

Architectural Design

Conceptual Architecture

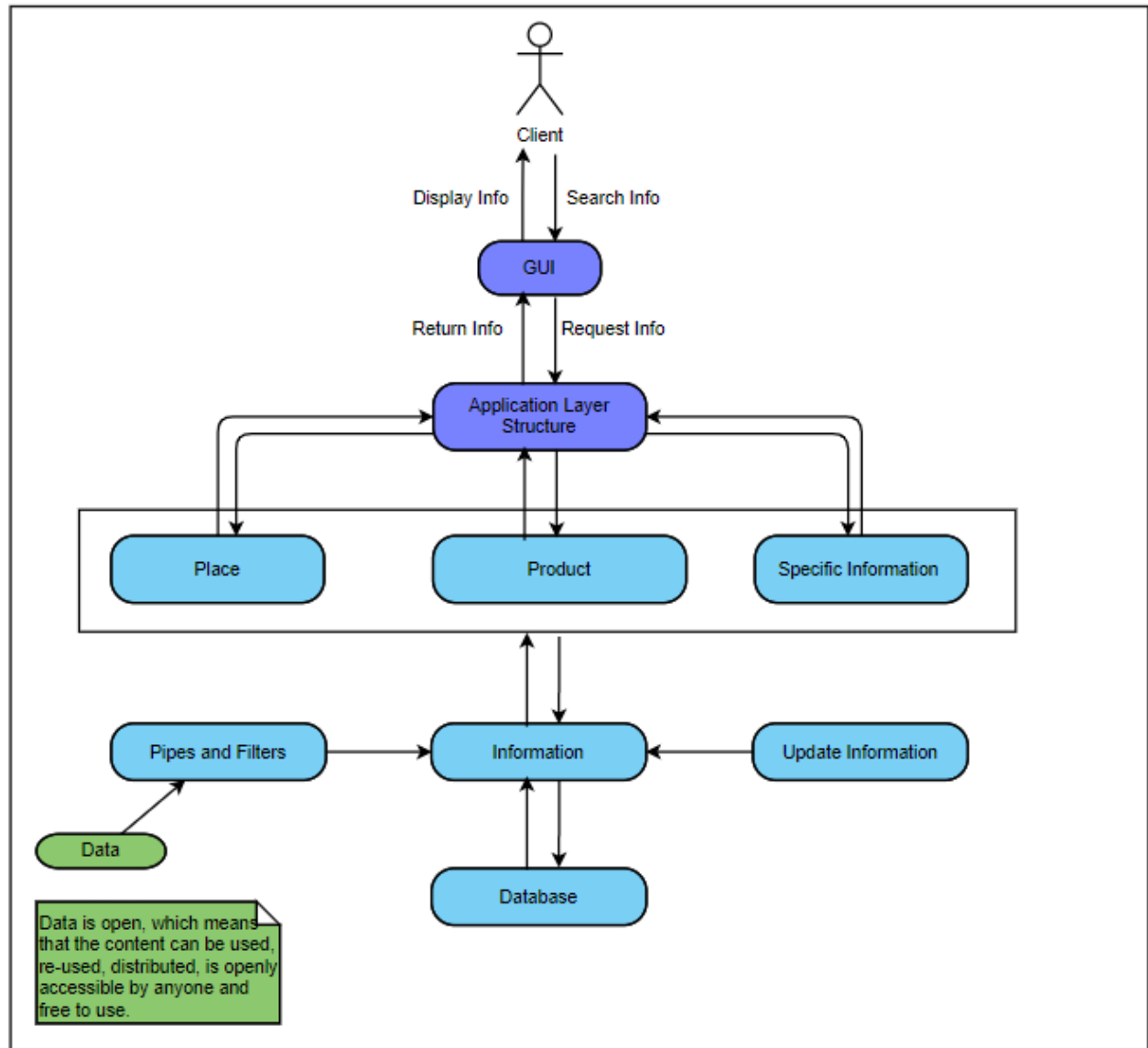


Figure 1

On figure 1 is shown the diagram for the conceptual architecture of the application, which by separating the whole system's architecture into smaller parts will give its conceptual integrity and technical consistency. It describes the responsibility for each component and how they intercommunicate with each-other.

In this application it is used a client-server design, which means the server distributes and is in charge of the services given to the client.

