

# Architectural Design

Using many architectural viewpoints, including the conceptual, executive, and implementational architecture, this document provides an overview of the base architecture of our web application. Additionally, it offers a few base guidelines for the development of the application.

Our application is a Java-based web application. Through a port, the client makes a direct connection to the web app server. MongoDB, a noSQL database management system, will be used to store all of the data that we need to keep. Each hotel's and client's information will be kept in the database, and only users who are logged in as admins can make changes to the hotel database. A hybrid architecture will be used for the final application.

## 1. Conceptual Architecture

Our project's conceptual architecture focuses on domain-level responsibilities. It will be used as initial architectural design upon which the further development of the project will be based. Analyzing the functional requirements will help us with the design.

Design process:

- i. Identifying key concepts from the requirements
  - a. Hotel locations
  - b. Database
  - c. Administrator
  - d. Adding/removing hotels
  - e. Users
  - f. Searching for hotels
  - g. External services as Booking.com
  - h. Map
  - i. User management
  - j. Register new users
  - k. Log in/Log out
  - l. User preferences
  - m. User credentials
  - n. Admin credentials
  - o. Booking
  - p. Navigation tool
- ii. Assigning every key concept to a category (data, system, stakeholder, function, hardware, abstract concept)
  - a. Hotel locations - data
  - b. Database - data
  - c. Administrator - stakeholder
  - d. Adding/removing hotels - function
  - e. Users - stakeholder
  - f. Searching for hotels - function

- g. External services as Booking.com - system
- h. Map - data
- i. User management - function
- j. Register new users - function
- k. Log in/Log out - function
- l. User preferences - data
- m. User credentials - data
- n. Admin credentials - data
- o. Booking - function
- p. Route - data
- q. Navigation tool - abstract concept
- iii. In accordance with this list, the diagrams will be created.
- iv. According to the diagram, the components for the conceptual architecture and their responsibilities are the following:
  - a. AppUI: Show hotels, display map, display booking system
  - b. AdminPanel: List map
  - c. NavigationService Navigate through hotels on UI, route computing
  - d. AdminService: Add hotel, remove hotel
  - e. Booking: redirect to booking page
  - f. UserManager: Register user, log-in user
  - g. GeoinformationManager: Add hotel, remove hotel
  - h. Search: Search hotels
  - i. Database: Store persistent data
  - j. RouteFinder: Compute route

## 2. Execution View

The analysis of the concurrency requirements and identification of processes are the goals of the execution (run-time) architecture. In terms of the system's active classes and instances, as well as how they relate to the OS threads and processes, we will define the system's process architecture.

In the diagram Execution Architecture, firstly the components of the executional architecture were identified, 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 the communication of these components. In the diagram which is Binding Conceptual and Execution architecture, we visually represent 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 is a description of 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. Angular framework will be used as a presentation layer, which will display the data. By this layer, the

presentation logic is provided, not the business logic. The presentation layer is separated into 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.

#### **4. Architectural Styles**

##### **i. Microservice Architecture**

The application is divided into various independent service units using this architectural pattern. For the purpose of our application, each microservice will be focused on a single business capability. Register, Search and Booking will be three individual microservices(modules) which communicate with each other and the communication between them will be stateless. Each of these modules will use a separate data table, and will be assigned with one specific business goal. With the help of this architecture loose coupling will be achieved which means that if there are changes in one module, the other modules won't be affected.

##### **ii. 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.

##### **iii. 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 from clients, process the data and then forward it to the server).