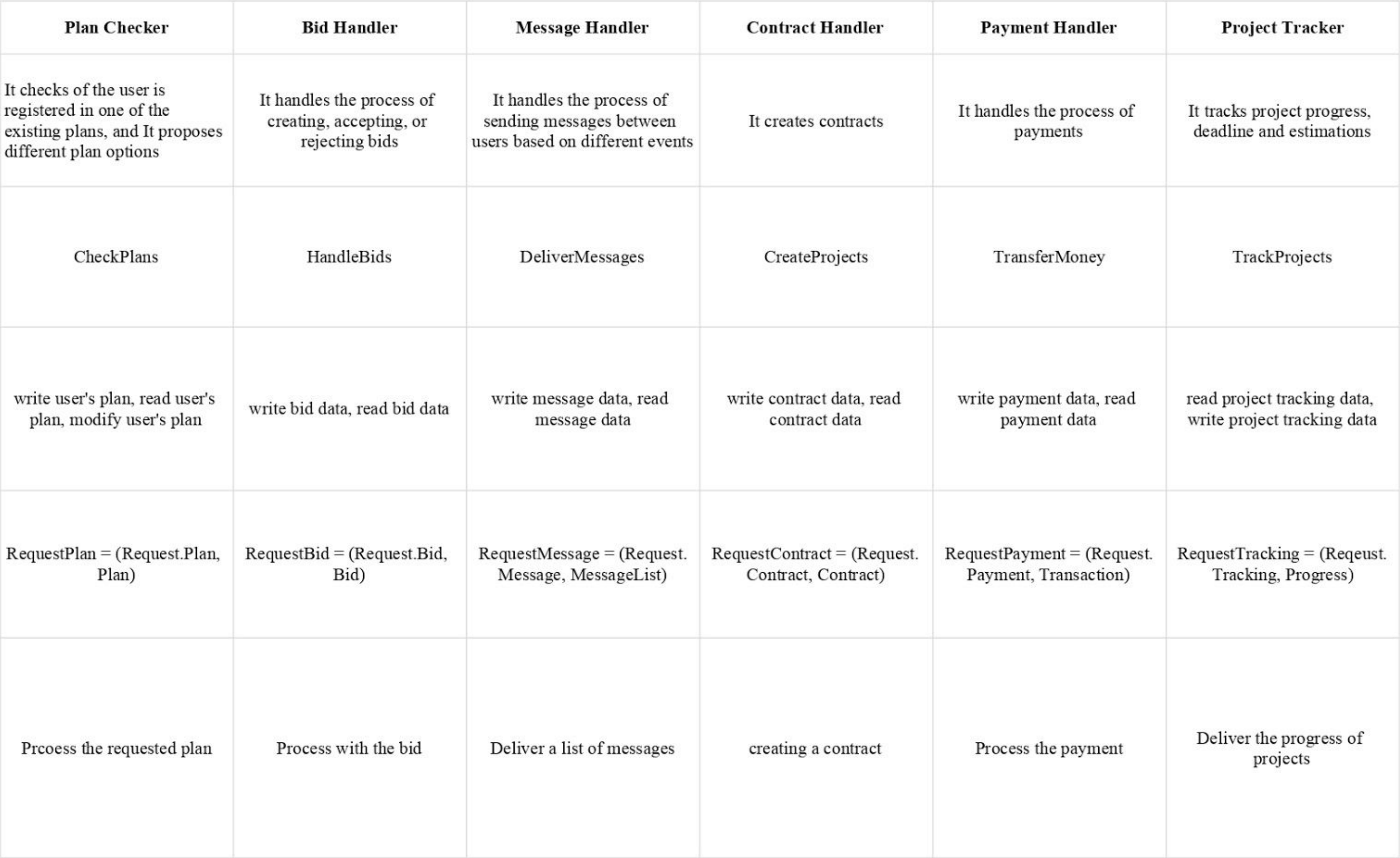
In this part, we will present our analysis of the project by following the GAIA methodologies to draw diagrams and also schemas. It is clear that this project can be analyzed in different ways, but we tried to make it as well as possible to make it suitable for implementation by different agents, and it may be changed in the next phases to make it more complete and accurate.

