Slide 1

Hello everyone. My name is Mihailov Emilian and today I'll present the project that is dedicated to everyday users, that will be able to search for a complete vacation package. As a processing core will be used a multiagent negotiation.

22 secunde

Slide 2

As a summary, first the general idea of the proposed topic will be presented. After that we will quickly go through a short description of similar projects. After that the solution applied in the solution project will be presented - as well as the design of the application, the used technologies, and their testing.

27 secunde

Slide 3

This project was chosen because it is related to tourism domain. And now this is a current theme, at least in Romania. Everyone wants to take a holiday, after that period of quarantine of the covid.

The second reason was the fact that everyone has travelled in their life, including me. Respectively it is easier to make an introduction to a field you have some tangent with.

As for multi-agent systems, they are slowly becoming popular. Plus, they are quite easy to use, with all the necessary documentation.

MAS offers a new alternative solution for solving problems, so this project provided a good opportunity to look at how to work with a MAS in a web-based system.

54 secunde

Slide 4

As alternative solutions for the proposed subject can be listed MAPWEB, CASIS and MATRES. It was necessary to understand how some of the "so-called competitor systems" work in order to know what the disadvantages of such a system are and what tricks are used to improve 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.

47 secunde

Slide 5

The proposed solution was to create an application that would provide a complete holiday package to the user. The architecture of the project was designed to be as simple as possible, i.e. to have the client - i.e. a client web application and the server containing all the business logic.

An equally important element was the implementation of a multi-agent system to process the user request.

28 secunde

Slide 6

The design of the app is quite simple. The 2-tier architecture was used. Presentation layer is the first and the business layer is the second. It is the one that includes both the application and database layers. The business layer is actually considered as the server.

As shown in the diagram, the user makes a request from the GUI, the MVC controller sends the data to the server. There the data is received by the Web API controller and passing through several services, the data reaches the MAS system.

43 secunde

Slide 7

As core technologies were used - ActressMAS (framework for agent modeling), .NET as programming environment, ASP.NET Core for Web application creation and API. Fluent Migrator was also used to create migrations that generate the database and then populate it.

OpenTripMap is an API that helped to provide data about tourist attractions in Europe.

And last but not least, PostgreSQL - database management.

38 secunde

Slide 8

In the development stage, the code was applied according to the ideas expressed at the design stage. Here is one of the most exciting processes of the project: MAS receiving a request to provide a complete vacation package.

Upon receiving a message, agents will have a common logic of action, but it will have many ramifications. Depending on the instruction received in the message, agents will choose a branch.

The logic is quite simple. All agents, upon receiving a request from the user who wants a recommendation, will read from memory his preferences and also his "Custom Agent" rating. That is to say, the agent's expert rating is based on the current client's exclusive feedback.

Based on his rating, the agent will choose a free task that he is best at solving. If the agent finds the information in its own database, it will notify the "Coordinator" agent that a task type is ready. And if not, the agent will ask the other agents for help.

And here is where broad communication between agents comes into play. On receiving any type of request for a recommendation, the agent will check if the task in focus has not already been performed by other agents at the time of retrieval and if not, it will look for a suitable recommendation, store it in memory and the "Coordinator" will be notified.

And this is it, the "Coordinator" will receive notifications from the agents about each type of service completed, and when the package is complete, it will be sent to the customer.

2 minute, 10 secunde

Slide 9

The app testing was more unusual. Normally, when testing a component, for best clarity, an input is set, and the output is compared to some expectations. So, we know exactly what should come out.

The given project was made with the idea that the result is unpredictable, as long as it is as close as possible to the user's preferences. Respectively, the testing was also more unusual. It was necessary to create a simple HTTP Client application to make calls to the server. In a way, this application simulates sending requests for a complete vacation package.

At the same time, inside the API, some process logging data was included. So that it is easy to monitor the correct execution of the processes and allow data analysis. In addition, although not the most brilliant idea, was used to manually check in the database if everything arrived safely and was stored correctly.

Above is the communication log between agents, and below is a snippet of the log report showing the ratings of the agents after completing a flow.

1 minute, 20 secunde

Slide 10

To test the graphical interface, there were manual debugging and testing sessions. Testing included ensuring that the data entered by the user is correctly sent for processing.

15 secunde

Slide 11.

To sum up, the project achieved the goal of providing a complete holiday package for the user. A MAS was used as a system for handling user requests. For an easier integration of the database, an application was created to generate and populate the data in the database using Fluent Migrations. On the architecture side, certain standards were established from the start, such as the use of a 2-tier architecture (a client which is the graphical interface and the server which includes the interaction with the database and the MAS system). It was also established to use Onion Architecture for the API project and to use the MVC pattern for the web application.

Also in the development process, some strategies to extend the application were established.

48 secunde

Slide 12.

For example, the possibility of applying more advanced search algorithms has been considered as a means of modernization. At present, agents take all the data they have in their own databases and check the best offer. But this process could obviously be accelerated by using for example Linear search, Binary search, Interpolation search or Hash table.

Another improvement would be to move from REST to Web Socket architecture. The biggest advantage of this being the creation of a channel between client and server and maintaining continuous real-time communication.

.NET uses the ML.NET library for machine learning. This framework uses matrix factorization for training the data and estimating a suitable recommendation.

Another rather challenging optimization would have been the use of cloud services for the server-side API.

Optimizing memory consumption is a very important aspect, especially if you want to integrate with a cloud service. It must be taken into consideration that every process that is produced consumes memory. If you want to develop this application, a decisive step would be to optimize access to resources, but also to vigilantly monitor processes and their consumption rate.

1 minute 35 secunde

Slide 14.

This was the presentation of the project on "Search for a complete vacation package using multi-agent negotiation". Thank you for your attention.

14 secunde

12 minutes of presentation