GROUP# 11
Hetalben Virani (30183515)
Jahnavi Akuri (30180821)
Ayesha Singh (30194192)

SENG 696: Agent-based Software Engineering – Fall 2022

Online Abroad Consultancy

Project Overview

Steps towards Abroad studies:

i) Student registration.
ii) Data provided by the student is stored in the database.
iii) Successful student login after authentication.
iv) A query is made to list the universities based on student program.
v) Student can apply, bookmark, or compare the universities.
vi) Student can view the bookmarked universities in their profile.
vii) A query is made to compare the universities based on different criteria like ranking of the university based on course, fee structure, Alumni feedback etc.,
viii) Student can save and view the list of universities in their profile history.

Agents Outline:

i) Interface Agent – Connect to backend server.
ii) Authorization Agent – Authorizes the user data.
iii) Classifier Agent – Request to Search API to fetch data from DB.
iv) Compare Agent – Request to Compare API to compare data.
v) Database Agent – Store and retrieves the data from the database

Technology Stack:

i) Frontend: Angular 6.0
ii) Backend: Node JS
iii) Database: MySQL || Local DB: User data Storage
iv) JADE-Agent Development Environment
v) Gaia- Agent Methodology

System Requirements:

i) Frontend app will have a user-friendly GUI.

ii) Student would register into database.

iii) Student would be authenticated based on the data provided using HTTP API requests.

iv) Backend server must be capable of receiving, handling, and responding to API request from the app.

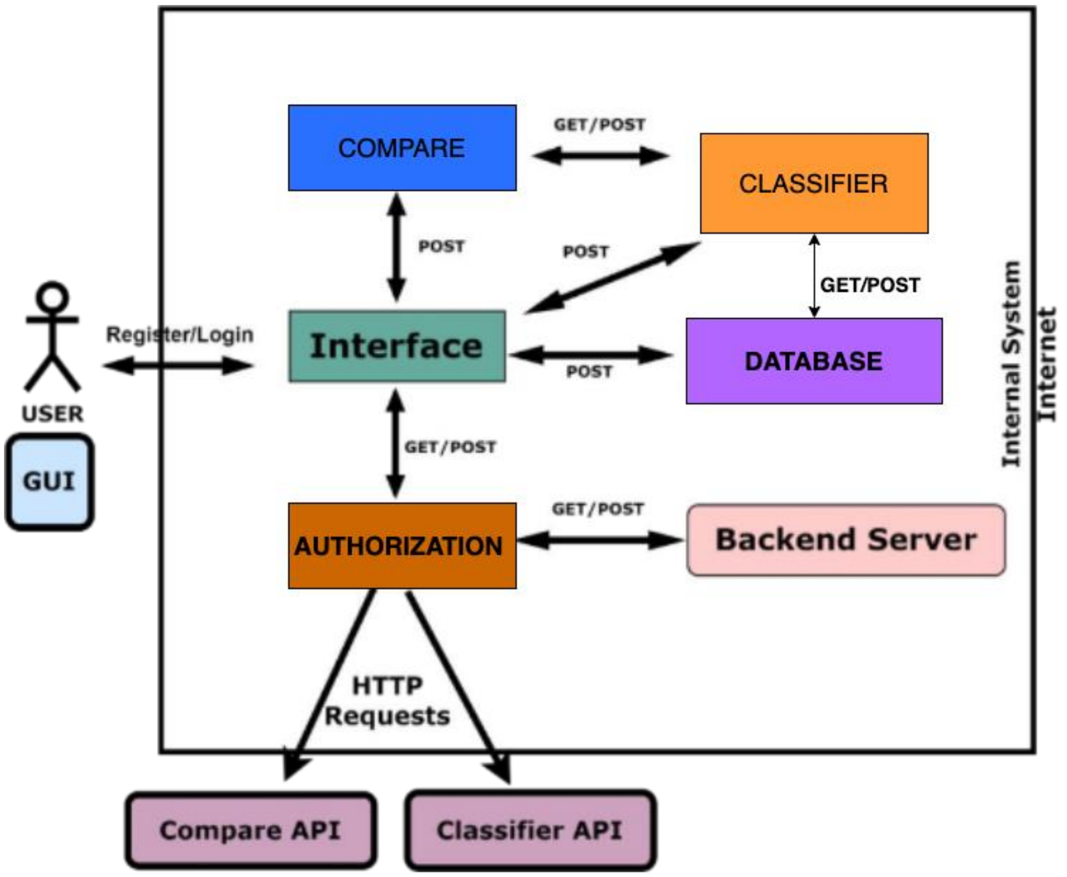
v) Upon receiving request, server will hand off processing to agents.