2.1 Roles Model

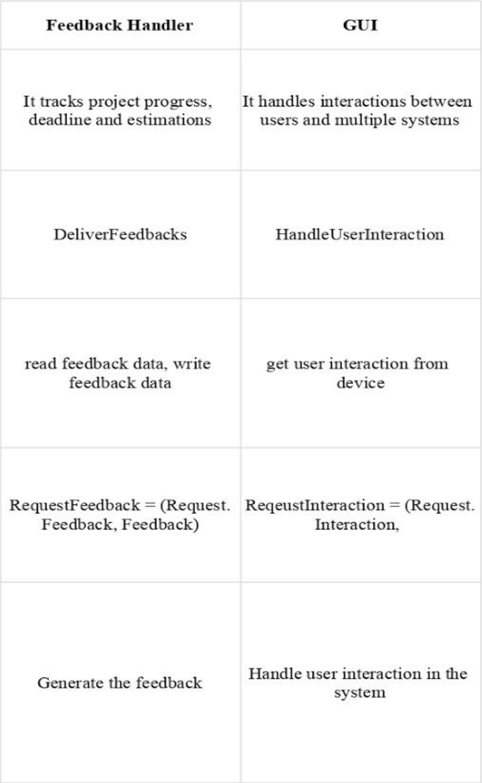
For finding the roles model which are the key roles of the system, we investigated the requirements provided in the project specification to find roles. Then we will provide a short description. After that, we will take a look at their permissions, responsibility, and also protocols and activities.



Roles Model	Role Name	**Registration**	**Authentication**	**Project Change Handler**	**Provider Search**	**Project Creation**
	Description	It handles the process of sign up for clients and providers	It handles the process of authentication to find if the user is logged in or not, also it can detect wheter the user is provider, client or guest	It handles the process of changes requested in projects	It handles the process of searching provider by different provided criteria	It handles the process of creating the project based on the client request
	Protocols and Activities	RegisterUser	AuthenticateUser, FindUserRole	ChangeProject	SearchProviders	CreateProjects
	Permissions	read user data, write user data	read user data, authenticate user data	read project change data, write project change data	read provider data	write project data
Responsibilities	Liveness	Register = (Register, Client)	ReqeustAccess = (RequestAccess, Client)	ReqeustChange = (Request. change, project)	RequestQuotes = (Request. Quotes, ProviderList)	RequestProject = (Request. Project, Project)
	Safety	Create a profile in the system	Grant system access	Deliver the changed project	Deliver a list of providers	Generating a project



