



SAPIENZA
UNIVERSITÀ DI ROMA

DEPARTMENT OF INGEGNERIA INFORMATICA,
AUTOMATICA E GESTIONALE "ANTONIO RUBERTI"

PlaceAdvisor

LABORATORY OF ADVANCED PROGRAMMING

Professor:

Massimo Mecella

Students:

Cristian Santaroni 1800659

Francesco Fortunato 1848527

Academic Year 2022/2023

Contents

1	Introduction	2
2	User Requirements	3
2.1	Introduction	3
2.2	User Authentication and Registration	3
2.3	Integration with Google Photos	3
2.4	User Profiles	3
2.5	Points of Interest Discovery	3
2.6	Detailed Place Information	4
2.7	Chat with Support	4
2.8	User Feedback Feature	4
2.9	Kibana and Elasticsearch Dashboard	4
3	User Stories	6
4	Function Points	12
4.1	Complexity Tables for Function Points	12
4.2	Internal Logical Files (ILF)	13
4.3	External Interface Files (EIF)	13
4.4	Calculation of Function Points	13
4.5	Total Unadjusted Function Points (UFP)	15
5	REST Interface	16
5.1	API Tags	16
5.2	Security Schemes	16
5.3	API Components	17
5.4	API Endpoints	17
5.5	API Example	20
6	System Architecture	21
6.1	Microservices	21
6.2	Integration with External Services	22
6.3	Messaging and Communication	22
6.4	Data Storage and Persistence	22
6.5	Web Socket	23
6.6	Security	23

1 Introduction

This report provides an overview of the PlaceAdvisor web application, including the design of the system, software architecture, and the project's development process.

PlaceAdvisor is a platform designed to help users discover points of interest in various cities and locations. After obtaining all the relevant information about a point of interest (address, map, historical background, and more), users can also view reviews from other users and, if desired, contribute their own reviews.

The report is structured as follows:

1. **User Requirements:** This section outlines the requirements gathered from users and stakeholders to develop the PlaceAdvisor application.

2. **User Stories:** Here, you will find detailed user stories and mock-up that illustrate the expected user interactions and functionalities of the application.

3. **Function Points:** This section provides an explanation of how function points were calculated to estimate the software size.

4. **Rest Interface:** Here we have the RESTful interface developed to provide access to the application's functionalities.

5. **System Architecture:** Here we will provide an overview of the architecture of PlaceAdvisor, including the microservices, data storage, and external service integrations.

2 User Requirements

2.1 Introduction

In this section, we will introduce the user requirements, that is the needs, expectations, and preferences of the people who will use a system (users). They are essential for designing and developing a system that meets the goals and solves the problems of the users. Let's start.

2.2 User Authentication and Registration

Users should be able to authenticate using their Facebook account.

PlaceAdvisor acknowledges the significance of user convenience and the importance of utilizing established, secure identity providers. By allowing users to authenticate using their Facebook accounts, we ensure a seamless and secure login experience, eliminating the need to create a new set of credentials for our platform.

2.3 Integration with Google Photos

OAuth flow for read-only access.

Users can access their Google Photos in a read-only mode, enabling them to seamlessly integrate their personal media collections into their PlaceAdvisor experience. This integration simplifies the process of sharing images and memories of their visits to points of interest.

2.4 User Profiles

User's activity, basic info, and submitted feedback/reviews.

User profiles serve as a central hub for personalization and user engagement. In this context, PlaceAdvisor collects basic user information and showcases their activity, highlighting their interactions with points of interest. Users also have the ability to submit feedback and reviews, fostering a sense of community and engagement.

2.5 Points of Interest Discovery

Easy search by city, category, and radius.

The core feature of PlaceAdvisor, point of interest discovery, is designed to be intuitive and efficient. Users can seamlessly search for points of interest by specifying key criteria such as city, category, and radius, ensuring that they can narrow down their choices to match their preferences effortlessly.

Results displayed in a user-friendly list with detailed options.

The search results are presented in a user-friendly list format, providing users with an

overview of the available options. In addition, detailed information is readily accessible, ensuring that users can make informed decisions about their chosen points of interest.

2.6 Detailed Place Information

Dedicated page with images, Wikipedia description, address, and weather.