vi) System can store the user's information in Database (MySQL).

vii) System agents can expect inputs from users and display the result of list of universities by using search functionalities.

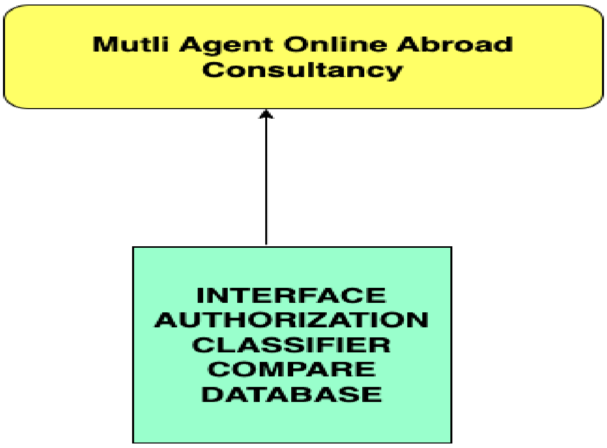
viii) Registered users can compare tuition fees, Duration of course of the universities.

ix) Users can save multiple universities and it would be displayed under the bookmarks.

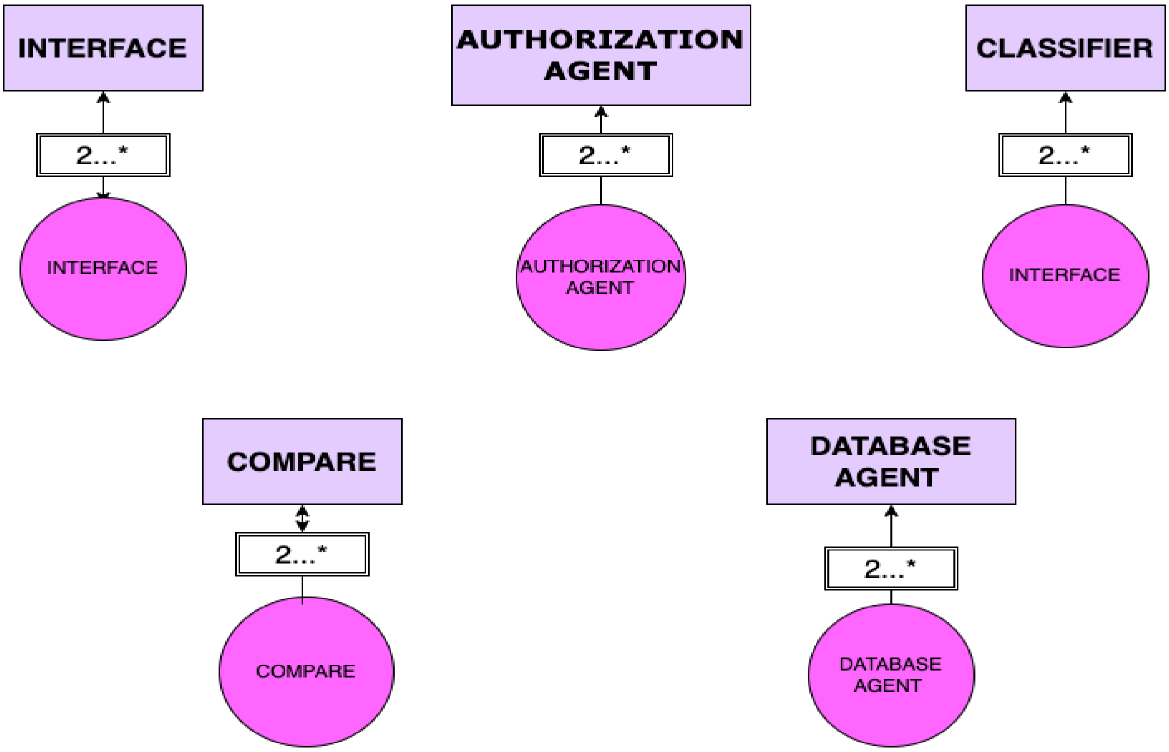


Analysis:

Role Model:



Agents Model:



Role Schemas:

Role Schema	**Interface**
Description	To provide the user data and to receive the data after processing
Protocols and Activities	– Transfer user's data – Receive list of universities
Permissions	Read and write from Backend Server
Responsibilities	-Liveness: RequestService = (RequestService.SERVICE) - Safety:

Role Schema	**Authorization**
Description	Sends the request to Database for the authorization of user data
Protocols and Activities	Sends the Request to DB for user data authorization as a POST method and retrieves the record from database as a GET method.
Permissions	HTTP Internet Access. Connect to Database agent
Responsibilities	-Liveness: Authorization Service = (GatewayService.SERVICE) -Safety: Secure connection with search and compare API

Role Schema	**Classifier**
