Универзитет „Св. Кирил и Методиј“ во Скопје
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

SOFTWARE DESIGN AND ARCHITECTURE

Architectural design

Prepared by team: **ADELE-CORP**

Authors:	Mentors:
1.Drin Kadriu (211505)	проф. д-р Љупчо Антовски.
2.Leon Saraqini (211508)	проф. д-р Петре Ламески
3.Engjell Vlashi (211519)	
4.Ardit Sakipi (211545)	
5.Eroll Sakipi (211590)	

Contents

List of pictures

List of tables

Introduction

The "ADELE-CORP" application is a comprehensive digital platform aimed at preserving, exploring, and promoting the rich tapestry of North Macedonia's cultural and historical sites. The application serves as an interactive guide that provides detailed information about various landmarks, historical events, and cultural narratives that form the unique heritage of the region.

The scope of the application encompasses an extensive database of heritage sites, complete with multimedia galleries, descriptions, historical facts, and visitor information. It facilitates educational endeavors, tourist activities, and scholarly research by offering user-friendly navigation and personalized experiences based on user preferences and interests.

Designed to be accessible for a diverse audience range, including local residents, international tourists, educators, students, and history enthusiasts, the app aims to foster a deeper understanding and appreciation of North Macedonia's historical and cultural significance.

Through this application, we are committed to leveraging technology to safeguard cultural narratives and ensure that the legacy of North Macedonia's historical assets is available to a global audience, now and in the future.This architectural design document details the technical framework and strategies employed to realize this vision, ensuring that the application is robust, scalable, and capable of delivering a rich, interactive experience to its users.

Conceptual Architecture

The conceptual architecture provides us with the initial architectural design and domain-level responsibilities that emerge from the functional requirements of the stakeholders. We generally created an initial picture of the application as a whole and as an initial step, we examined the functional and non-functional requirements and marked the most important concepts and words shown in the table below.

Data	Function	Stakeholder	System	Abstract Concept
Site Information	Provide details about heritage sites	Users, Historians, Tourists	Content Management System	Information Repository
User Preferences	Tailor experience to user interests	Users	User Profile Management	Personalization
Geolocation Data	Locate and navigate to heritage sites	Users, Tour Operators	Mapping Service	Spatial Data
Historical Timelines	Present historical context of sites	Educators, Students	Educational Tools	Contextual Framework
Multimedia Content	Enhance site listings with images and videos	Content Creators	Media Server	Digital Asset
Reviews and Ratings	Gather user feedback on sites	Users, Site Managers	Review System	User Engagement
Authentication Data	Secure user accounts and data access	Users, Administrators	Authentication Service	Security
Analytics Data	Analyze user engagement and site popularity	Administrators, Marketers	Analytics Engine	Business Intelligence

Table1:Conceptual Architecture

The image below shows a view of the conceptual architecture and is represented by AppUI with Navigation UI Logic which connects to Search and the business logic and searches for the appropriate historical cultural and historical heritage, accesses the database, finds the coordinates, and returns that view.

