Design Document

1. Student

Name	Minh Nguyen Quang Le
Email	minh.le4@ucalgary.ca
UCID	30295926

2. System Name

Simple Travel Planner

3. Methodology: GAIA

In the document, GAIA is applied to design the Simple Travel Planner Multi-Agent System. The system consists of 5 interacting agents that collectively create the top 3 optimal travel plans for a user based on destination, budget, and travel dates.

3.1 The Analysis Phase

3.1.1 Goal identification

The goals below are used to clearly assign role responsibilities. Each of these goals corresponds to one or more roles within the system.

Description
Capture travel requirements from the user (destination, dates, budget)
Obtain available transportation options (bus, train, flight)
Obtain available hotel accommodations
Combine and evaluate options to produce the top 3 travel plans within the budget
Simulate payment and confirm booking
Return finalized plan to the user

3.1.2 GAIA Roles Model

The analysis phase led to the identification of five roles: Requester, Planner, TransportProvider, HotelProvider and Payment

Role Schema		Requester
Description		represents the user who handles travel preferences (destination, budget, travel dates) Communicates with the MAS to request and confirm plans
Protocols and Activities		ProvideRequirements, ReceiveProposals, SendStatus (ConfirmPlan, RejectPlan), ReceiveConfirmation
Permissions		read user input, read proposal plans, read confirmation status generate approval decisions
Responsibilities	Liveness	REQUESTER = (ProvideTravelPlan) PROVIDETRAVELPLAN = ProvideRequirements.ReceiveProposals.SendStatus(ConfirmPlan | RejectPlan).ReceiveConfirmation
	Safety	- a successful connection with MAS - a successful providing valid input to Planner

Role Schema	Planner

Description		acts as the central coordinator. Receives user requirements, queries to transportation and hotel providers, collects options, generates the top 3 travel plans, and interacts with Payment for finalization
Protocols and Activities		ReceiveRequirements, QueryProviders (QueryTransport, QueryHotel), ReceiveOptions, GeneratePlan, SendPlansToRequester, ReceiveConfirmationFromRequester, TriggerPayment, ReceivePaymentConfirmationFromPayement, SendPaymentConfirmationToRequester
Permissions		read provider data, read user request and status write plan proposals, write user request and status
Responsibilities	Liveness	PLANNER = (PlanTrip) PLANTRIP = ReceiveRequirements.QueryStage.PlanningStage.PaymentStage QUERYSTAGE = QueryProviders.(QueryTransport || QueryHotel).ReceiveOptions PLANNINGSTAGE = GeneratePlan.SendPlansToRequester.ReceiveConfirmationFromRequester PAYMENTSTAGE = TriggerPayment.ReceivePaymentConfirmationFromPayment.SendPaymentConfirmationToRequester
	Safety	- a successful conncection with requester, providers and payment - a successful plan must combine at least one transport and one hotel option with total_cost ≤ budget - handle cases with no options by notifying rejection - a successful payment occurs only after the user confirms a valid travel plan

Role Schema		TransportProvider
Description		supplies available travel options (flight, train, bus) based on planner query
Protocols and Activities		ReceiveTransportQuery, SearchTransportDatabase, SendTransportOptions
Permissions		read transport database
Responsibilities	Liveness	TRANSPORTPROVIDER = (ProvideTransportOptions) PROVIDERTRANSPORTOPTIONS = ReceiveTransportQuery.SearchTransportDatabase.SendTransportOptions
	Safety	- a successful connection with the Planner - all provided options must match the requested dates and destination

Role Schema	HotelProvider

Description		supplies available hotel options based on planner query
Protocols and Activities		ReceiveHotelQuery, SearchHotelDatabase, SendHotelOptions
Permissions		read hotel database
Responsibilities	Liveness	HOTELPROVIDER = (ProvideHotelOptions) PROVIDERHOTELOPTIONS = ReceiveHotelQuery.SearchHotelDatabase.SendHotelOptions
	Safety	- a successful connection with the Planner - must return options that are available for the requested dates and destination