Description	To receive the user's input via GUI and search for the list of universities based on user's data
Protocols and Activities	Request list of universities from Search API, as a POST method.
Permissions	Connects to Interface and Compare Agents
Responsibilities	-Liveness: DataClassifierService = (DataClassifierService.SERVICE) - Safety: Secure connection with Search API.

Role Schema	**Compare**
Description	To receive the list of universities text from classifier, request data from Compare API and send results back to Interface agent
Protocols and Activities	Query the Compare API, as a GET method.
Permissions	Connects to Classifier and Interface Agents
Responsibilities	-Liveness: CompareDataService = (CompareDataService.SERVICE) -Safety: Secure connection with Compare API.

Role Schema	**Database Agent**
Description	To receive input from user and save the data into the local database for ease of future reference
Protocols and Activities	Retrieves the records from DB based on user input
Permissions	Connect to Classifier, Authorization and Interface Agents
Responsibilities	-Liveness: SaveDataService = (SaveDataService.SERVICE) FetchDataService = (FetchDataService.SERVICE) -Safety: Secure connection with Classifier API.

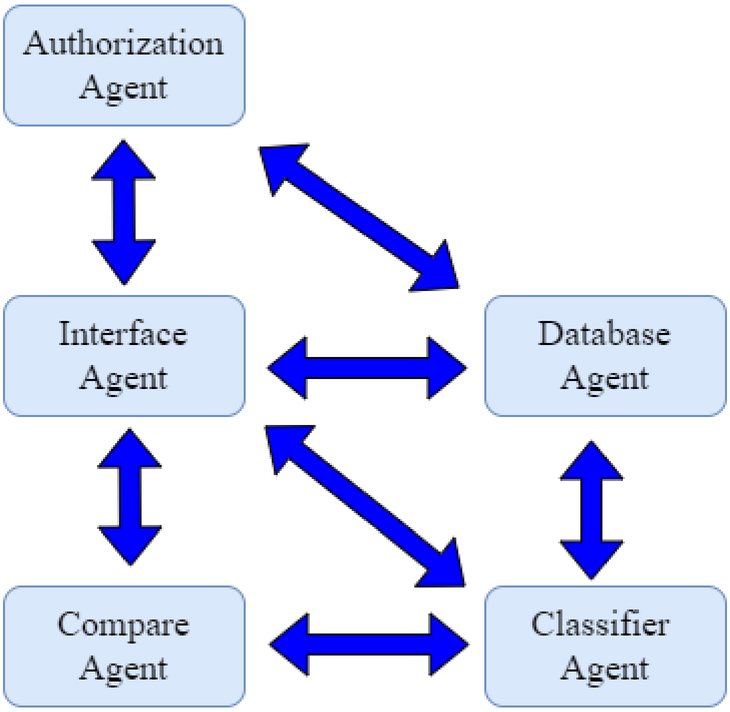
Interaction Model:

