Architectural Design

This document represents an overview of the base architecture of our web application using different architectural views such as the conceptual, executive and implementational architecture. It also provides some base guidelines for the development process of the application.

This application is a Java based web application. The client connects directly to the web app server through a port. The page on the client side is built with Angular which renders pages in response to the user actions. All the data we need to store will be stored in MongoDB – noSQL database management system. In the database, information about each hotel and client will be stored and only the users who are logged as admin can make changes in the hotel database.

The final application will have a hybrid architecture.

1. Conceptual architecture

The conceptual architecture of our project focuses on the domain-level responsibilities. We will use it as initial architectural design upon which the further development of the project will be based. We will design it by analysing the functional requirements.

The design process:

1) Identifying key concepts from the requirements
   - Hotel locations
   - Database
   - Administrator
   - Adding/removing hotels
   - Users
   - Searching for hotels
   - External services as Booking.com
   - Map
   - User management
   - Register new users
   - Log in/Log out
   - User preferences
   - User credentials
   - Admin credentials
   - Booking
   - Navigation tool

2) Assigning every concept to a category (Data, function, stakeholder, system, hardware, abstract concept)
   - Hotel locations – Data
   - Database – Data
   - Administrator – Stakeholder

- Adding/removing hotels – Function
- Users – Stakeholder
- Searching for hotels – Function
- External services as Booking.com – System
- Map – Data
- User management – Function
- Register new users – Function
- Log in/Log out – Function
- User preferences – Data
- User credentials – Data
- Admin credentials – Data
- Booking – Function
- Rout – Data
- Navigation tool – Abstract concept

3) According to this list, we will proceed with the creation of the diagrams, which will be included in the folder.

4) According to the diagram, the components for the conceptual architecture and their responsibilities are the following:
   - AppUI: Show hotels, display map, display booking system
   - AdminPanel: List map
   - NavigationServiceL Navigate through hotels on UI, route computing
   - AdminService: Add hotel, remove hotel
   - Booking: redirect to booking page
   - UserManager: Register user, log-in user
   - GeoinformationManager: Add hotel, remove hotel
   - Search: Search hotels
   - Database: Store persistent data
   - RoutFinder: Compute route

2. Execution view

The purpose of the execution (run-time) architecture is to analyze the concurrency requirements and identify processes. We will define the system's process architecture in terms of its active classes and instances and the relationship with the os threads and processes.

In the diagram execution architecture #1, firstly we identified the components of the executional architecture which are GUI, service, route finder, booking, the database and the external system with which the booking component communicates (in our case that is booking.com), as well as how these components communicate. In the next diagram which is binding the conceptual and execution architecture, we visually represented the conceptual architecture, identifying what group of components are mapped into the execution architecture. Moreover, there are two diagrams representing the execution behavior which describe two use cases, booking a room and searching for a hotel.

3. Implementation View

The implementation view describes the system in terms of implementational subsystems and elements such as files, source code and data.

The presentation layer is the frontend of the application. As mentioned before, we will use Angular framework as a presentation layer, which will display the data. This layer provides the presentation logic, not the business logic. The presentation layer is separated in a view layer, controller layer and a frontend service layer which makes the client MVC-capable.

The application layer will be implemented in Java and will connect the presentation layer to the data layer. Spring boot is the component (collection of services) which will deliver data between the data layer and the presentation layer.

The database system will represent the data layer. MongoDB will be used as DBMS.

The diagrams for this architectural view are uploaded in the same folder.

4. Architectural Styles

1) Microservice architecture

This architectural pattern divides the application into various independent service units. For the purpose of our application, each microservice will be focused on a single business capability. There will be three individual microservices(modules) which will communicate with each other. Register, Search and Booking will represent those three modules. The communication between those modules will be stateless. Each module will use a separate data table. Each module will be assigned with one specific business goal. With this architecture we will achieve loose coupling so that changes in one module will not affect the other modules.

2) Notification architecture

Our implementation of the notification architecture will be in the search component, which with the debounce time and pipe functions in angular will trigger the database upon entering a letter in the search bar, every 500 milliseconds.

3) Distributed architecture

It is a distributed architecture because it has a three tier, the middle tier will do all the processing and decision making (it will receive requests form clients, process the data and the forward it to the server).