Slide 1: Title: "Search for a Complete Vacation Package using Multi-Agent Negotiation."
Hello everyone! I'm Mihailov Emilian, and today I'm excited to present my project, which focuses on providing everyday users with a complete vacation package search solution. To accomplish this, I've used a multi-agent negotiation approach as the project's processing core. It's important to mention from the beginning that in this scenario the keyword "negotiation" is not about "bargaining".
In cooperative negotiation, agents work together to find a shared solution. They ask for help and if they get it, they give a trust degree to that agent.

42 seconds

Slide 2: Title: Project Overview
To begin, we will have a quick overview of the project's main idea. Then, we'll explore similar projects briefly. Afterward, I'll go in depth into the solution implemented in this project, including the application design, the technologies employed, and the testing process.

26 seconds

Slide 3: Title: Why this subject?
This project was chosen for two important reasons. Firstly, the tourism industry, especially in Romania, has seen a rise in demand for vacations following the COVID-19 quarantine period. This project aligns perfectly with the current theme. Secondly, I personally have experience with travel (more or less), making it easier to introduce a field I am familiar with. Additionally, multi-agent systems (MAS) are gaining popularity, and their ease of use and well-documented nature made them an excellent fit for this web-based system.

50 seconds

Slide 4: Title: Overview of Similar Projects
In exploring alternative solutions, I studied projects such as MAPWEB, CASIS, and MATRES. Understanding these "competitor systems" helped identify their disadvantages and the strategies employed to enhance functionality. For example, MATRES had two versions. In the last one they added the improvement of recommendations by assumption. At the time when agents didn't get help, they could assume some details about the user's preferences in order to finish the task.

45 seconds

Slide 5: Title: Proposed Solution
My solution involves developing an application that offers users a complete vacation package. The more the users expose their preferences, the more satisfying will be the results of the multi-agent system for them.
The project's architecture was designed to be simple, with a client web application and a server housing the business logic. A key element was the implementation of a multi-agent system to process user requests effectively.

33 seconds

Slide 6: Title: Application Design
The application follows a 2-tier architecture, consisting of the client (graphical interface) and the server (which handles interactions with the database and the MAS system). The user sends a request from the UI, which is then passed to the server via the MVC controller. The Web API controller receives the data and forwards it through various services until it reaches the MAS system. I will provide a more detailed explanation shortly.

41 seconds

Slide 7. Database Design

- A significant effort was dedicated to creating a functional and realistic database, close to real-world scenarios for the three services: flights, properties, and tourist attractions.
- Flight service involved creating weekly flight schedules, incorporating countries, cities, airports, and airplane flight programs.
- The database was designed to facilitate the multi-agent system's search for optimal solutions, allowing efficient parsing of table relations to find the best matches.
- Furthermore, the database had to support the smooth functioning of the multi-agent system, including recorded recommendations, agent interactions, and other related information. There was a certain level of dependency between the multi-agent system and the database, adding complexity to the database design.

1 minute 6 seconds

Slide 8: Title: Core Technologies
The project utilized several core technologies:

- ActressMAS: This framework was employed for agent modeling.
- .NET: The project was developed using the .NET programming environment.
- ASP.NET Core: This technology facilitated the creation of the web application and API.
- Fluent Migrator: It assisted in creating database migrations and populating the database.
- OpenTripMap API: This API provided valuable tourist attraction data for Europe.
- PostgreSQL: The chosen database management system.

47 seconds

Slide 9: Title: Development Stage

During development, the project's code was aligned with the ideas expressed in the design stage. One of the most exciting processes was the multi-agent system receiving a request to provide a complete vacation package. When an agent receives a message, it follows a common logic of action, branching out based on the instruction received. Agents consider the user's preferences and the "*Custom Agent*" rating derived from clients' exclusive feedback. Agents select tasks they excel at solving, and if they find the information in their own database, they notify the "Coordinator" agent. If not, they seek assistance from other agents, encouraging broad communication. The "Coordinator" receives notifications on completed services until the package is ready to be sent to the customer.

1 min 12 sec

Slide 10: Best recommendation? Hmm...

For choosing the best offer, the agents used Euclidean Distance between user preferences and each potential offer from agent's own database. Here we talk about multi-objective optimization and finding the most optimal condition, that is not the same as one-dimensional decision. The biggest dilemma here was *Pareto front* concept where agents would be forced to prioritize user preferences so that if the best solution cannot be found, less important specifications would be ignored.

In order to eliminate the idea of ignoring any customer preferences, the decision was taken to select the solution that matches most of user's preferences.

53 seconds

Slide 11: The agent – the expert

The application has been designed so that agents can gain experience in the services they provide. This will help them next time to choose a task to perform. All the secret is in the feedback provided by the client and the interaction between agents. If the client is satisfied, then the agents increase their rating, and vice versa. The same goes for the trust mechanism between agents. If the customer has given positive feedback, then trust will increase in the agent who has helped him to complete the request.

Note> I should mention that it was also added that the older the customer feedback is, the lower its impact on the agent's rating will be.

54 seconds

Slide 11: Title: API Testing

Testing the application took an unusual approach.

Given the unpredictable nature of the results, which intended to conform to user preferences, a simple HTTP client application was created to simulate full holiday package requests.

For the purpose of monitoring and data analysis, log data was included in the API. Additionally, manual checks were performed in the database to ensure data integrity. After the completion of a flow, were analyzed the logs of the interactions between agents as well as their updated logged expert degrees.

50 seconds

Slide 12: Title: Graphical Interface Testing
The graphical interface has been subject to manual debugging and testing sessions. This included verifying the accurate transmission of user-entered data for processing and correct receiving of data.

18 seconds

Slide 13: Title: Conclusions
In conclusion, the project successfully achieved the goal of providing users with complete vacation packages. The multi-agent system efficiently handled user requests by communicating and adjusting their expertness degree. To facilitate database integration, another application was developed using Fluent Migrations to generate and populate the data.
The project adhered to a 2-tier architecture, with the Onion Architecture pattern implemented for the API and the MVC pattern for the web application.
The project also incorporated strategies for future extensions.

40 seconds

Slide 14: Title: Future Improvements
To enhance the project's capabilities, several future improvements were considered. For example, incorporating more advanced search algorithms such as linear search, binary search, interpolation search or hash table could speed up the process of finding the best deal.
Exploring the transition from HTTP communication to Web Socket could enable real-time communication between the client and server.
The use of ML.NET, a machine learning library in .NET, would allow training and estimating suitable recommendations using *matrix factorization* algorithm. Additionally, optimizing memory consumption and considering cloud services for the server-side API would be crucial for scalability and resource efficiency.

1 min. 5 seconds.

Slide 15: Title: Final words
Thank you for your attention!

12 minute aproximativ.