Protocol	**Interface**	**Authorization**	**Classifier**	**Compare**	**Database**
Purpose/ Parameters	Provides a registered user detail. Sends and receives request to store and fetch data.	Acts as a Verification module in order to authorize the user login data	Receives user input, request data from databaseto and sends to the User based on search classification	Receives the Universities from Classifier, requests data from Search API, and sends Comparison back to interface	Stores and retrieves the data based on user input
Initiator(s)	User and backend server	Interface request	Search	Classifier	Interface Query
Receiver(s)	Authorization	Classifier	Interface	Interface	Interface
Processing	Query the database based on user input	Request to fetch data from database	Returns JSONdata from API calls to agents that requested it	Returns the result of the Comparison request back to the Interface Agent.	Save and retrieves the records based on the quries

Services Model:

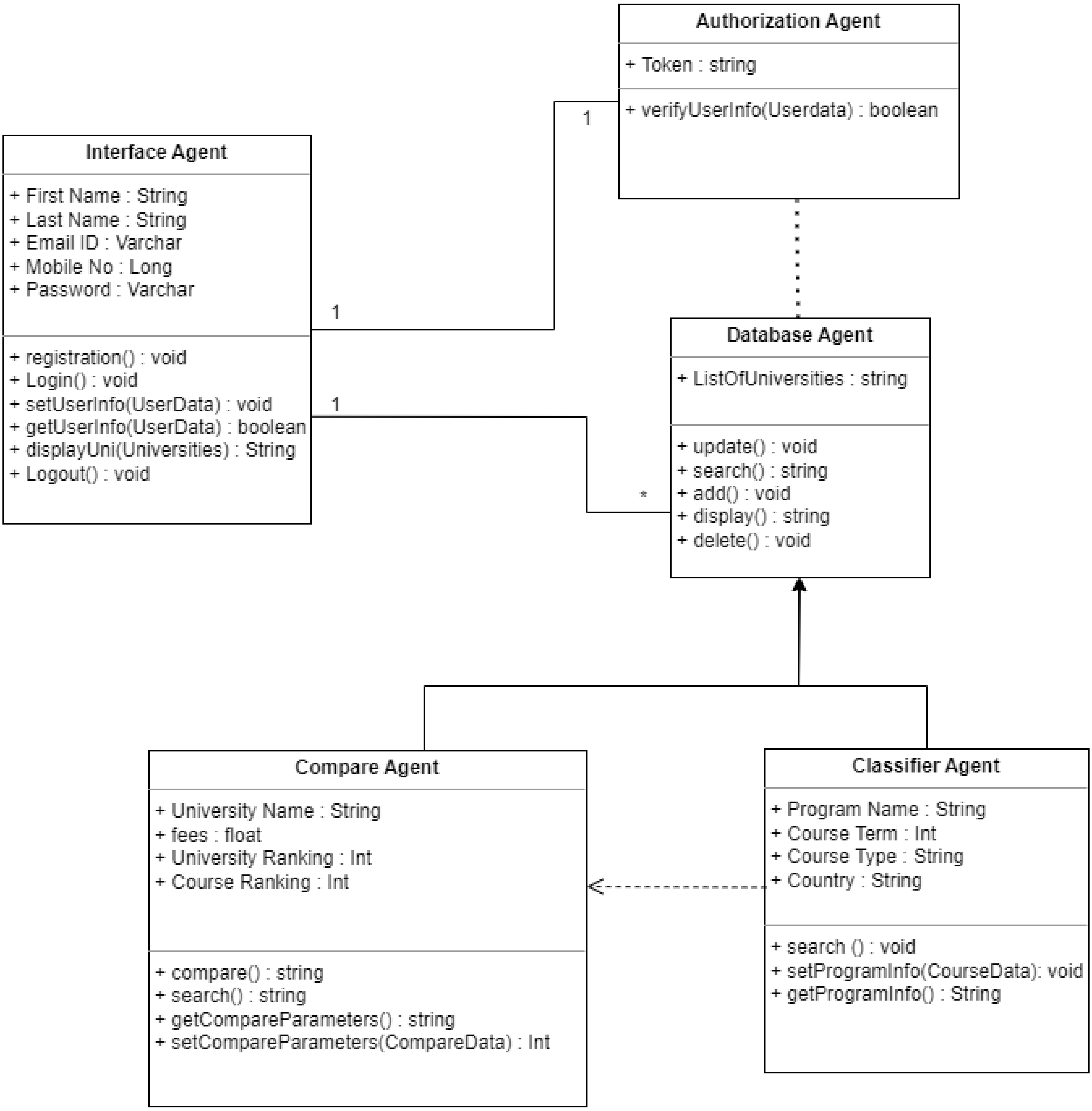
Services	Inputs	Output	Preconditions	Postconditions
Interface	User Data	List of University based on user input.	Users send data through GUI.	We can review the complete list of universities.
Authorization	Handles HTTP requests.	Processing HTTP and sends response.	Fetch the HTTP'S request and sends it to the interface agent.	Takes the output and sends it to the interface.
Classifier	Receives data from database based on user input.	Returns JSON data.	Sends the search query to DB after authorization of user's data	Retrieves records from DB and send it to the interface
Compare	Receives input from classifier for comparison.	List of Universities after comparison.	Compares the data fetched from database based on user input.	Returns the comparison list to the user.
Database	Receives input from interface	Retrieves the records based on user queries	Authorizes the user data.	Saves the data into the database for future reference based on user request