Role Schema		Payment
Description		handles the final booking process, simulating payment and generating confirmations upon approval
Protocols and Activities		ReceiveBookingTrigger, ProcessPayment, SendPaymentConfirmation
Permissions		Writes confirmation data Read payment info
Responsibilities	Liveness	PAYMENT = (PaymentProcess) PAYMENTPROCESS = ReceiveBookingTrigger.ProcessPayment.SendPaymentConfirmation
	Safety	- a successful connection with the Planner - simulate success only if plan is valid

3.1.3 Interactions Model

Interaction model denotes which action returns from a request along with the roles that can initiate a request and the corresponding responders

Protocol	**RequestService. Requester**	**RequestService.Planner**	**Response.TransportProvider**	**Response.HotelProvider**
Purpose / Parameters	User requests travel planning service	Planner requests transport and hotel availability	Provide available transport options	Provide available hotel options
Initiator	Requester	Planner	TransportProvider	HotelProvider
Receiver	Planner Agent	TransportProvider / HotelProvider	Planner	Planner

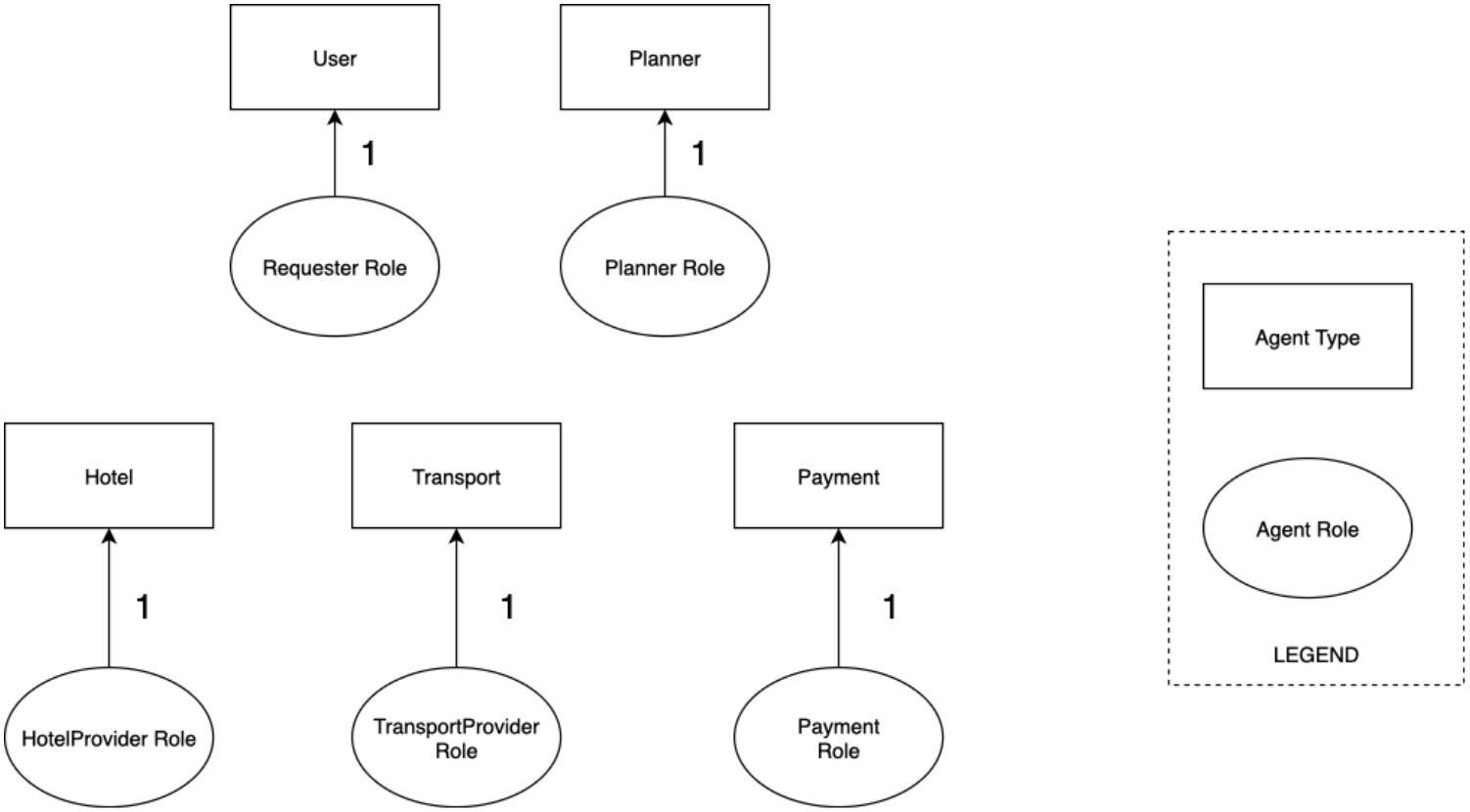
Inputs	Destination, dates, budget, preferences	Destination, dates, budget	Request parameters	Request parameters
Outputs	Acknowledgment	Request message	List of transport options	List of hotel options
Processing	Planner receives requirements and starts service planning	Planner queries providers based on user needs	TransportProvider provides transport options and Planner receives these options	HotelProvider provides hotel options and Planner receives these options