In the diagram we can see the logical separations of the application. In the first part we have the client from the application layer of its device trying to get a response for the request made in the web site. On the second part we have the interaction of the client with what he wants to search for in the application (product, place or some other information). And on the third part we have the base format of the application which is to provide services to the upper part. The information is going to be taken from the database, it is going to be updated and sent to the middle layer. If a client is interested in a product, it'll be shown all the necessary information for the product, the price it has from each tech-store and the location starting from the closest distance of the user.

All this information will be transmitted from the lower layer to the upper layer, each of them giving a unique service to it.

The application uses pipes and filtering, so every information will be classified in multiple, separate files.

The user will be able to give its personal thoughts on the application by rating it, which takes place on the “Update Information” field and will be saved further in the data files.

Executorial Architecture

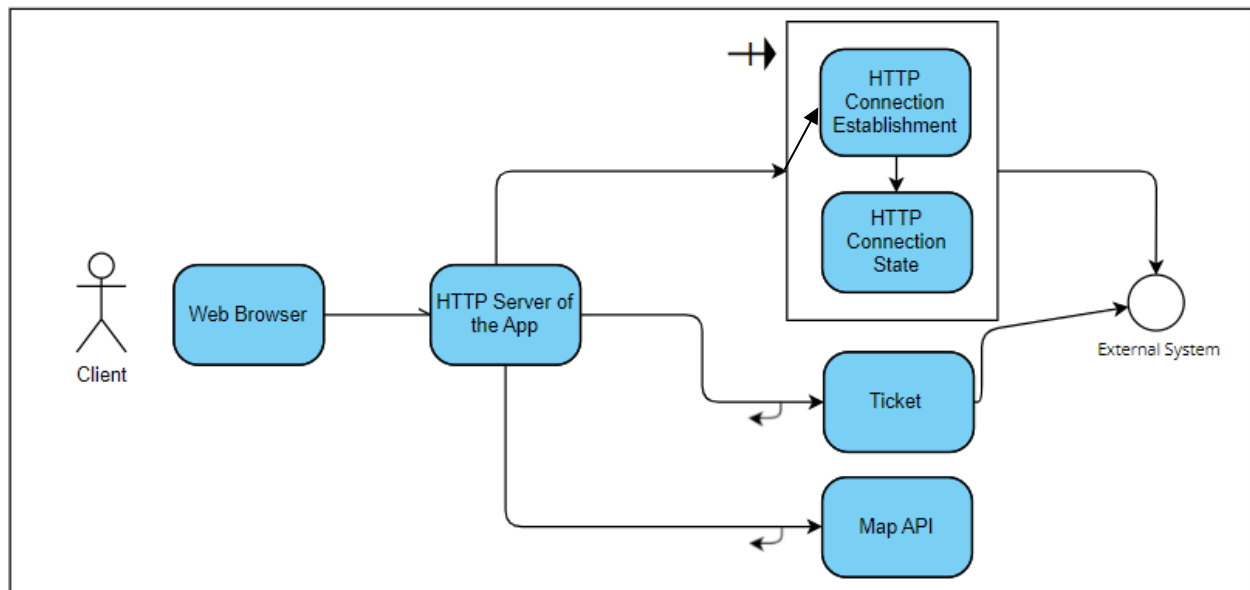


Figure 2

On figure 3 is shown the diagram for the executorial architecture of the application, which groups the functionalities into subsystems and components by (loosely) connecting them all and shows how they communicate with each-other. The overall structure of the system is done through pipes and filters and captures the components that exist during run-time. It is composed of elements, processes and threads.

When the user wants to find the desired product from a tech-market, it will do so by searching for it in the web browser (any web browser will do) from its device, via asynchronous calls. Then the HTTP Server will continue the communication via synchronous calls to the API and by ticketing system, to keep the workload efficient and deal with different incidents.