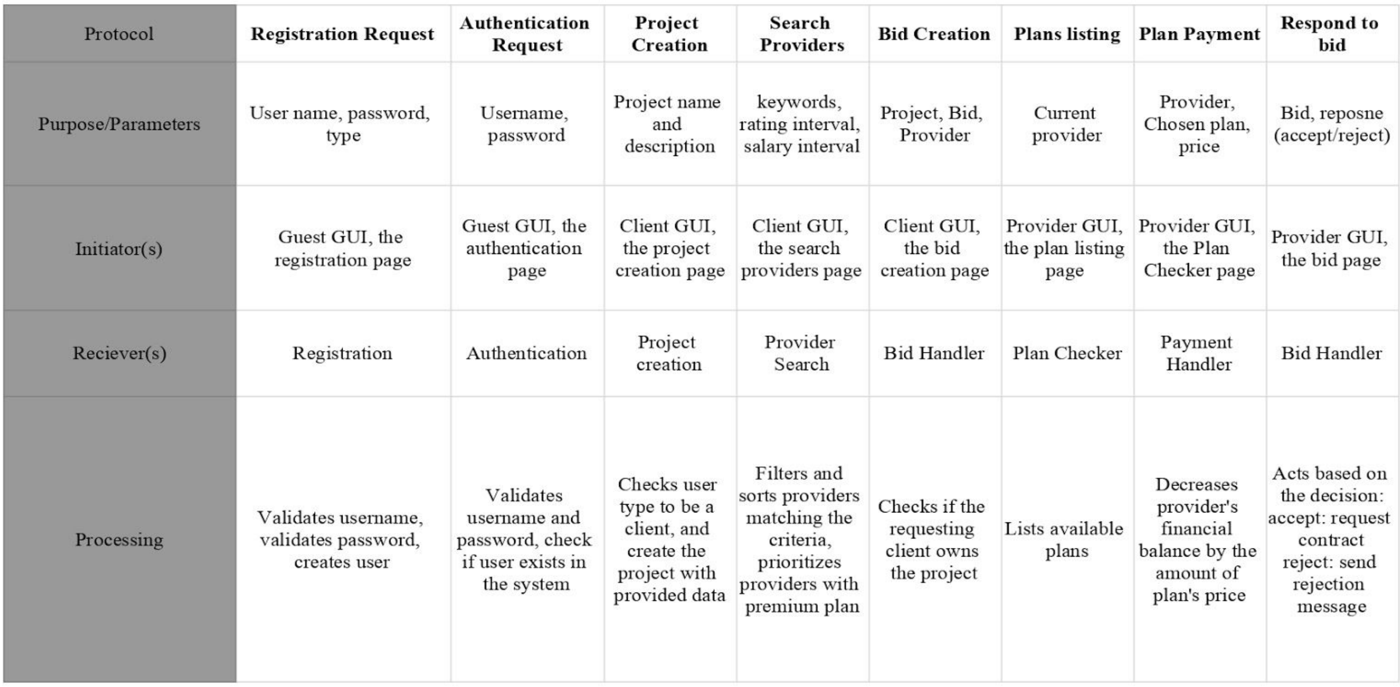
Plan Checker	Bid Handler	Message Handler	Contract Handler	Payment Handler	Project Tracker
It checks of the user is registered in one of the existing plans, and It proposes different plan options	It handles the process of creating, accepting, or rejecting bids	It handles the process of sending messages between users based on different events	It creates contracts	It handles the process of payments	It tracks project progress, deadline and estimations
CheckPlans	HandleBids	DeliverMessages	CreateProjects	TransferMoney	TrackProjects
write user's plan, read user's plan, modify user's plan	write bid data, read bid data	write message data, read message data	write contract data, read contract data	write payment data, read payment data	read project tracking data, write project tracking data
RequestPlan = (Request.Plan, Plan)	RequestBid = (Request.Bid, Bid)	RequestMessage = (Request. Message, MessageList)	RequestContract = (Request. Contract, Contract)	RequestPayment = (Request. Payment, Transaction)	RequestTracking = (Reqeust. Tracking, Progress)
Prcoess the requested plan	Process with the bid	Deliver a list of messages	creating a contract	Process the payment	Deliver the progress of projects