Protocol	**Submit.PlanToUser**	**Submit.UserConfirmation**	**Submit.Payment**	**Response.Payment**
Purpose / Parameters	Present travel plan to user for confirmation	User confirms or rejects the proposed travel plan	Proceed with the order and payment	Payment responds with transaction confirmation
Initiator	Planner	Requester	Planner	Payment
Receiver	Requester	Planner	Payment	Planner
Inputs	Combined transport + hotel options, total cost	User decision	Payment details, total cost	Payment request
Outputs	User view of plan	Confirmation status	Payment request	Payment confirmation
Processing	Provide plans and user reviews the proposed plans	Planner processes user decision and proceeds if accepted	Payment Agent initiates payment transaction	Planner finalizes the booking and notifies the user

3.2 Design Phase

During this phase the Agent model is achieved, along with the services and acquaintance models.

3.2.1 Agent Model



Agent Type	Role Played	Cardinality	Description
User Agent	Requester Role	1:1	A single instance acts as the user interface agent, handling input and decision
Planner Agent	Planner Role	1:1	A single instance serves as the central coordinator for planning
Transport Agent	TransportProvider Role	1:1	A single instance provides transport options
Hotel Agent	HotelProvider Role	1:1	A single instance provides hotel options
Payment Agent	Payment Role	1:1	A single instance handles payment