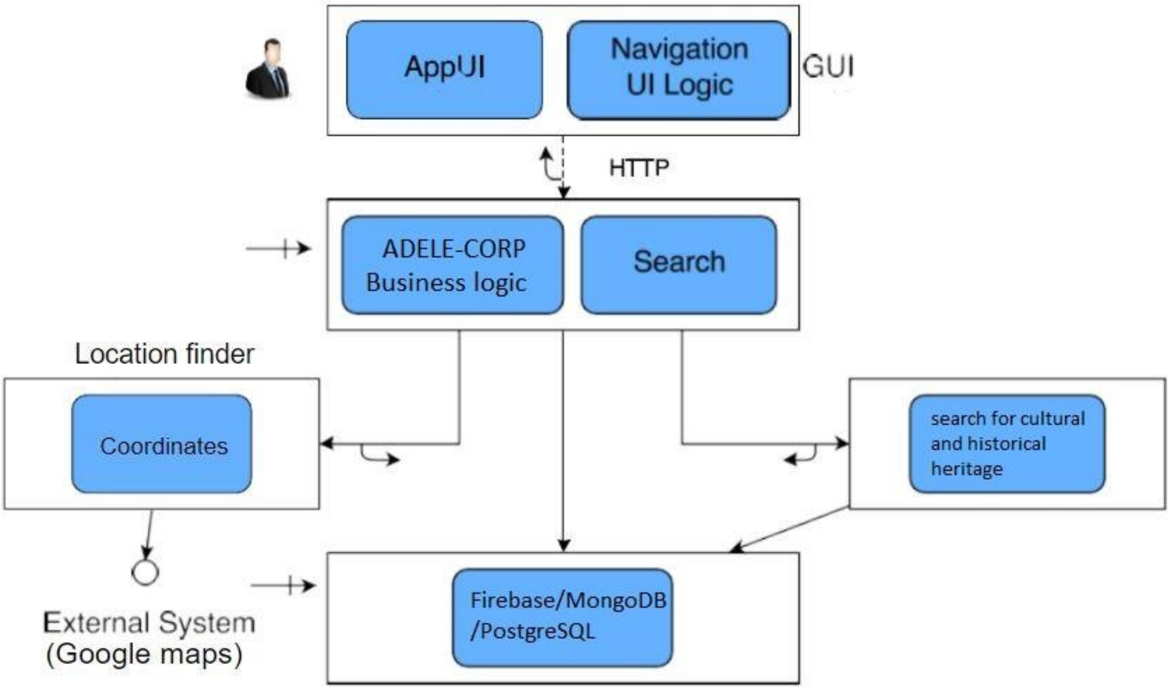
AppUI
Navigation UI Logic
GUI
HTTP
ADELE-CORP Business logic
Search
Location finder
Coordinates
search for cultural and historical heritage
External System (Google maps)
Firebase/MongoDB /PostgreSQL

Image 1.Conceptual Diagram

The next image shows the behavior as it moves through the system, that is, a behavioral model i.e., an exploration of behavior.

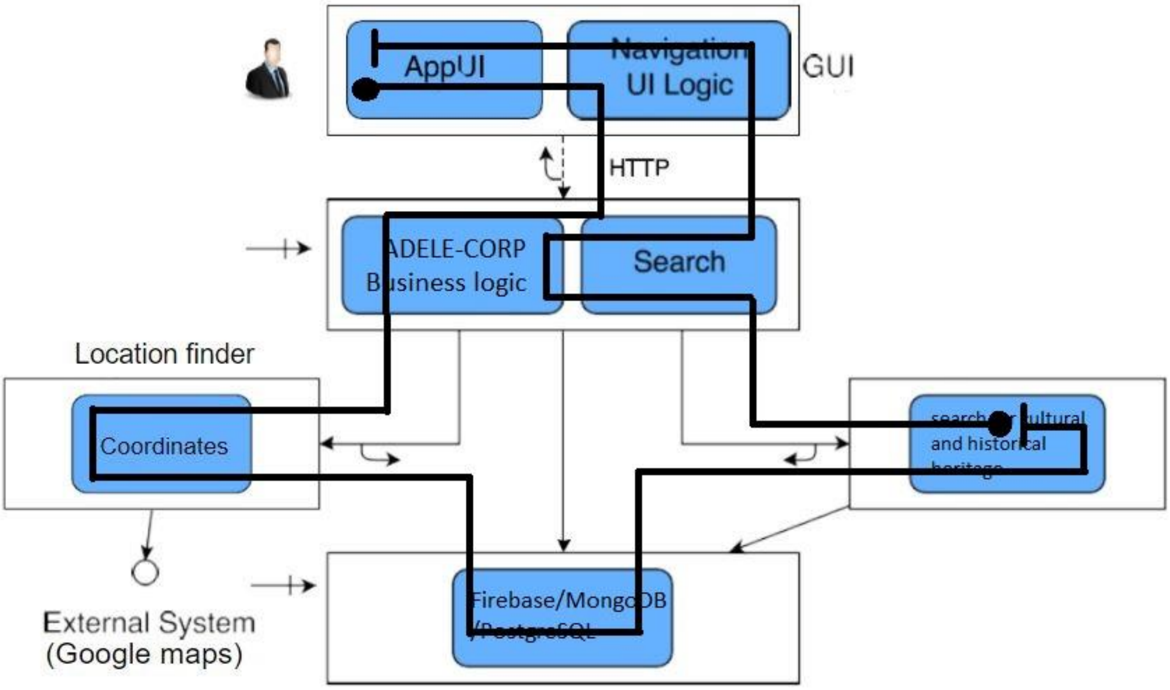
AppUI
Navigation UI Logic
GUI
HTTP
ADELE-CORP Business logic
Search
Location finder
Coordinates
search for cultural and historical heritage
External System (Google maps)
Firebase/MongoDB /PostgreSQL