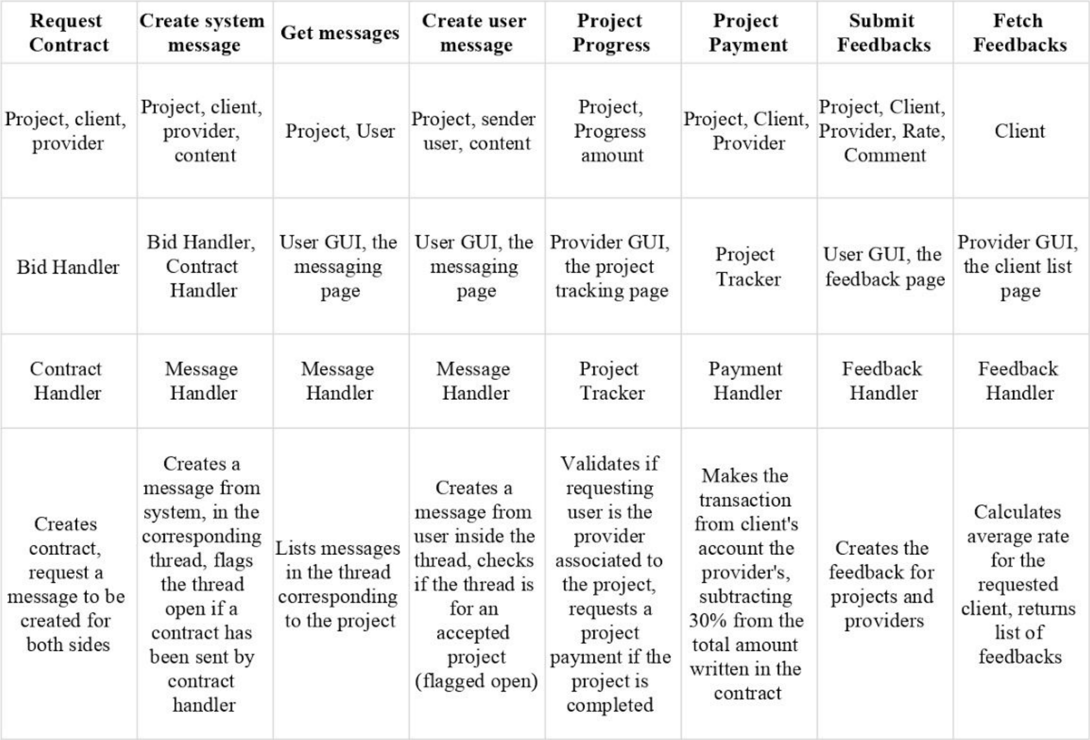
Feedback Handler	GUI
It tracks project progress, deadline and estimations	It handles interactions between users and multiple systems
DeliverFeedbacks	HandleUserInteraction
read feedback data, write feedback data	get user interaction from device
RequestFeedback = (Request. Feedback, Feedback)	ReqeustInteraction = (Request. Interaction,
Generate the feedback	Handle user interaction in the system

2.1 Interaction Model

In this part, we will present the interaction model which shows the interaction between different roles in the system. In the figure below we will show the purpose, initiator role, responder role, and processing.



Protocol	Registration Request	Authentication Request	Project Creation	Search Providers	Bid Creation	Plans listing	Plan Payment	Respond to bid
Purpose/Parameters	User name, password, type	Username, password	Project name and description	keywords, rating interval, salary interval	Project, Bid, Provider	Current provider	Provider, Chosen plan, price	Bid, reposne (accept/reject)
Initiator(s)	Guest GUI, the registration page	Guest GUI, the authentication page	Client GUI, the project creation page	Client GUI, the search providers page	Client GUI, the bid creation page	Provider GUI, the plan listing page	Provider GUI, the Plan Checker page	Provider GUI, the bid page
Reciever(s)	Registration	Authentication	Project creation	Provider Search	Bid Handler	Plan Checker	Payment Handler	Bid Handler
Processing	Validates username, validates password, creates user	Validates username and password, check if user exists in the system	Checks user type to be a client, and create the project with provided data	Filters and sorts providers matching the criteria, prioritizes providers with premium plan	Checks if the requesting client owns the project	Lists available plans	Decreases provider's financial balance by the amount of plan's price	Acts based on the decision: accept: request contract reject: send rejection message