3.2.2 Service Model

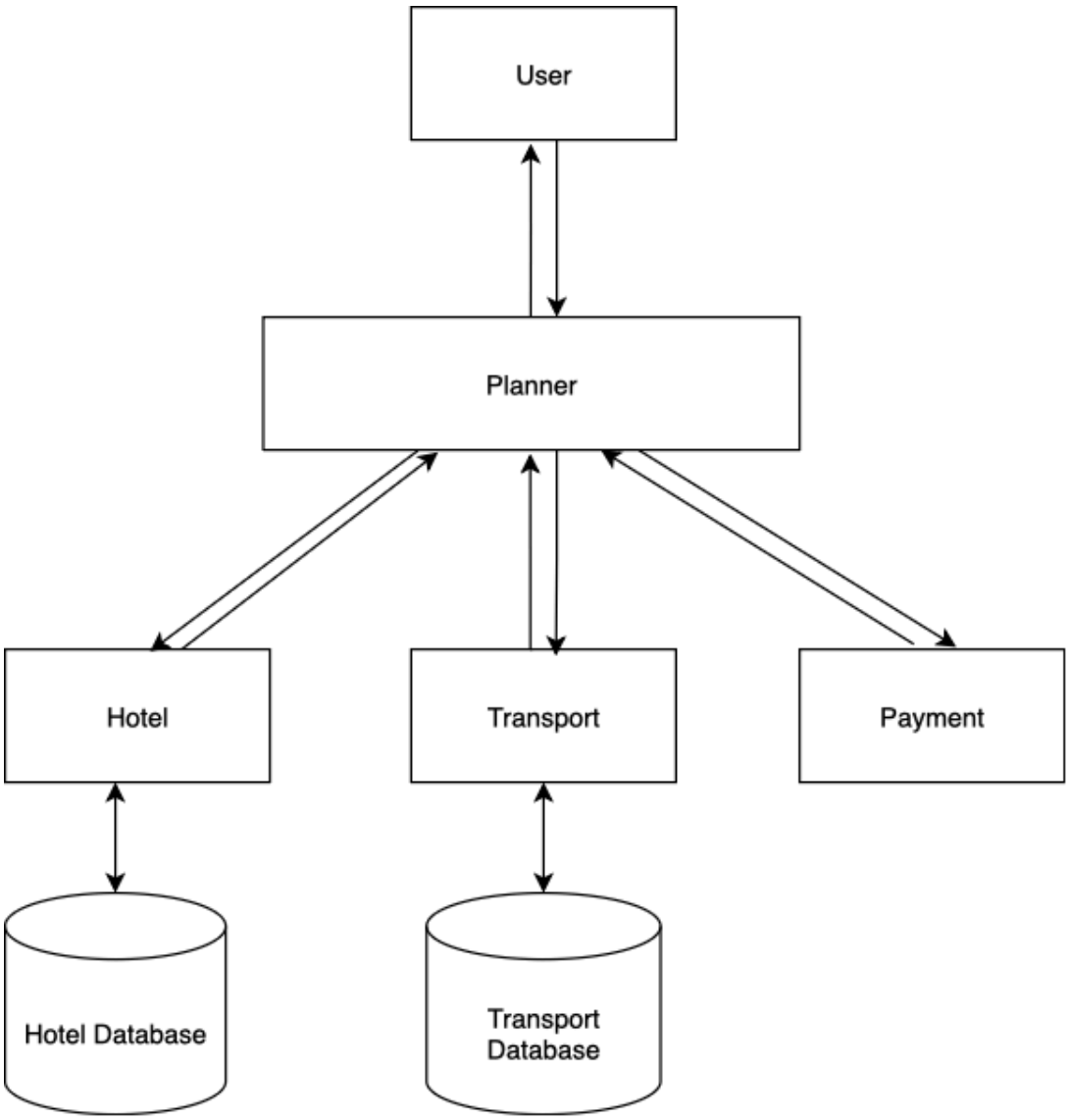
The following table detail the services for each agent, grouped by agent type as defined in the Agent Model

Agent	Service Name	Inputs	Outputs	Pre-conditions	Post-conditions
User	ProvideRequirements	requirement info (destination, dates, budget)	travel requirements message sent to Planner agent	user input available and valid	requirements transmitted and planning process triggered.
Planner	CoordinatePlanning	travel requirements from User agent	plan proposal message to User agent	requirements valid and providers available	plan generated or user notified of no viable options
Planner	EvaluateOptions	transport options and hotel options from providers	selected plan (internal data structure)	all options received	best plans identified and ready for proposal
Planner	TriggerBooking	approval (accept) from User agent	booking trigger message to Payment agent.	plan approved	Execution started; confirmation expected.
Transport	SupplyTransportOptions	transport query from Planner agent (dates, destination)	list of transport options	query parametes valid	options provided (empty list if none match)
Hotel	SupplyHotelOptions	hotel query from Planner agent (dates, destination)	list of hotel options	query parametes valid	options provided (empty list if none match)

Payment	ConfirmBooking	booking trigger from Planner agent (plan details)	confirmation message to Planner agent	plan details complete and valid	booking confirmed