Implementation Architecture

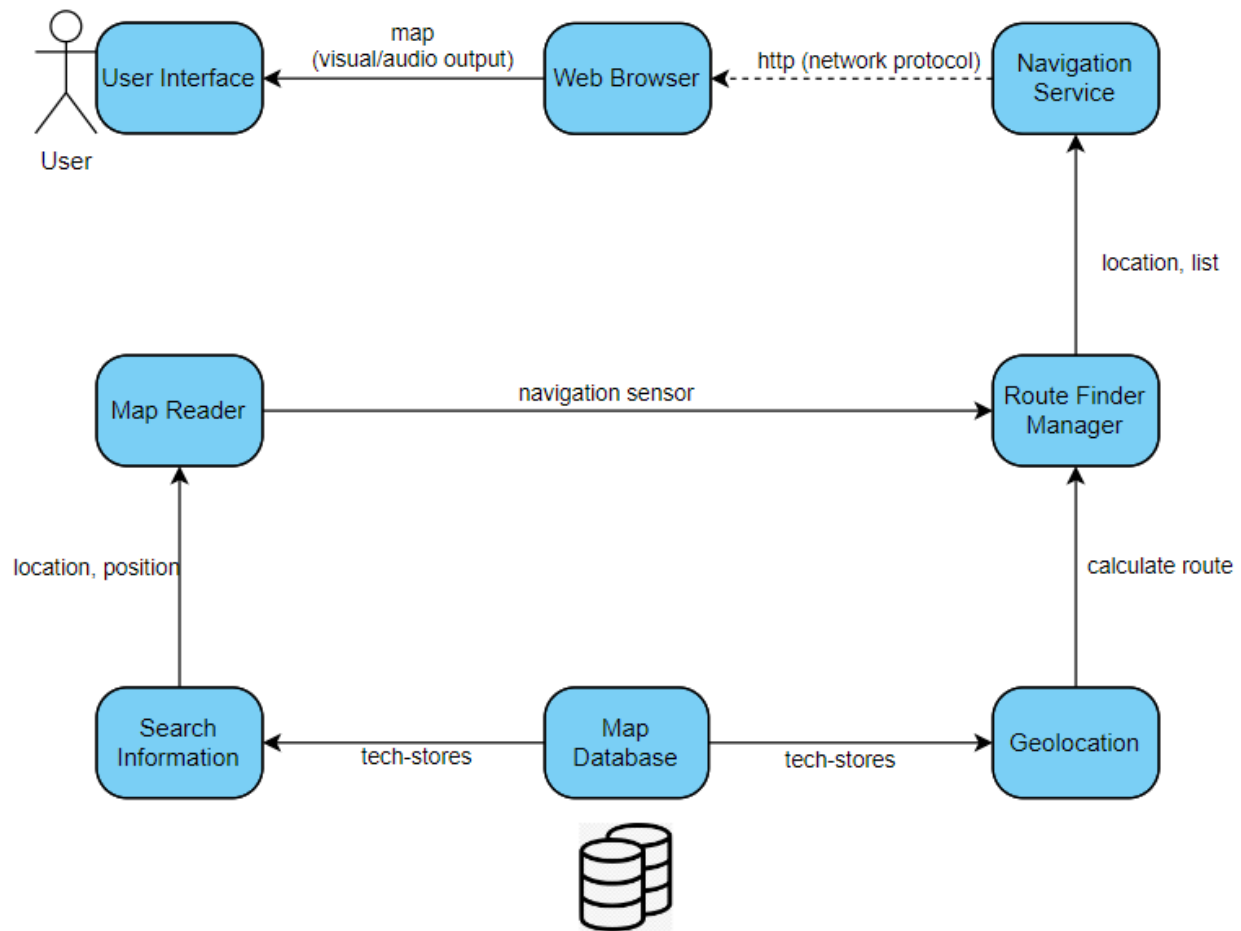


Figure 3

On figure 3 is shown the diagram for the implementation architecture of the application, which tells how the app is built. The components, elements, connectors all represent an implementation perspective on what to create in the application, how to structure the modules, how to interconnect them between each-other and how they should behave.

The database has all the information for the tech-stores in the map and it passes the same to the search information and the geolocation. The geolocation calculates the optimal route between the store and the user. Then the route finder will list the places by using different parameters (cheapest product or shortest path) and through the network protocol it will output it to the user.

Technologies used for implementation:

For this application we use MVC (Model-View-Controller) architectural pattern, for separating the components depending on their functionalities. That's because when the client enters a request in the browser's URL it will go to the server host by a controller. The displayed information to the client's interface is handled by the View, meanwhile the Model which is the lowest level of in the pattern is responsible for logically maintaining the data. As an MVC framework in the application is used Django and Leaflet.