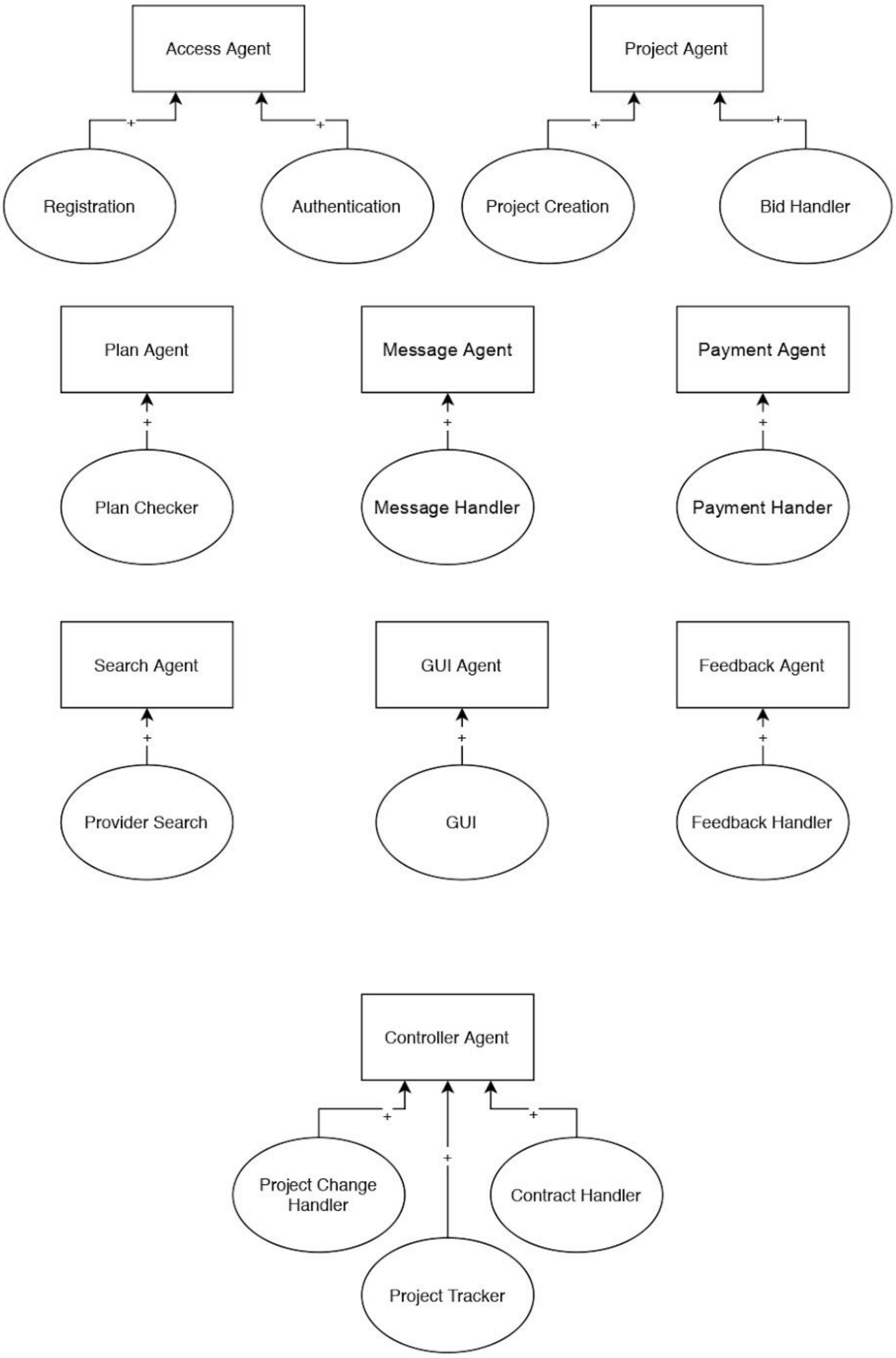
Request Contract	Create system message	Get messages	Create user message	Project Progress	Project Payment	Submit Feedbacks	Fetch Feedbacks
Project, client, provider	Project, client, provider, content	Project, User	Project, sender user, content	Project, Progress amount	Project, Client, Provider	Project, Client, Provider, Rate, Comment	Client
Bid Handler	Bid Handler, Contract Handler	User GUI, the messaging page	User GUI, the messaging page	Provider GUI, the project tracking page	Project Tracker	User GUI, the feedback page	Provider GUI, the client list page
Contract Handler	Message Handler	Message Handler	Message Handler	Project Tracker	Payment Handler	Feedback Handler	Feedback Handler
Creates contract, request a message to be created for both sides	Creates a message from system, in the corresponding thread, flags the thread open if a contract has been sent by contract handler	Lists messages in the thread corresponding to the project	Creates a message from user inside the thread, checks if the thread is for an accepted project (flagged open)	Validates if requesting user is the provider associated to the project, requests a project payment if the project is completed	Makes the transaction from client's account the provider's, subtracting 30% from the total amount written in the contract	Creates the feedback for projects and providers	Calculates average rate for the requested client, returns list of feedbacks

3. Design

In this part, we will present a detailed analysis of the project by following the GAIA methodologies to draw diagrams and also schemas.