Image.2 Behavioral model

Execution Architecture

The execution architecture refers to what the system looks like during execution. The following 2 images represent diagrams of what the view of the system's execution architecture looks like.

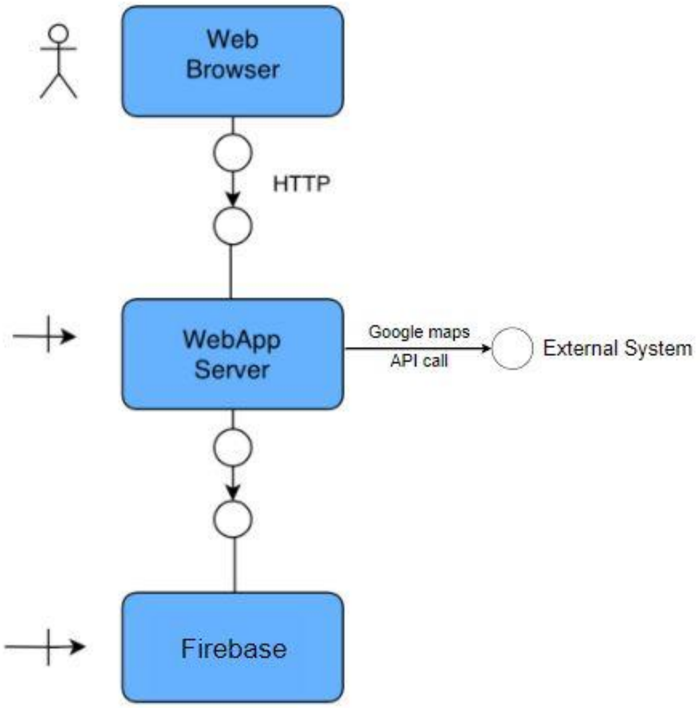

Image 3 Execution architecture

We have also the following diagram

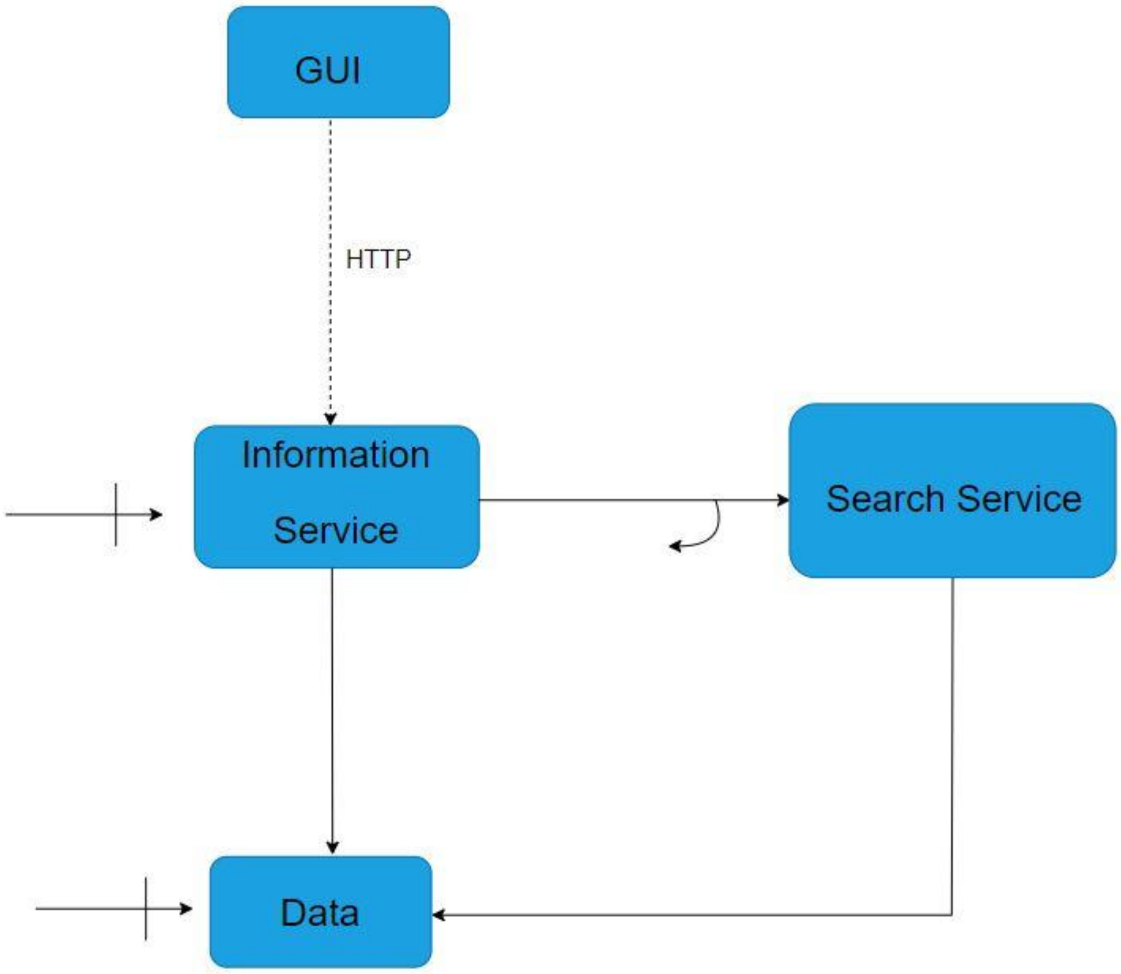

Image 4 Execution architecture

Implementation Architecture

The implementation architecture focuses on how the system is built, with all the technical elements that need to be implemented.

In our case, the Web Browser that will use the application will send an HTTP request to the server, where Tomcat intercepts the request and carries it to the appropriate servlet. The request is processed in the Spring Boot container whose application components make API calls to the database (Firebase) and to Google Maps to display the location of the requested store(s).

The response through the servlet is carried to the web browser where the React application with the appropriate components displays the response as a map for the user with the located objects, as well as information and filters for additional searching.