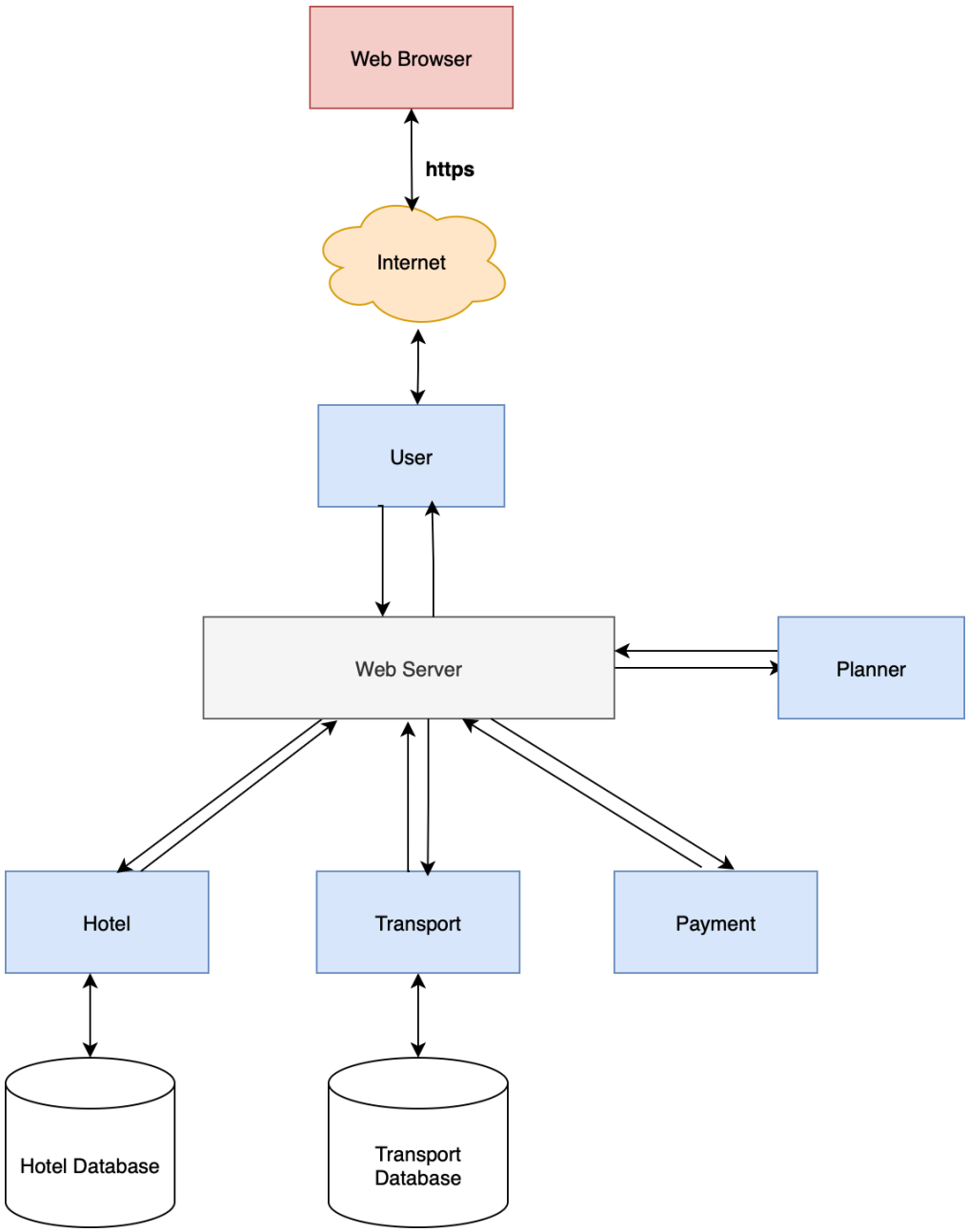
3.3.3 Acquaintances Model

I define the communication links that exist between 5 agent types



3.3.4 Overal system

The MAS communicates with the web browser over the internet using the HTTPS protocol. The web browser allows users to enter travel details, which are sent to the web server. The web server forwards these requests to the JADE platform, where agents such as the User, Planner, Transport, Hotel, and Payment Agents coordinate to process data and generate optimized travel plans. The final results are returned through the web server to the web browser for user display.



3.3.5 Communication (Internal): Message sequence chart

The User Agent initiates the communication by requesting a travel plan, providing input parameters such as destination, travel dates, and budget. Upon receiving this request, the Planner Agent sends separate queries to the Transport Agent and the Hotel Agent to obtain data from the Transportation and Hotel databases.

The Transport Agent and Hotel Agent each retrieve the relevant data from their respective databases and return the results to the Planner Agent. After receiving both responses, the Planner Agent processes and evaluates the data to generate the top three optimized travel plans based on the user's preferences and constraints. These plans are then sent to the User Agent for review.

Once the User Agent confirms the selected plan, the Planner Agent triggers a signal to the Payment Agent to initiate the payment process. After the transaction is successfully completed, the Payment Agent sends a confirmation message to the Planner Agent, which in turn forwards the confirmation to the User Agent to finalize the process.