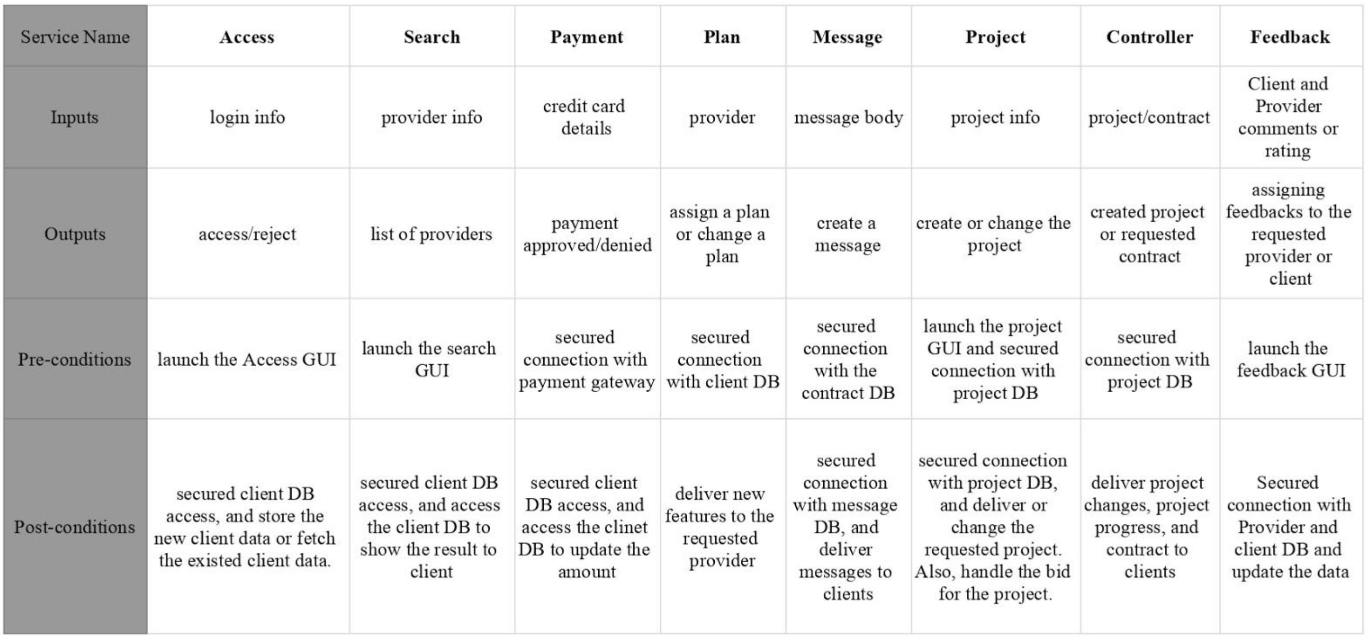
3.1 Agent Model

For finding the agents we take a look at roles and then group them if it is possible to make agents.