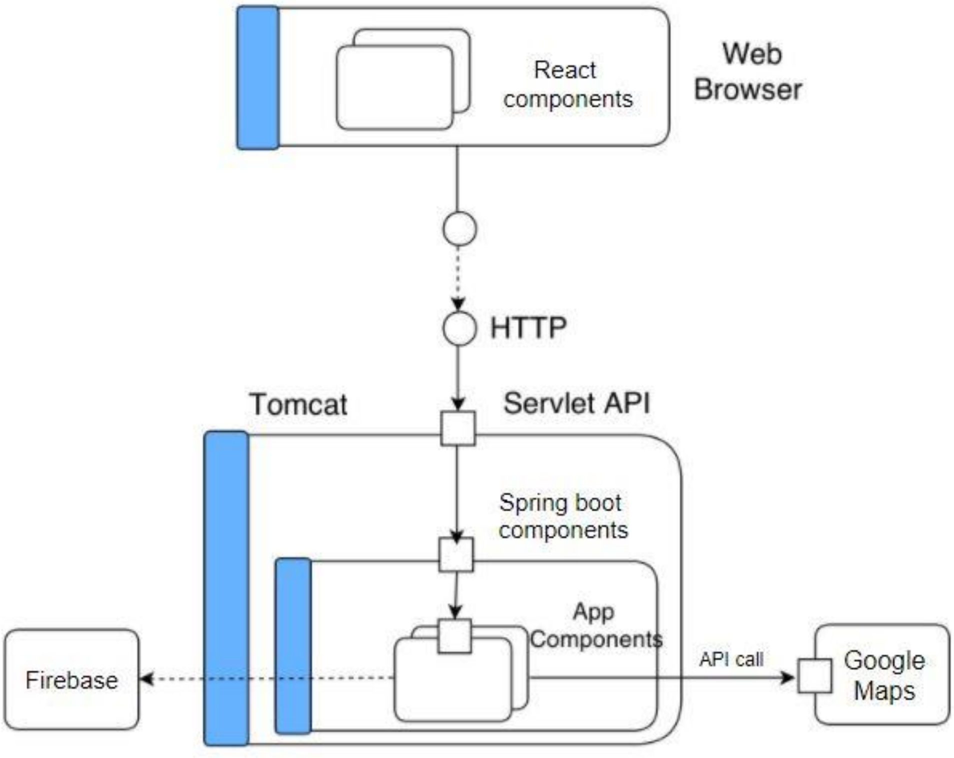

Image 5 Communication between web-browser and server

A more specific view of the implementation architecture is given in this image.