Acquaintances Model:

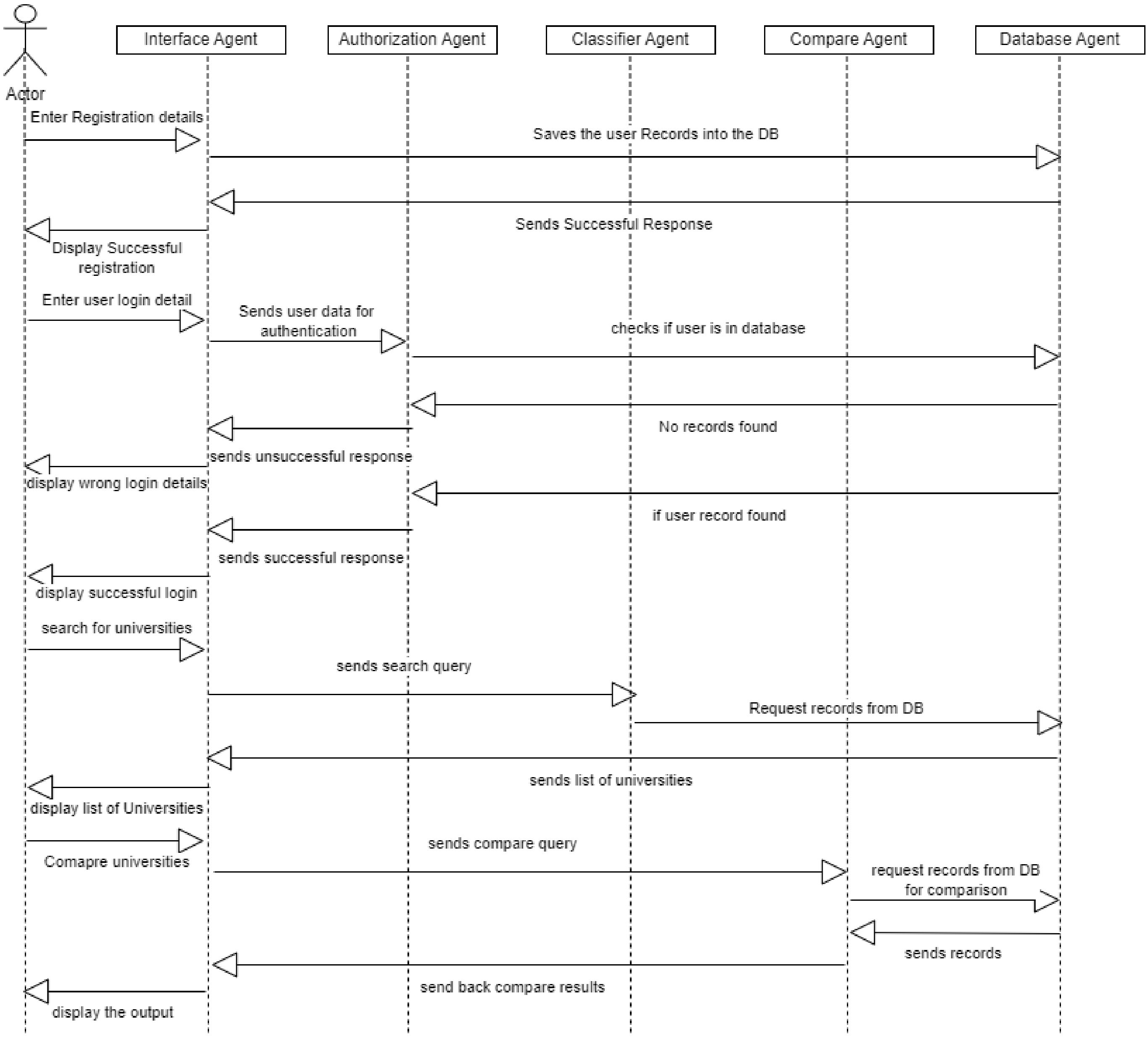


Part 2: Detailed Development Document

1. Class Diagram



2. Message Sequence Chart(i.e.interactions and protocols between the agent


Actor
Interface Agent
Authorization Agent
Classifier Agent
Compare Agent
Database Agent
Enter Registration details
Saves the user Records into the DB
Sends Successful Response
Display Successful registration
Enter user login detail
Sends user data for authentication
checks if user is in database
No records found
sends unsuccessful response
display wrong login details
if user record found
sends successful response
display successful login
search for universities
sends search query
Request records from DB
sends list of universities
display list of Universities
Comapre universities
sends compare query
request records from DB for comparison
sends records
send back compare results
display the output

3. Use Case Definition (for all agents):

- **Interface Agent:**

Brief Description	handles the interface and GUI of the overall system
Precondition	GUI has been loaded
Postcondition	Results are showing
Process Steps:	
1	Sends the User Data to the All Agent
2	Receives Data from Database Agent and sends it to GUI

- **Authorization Agent:**

Brief Description	make HTTP API requests outside of the Agent network
Precondition	User enter a string and can make API calls
Postcondition	Send back Authentication, List of Universities to agents
Process Steps:	
1	Listen for message from Interface Agent
2	Validate data from Database and reply to Interface
3	Process data from database (via API call)
4	Reply to Interface with Authenticated Data

- **Classifier Agent:**

Brief Description	handle data from Database and forward to Interface
Precondition	Sends the User input to the Database agent.
Postcondition	Returns the Jason and sends it to the List of data based on user request.
Process Steps:	
1	Take input from user through Interface Agent.
2	Send request for fetching data from database
3	Wait for an Authentication from Database.
4	Send Interface Agent the List of Universities.

- **Compare Agent:**

Brief Description	Responsible for comparing from the database
Precondition	Sends the User input to the Classifier agent.
Postcondition	Returns the Jason and sends it to the comparison based on user request.
Process Steps:	
1	Get the data from the User.
2	Reads the data to check whether users' data exist or not.
3	Display the List of Universities to the Classifier Agent.

- **Database Agent:**

Brief Description	Responsible for Save data in database.
Precondition	Sends the User input to the Interface, Classifier, Compare, Authorization Agent
Postcondition	Returns the Jason and sends it to the Authenticated data based onuser request.
Process Steps:	
1	Call the API to Fetch the data from the database
2	Display the List of Universities based on different Agent request.
3	Redirected back to the agent after Authentication.