Team 1 - Milestone 2

Travel Berlin Web Application Design Document

Date: July 2nd, 2019

Contact Information:

Suma Cherkadi (scherkadi3@gatech.edu),
Sanjana Badhya (sbadhya3@gatech.edu ,
Salina Nihalani (snihalani7@gatech.edu),
Michelle Hou (houml812@gatech.edu),
Joha Kim (jkim3454@gatech.edu),
Andrew Babbitt (ababbitt3@gatech.edu)

Table of Contents

1. Introduction
    - 1.1 Background
    - 1.2 Design Goals
2. Architecture
    - 2.1 Introduction
    - 2.2 Data Model
    - 2.3 User-Interface

1 Introduction

1.1 Background

Traveling to a new country can be difficult and intimidating. In planning for their journey, travelers scour through blog posts and articles trying to identify the best places to visit. Even with intense research, travelers do not know what their experience will be like at these attractions.

Berlin is a city with many of these attractions. Thousands of tourists visit these attractions every day and develop varying opinions about them. These varying experiences are insightful and could help other travelers decide which attractions they would like to visit.

1.2 Design Goals

Travel Berlin is a web application for travelers to get to know Berlin as they are exploring the dynamic city. The application uses crowdsourced data to consolidate different users' experiences at different attractions in Berlin. Users can generate reviews to places they have recently visited. Users can also visit published pages for different attractions and view all the experiences other travelers have had.

2 Architecture

2.1 Introduction

Database Tier - Responsible for storage of user and profile data. Data storage implemented in a backend JSON file with each user as a nested JsObject. Also responsible for the storage of attraction information.

Controller Tier - Responsible for performing actions on data from database tier and pushing data to the front-end tier. Performs actions in a Scala controller class and is accessed through Scala routes from the front-end.

Front-End Tier - Responsible for display data and managing user interface. Fetches data from database through controllers and utilizes data to navigate the user through the application.

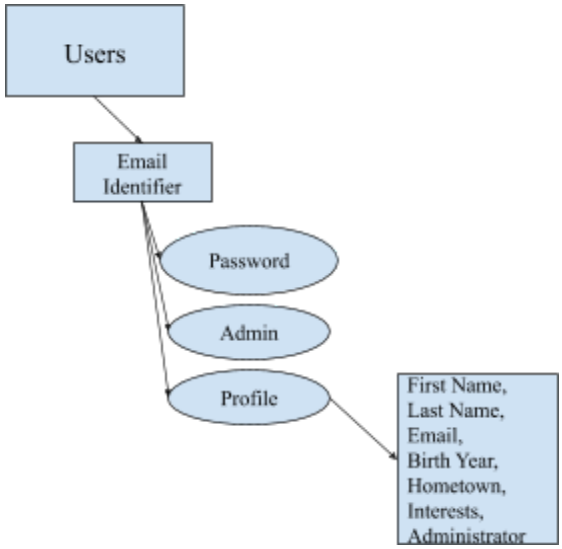
Quality Attributes of Architecture:

1. Flexibility - Separating the clients from the backend tier allows us to customize the performance of the backend and the frontend separately.

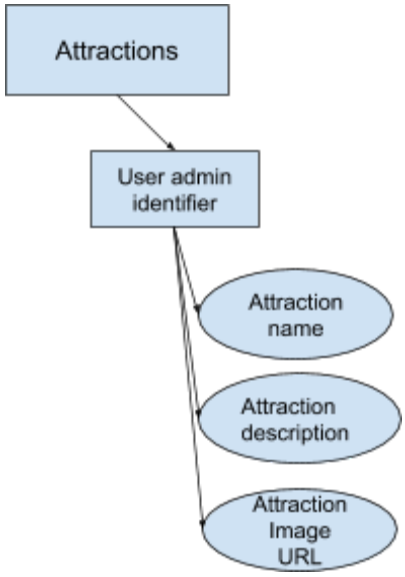
2. Modifiability - Changing the database to a more secure option in the future will not affect the usability on the client end of the application.

2.2 Data Model

The user database currently only stores 4 tiers of information: email, password, administrator authorization, and profile. This data is currently stored in nest JsObjects with each user's email as an identifier.



The attractions database currently stores 3 tiers of information: attractions, user admin information, and attraction information. The information is currently stored in nest JsObjects with each user's admin tag as an identifier.



2.3 User-Interface

Admin Screen

- The user is presented with options to either log in or register themselves.

Register

- If the user chooses to register, they will be prompted to enter their email and desired password. They will also need to select whether or not they wish to be an administrator. Once completed, they will be asked to log in.

Login

- The user will be asked to enter their email and password. If the credentials are valid, they will be redirected to the main page for this web application.

Landing Page

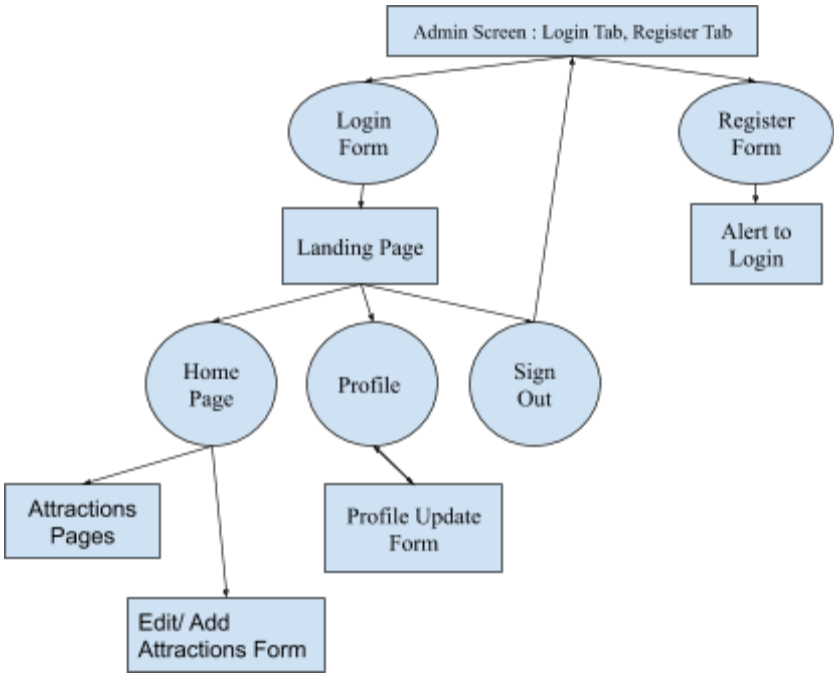
- Here, the user chooses between three tab options: Home, Profile, and Sign Out. If they choose to sign out, they will be redirected to the admin screen.

Homepage

- This is the homepage which will be populated with travel pages in Berlin.
- The user may select a travel page to learn more about the destination.
- If the user is an admin, they may edit existing travel pages as well as add additional pages.

Profile

- The user can update their profile credentials here. These credentials include first name, last name, email, birth year, hometown, interests, and administrator preference.


Admin Screen : Login Tab, Register Tab
Login Form
Register Form
Landing Page
Alert to Login
Home Page
Profile
Sign Out
Attractions Pages
Profile Update Form
Edit/ Add Attractions Form

Sequence UML Diagram for Login