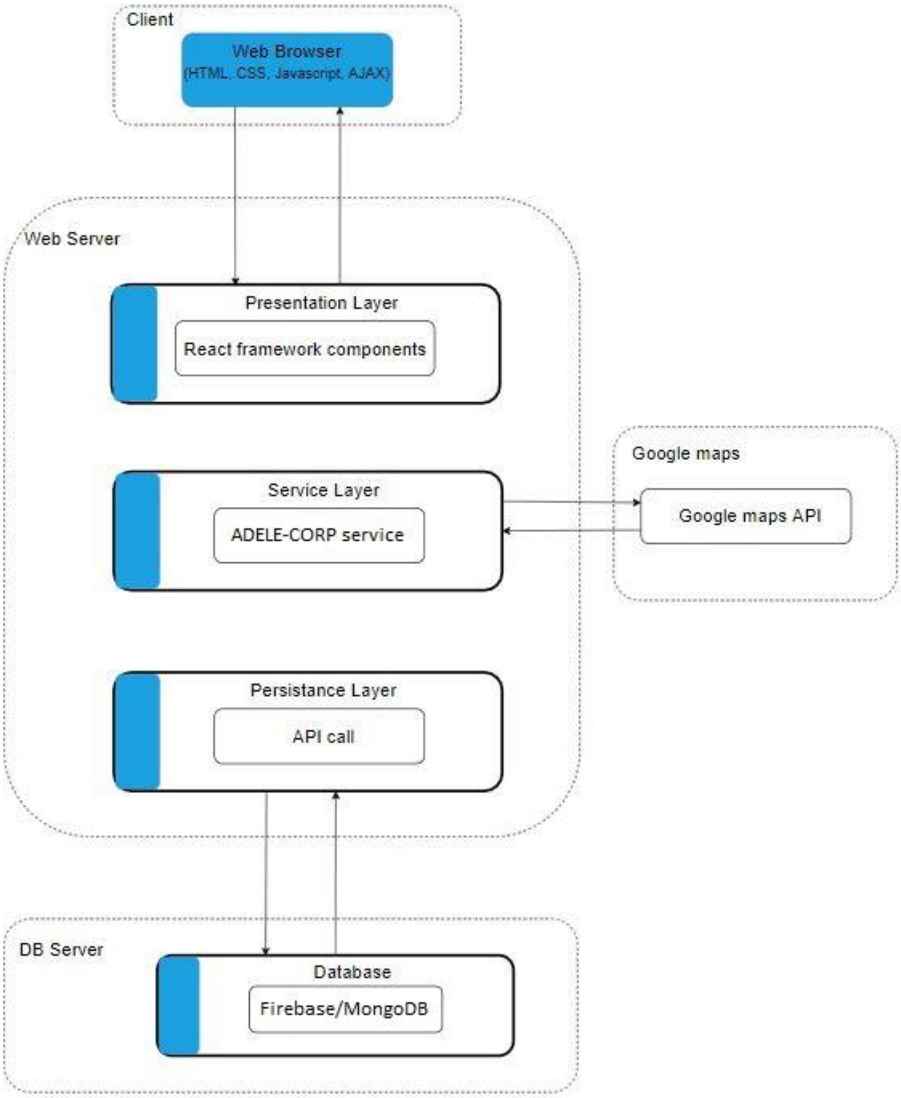

Image 6 Implementation architecture

Finally, we have made a sequence diagram that describes the process of finding a cultural and historical heritage place and how it "flows" through the system.

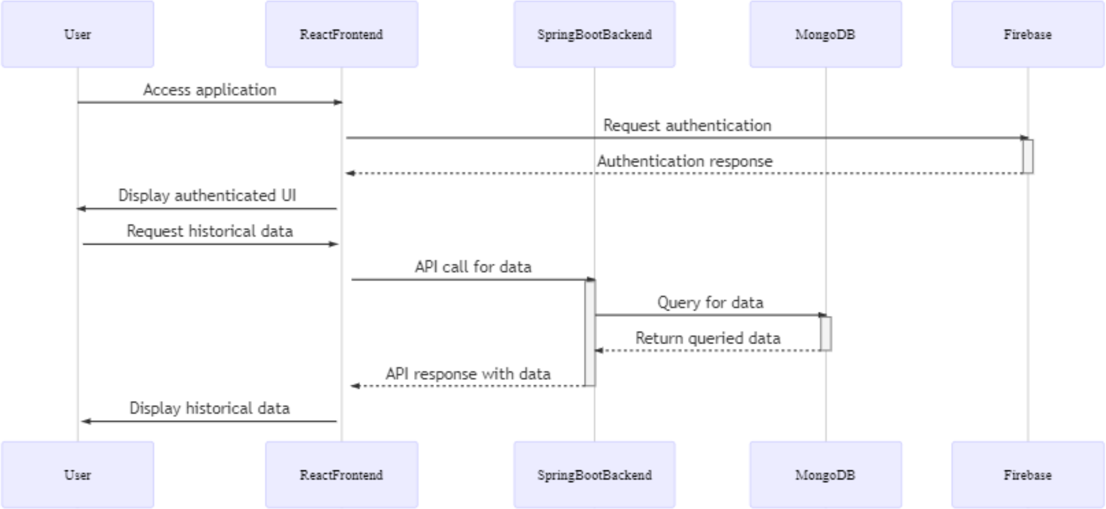

Image 6 Sequence Diagram

Appendix A : Glossary

Actor: a person performing a specific role, a software system, or a hardware device that interacts with a system to achieve a useful goal. Also called a user role.

Class:a description of a set of objects having common properties and behaviors, which typically correspond to real-world items (persons, places, or things) in the business or problem domain.

class diagram: an analysis model that shows a set of system or problem domain classes, their interfaces, and their relationships.

constraint a restriction that is imposed on the choices available to the developer for the design and construction to the developer for the design and construction of a product. Other types of constraints can restrict the options available to project managers.

context diagram: an analysis model that depicts a system at high level of abstraction at a high level of abstraction. The context diagram identifies objects outside the system that exchange data with the system, but it shows nothing about the system's internal structure or behavior.

user class: A group of users for a system who have similar characteristics and requirements for the system. Members of a user class function as actors when interacting with the system through use cases.

user requirement a goal or task that specific classes of users must be able to perform with a system, or a desired product attribute. Use cases, user stories, and scenarios are common ways to represent user requirements.