3.2 Service Model

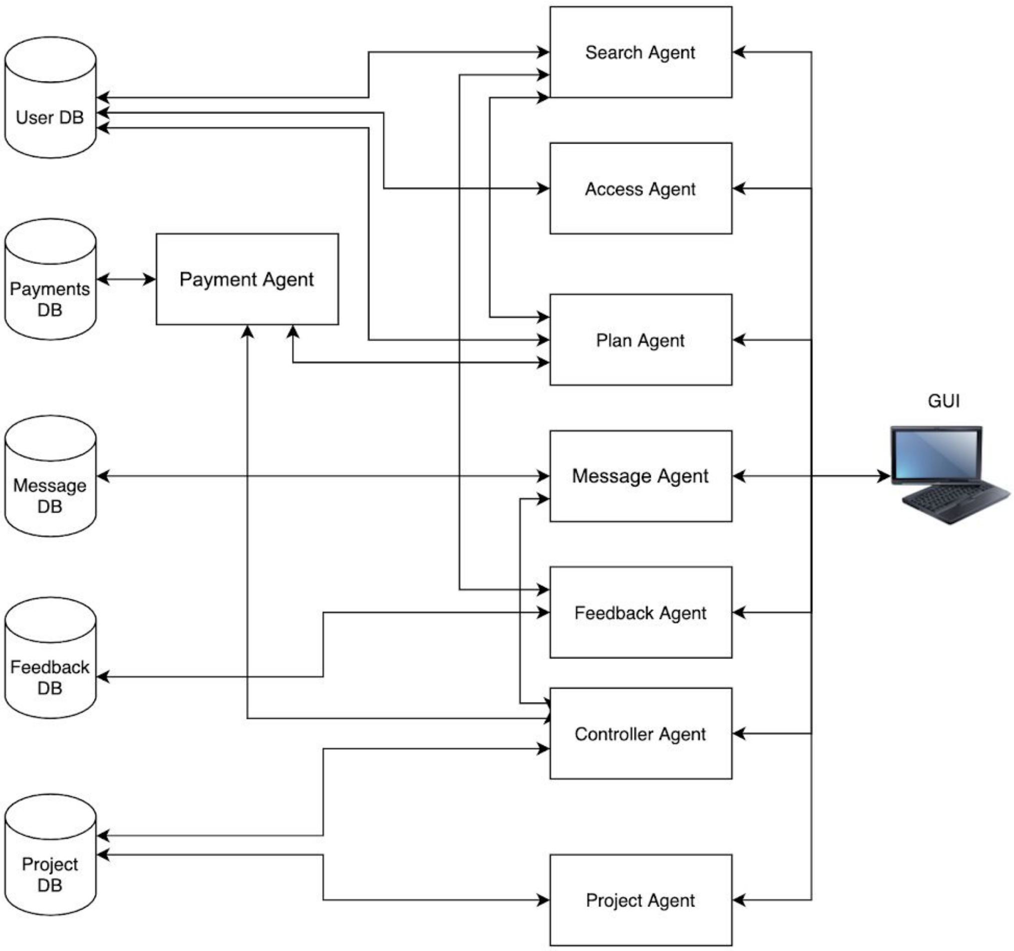
In this part, we will present a service model that shows the services assigned to each agent role. In the figure below, we will show inputs, outputs, pre-conditions, and post-conditions.



Service Name	Access	Search	Payment	Plan	Message	Project	Controller	Feedback
Inputs	login info	provider info	credit card details	provider	message body	project info	project/contract	Client and Provider comments or rating
Outputs	access/reject	list of providers	payment approved/denied	assign a plan or change a plan	create a message	create or change the project	created project or requested contract	assigning feedbacks to the requested provider or client
Pre-conditions	launch the Access GUI	launch the search GUI	secured connection with payment gateway	secured connection with client DB	secured connection with the contract DB	launch the project GUI and secured connection with project DB	secured connection with project DB	launch the feedback GUI
Post-conditions	secured client DB access, and store the new client data or fetch the existed client data.	secured client DB access, and access the client DB to show the result to client	secured client DB access, and access the clinet DB to update the amount	deliver new features to the requested provider	secured connection with message DB, and deliver messages to clients	secured connection with project DB, and deliver or change the requested project. Also, handle the bid for the project.	deliver project changes, project progress, and contract to clients	Secured connection with Provider and client DB and update the data

3.2 Acquaintance Model

For displaying the acquaintance model, we should define the communication links between agent types to identify potential bottlenecks to prevent problems later at runtime.


User DB
Payments DB
Message DB
Feedback DB
Project DB
Payment Agent
Search Agent
Access Agent
Plan Agent
Message Agent
Feedback Agent
Controller Agent
Project Agent
GUI