Software Architecture Report

Date: January 30, 2024
Version: v1.0 (Commit ID: b746c29)

1. Purpose of the Software Project
    a. This is a web application developed using ASP.NET Core MVC, Entity Framework, and Identity Framework. The project's purpose is to provide a comprehensive booking platform that facilitates the reservation of various hotels.
    b. Fulfilled Capabilities: At the current stage, the project allows users to:
        i) Browse available services and resources.
        ii) Register an account and log in securely using Identity Framework.
        iii) Make reservations with specified time slots.
        iv) View and manage their reservation history.
        v) Administer services, users, and reservations through an administrative dashboard.
2. Guides
    a. Running the Project Locally: To run the project locally, follow these steps:
        i) Clone the repository: proiect-inginerie-software-4ra/Traveling-Platform at main · inginerie-software-2023-2024/proiect-inginerie-software-4ra (github.com)
        ii) Open the project in Visual Studio or your preferred IDE.
        iii) Set up the database by running Entity Framework migrations.
        iv) Update the connection string in the appsettings.json file.
        v) Build and run the application.
        vi) Building the Project
    b. Deploying the Project: Running the application using IIS pipeline (the green start button at the top bar of the IDE) will automatically run the application by opening it in the local browser, on port 3000 (make sure it's available)
    c. Contribution Guide
        i) Patterns Used in the Application
            1. MVC (Model-View-Controller): Used to separate concerns and organize the codebase.
            2. Dependency Injection: Employed in managing component dependencies.
        ii) To add a new feature:

1. Create the database model if required (new entities) and add it to the database context file, then create a new migration and update the database
2. Create a new end point in the corresponding controller and implement the logic
3. Create a new view if needed

d. Application Entry Point

- i) HomeController: starting from the welcome page, through the buttons in the upper nav bar various features can be observed
- ii) Data Sources - Entity Framework: Acts as the primary data source, utilizing a SQL Server database.
- iii) Data Inputs: User inputs are primarily through web forms. Validation is implemented at both client and server sides to ensure data integrity.
- iv) Configuration Files - appsettings.json: Contains configuration settings, including database connection strings and other application-specific settings.

e. High-Level Diagrams of the Architecture

- i) User/Data Journeys
  1. User Registration:
     - a. User navigates to the registration page.
     - b. Enters required information.
     - c. Submit the registration form.
  2. Reservation Booking:
     - a. User logs in.
     - b. Explores the Countries – Cities – Hotels pages in search for a destination
     - c. From the available rooms at desired dates, the user choses a trip for a time interval
- ii) Most Valuable Output: The most valuable output of the system is the seamless user experience in booking and managing reservations

f. Deployment Plan

- i) As stated before, the application is currently running locally, with the project files available on the main branch of the Github repository
- ii) CI: a CI workflow is set up using GitHub Actions to build and restore dependencies. The workflow triggers on pushes to the "main" branch and pull requests targeting the "main" branch. This is available and editable via the .yaml file

g. External Dependencies

   i) APIs Used: As of today, the application doesn’t use any external APIs

   ii) Libraries

      1. Bootstrap: Utilized for responsive and aesthetically pleasing UI.

      2. Entity Framework Core: Responsible for querying and designing the database

   iii) Dependency Vulnerability – As of today, there is no known dependency vulnerability