For each point of interest, PlaceAdvisor offers a dedicated page filled with a wealth of information. This includes vivid images, Wikipedia descriptions, precise addresses, and real-time weather data. This rich, detailed content empowers users with comprehensive insights into their selected destinations.

Allows user-generated reviews with text and optional photos.

The interactive aspect of PlaceAdvisor is enhanced through user-generated reviews. Users can share their experiences and insights, providing invaluable information to the community. The ability to include optional photos adds depth and authenticity to their contributions.

2.7 Chat with Support

Users should be able to initiate a chat with support in real-time.

In acknowledging the importance of immediate assistance, PlaceAdvisor offers real-time chat support. This feature ensures that users can reach out for help or information whenever they require it.

Users should receive timely responses from customer support.

Timeliness is paramount in customer support. PlaceAdvisor's commitment to user satisfaction is exemplified by our dedication to providing quick and effective responses from our support team.

2.8 User Feedback Feature

Convenient form for insights and suggestions.

To foster an open and responsive relationship with our users, PlaceAdvisor offers a user feedback feature. This convenient form allows users to share their insights, suggestions, and concerns, helping us to continuously enhance our platform.

Users should be able to track submission status on personal info page.

Users' engagement doesn't end with feedback submission. PlaceAdvisor enables users to monitor the status of their submissions directly from their personal information page, promoting transparency and accountability.

2.9 Kibana and Elasticsearch Dashboard

Users should have access to a Kibana and Elasticsearch-based dashboard.

PlaceAdvisor is dedicated to providing users with insights into their own interactions

and preferences. The Kibana and Elasticsearch-based dashboard is designed to offer a powerful tool for users to gain deeper understanding, make data-driven decisions, and tailor their experiences on the platform.

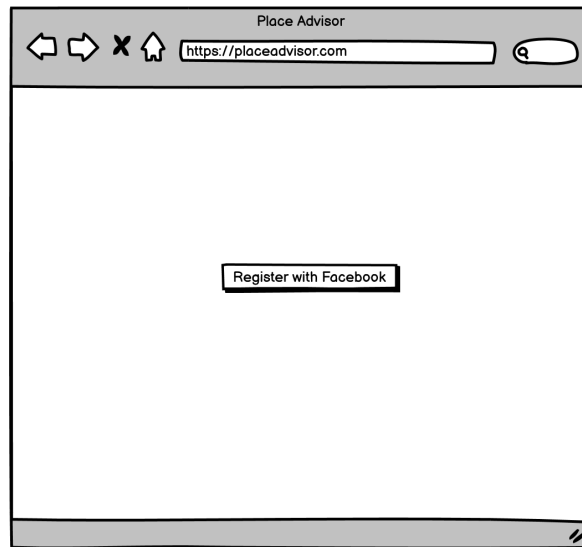
In the following sections of this report, we will delve deeper into the product backlog, where these user requirements are translated into actionable user stories, and discuss the roadmap for the development of PlaceAdvisor. Additionally, we will conclude with a demonstration of the core features to illustrate how the user requirements manifest in the live application.

3 User Stories

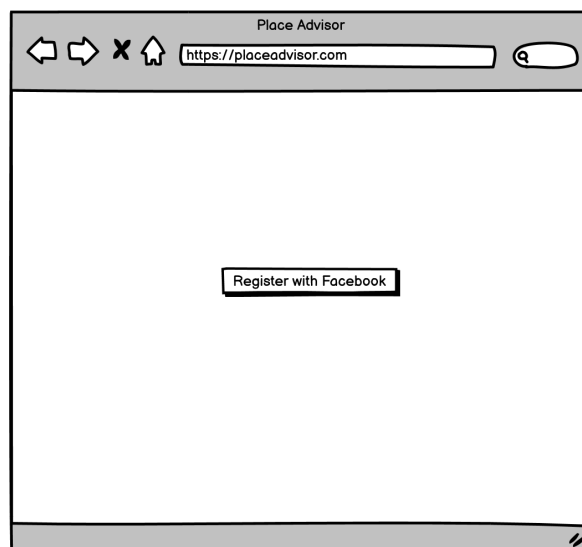
In this section, we will break down the user requirements into user stories, providing also some mockup crafted with balsamiq. Each user story represents a specific piece of functionality that adds value to our web-application, providing a clear, concise description of what needs to be implemented to better guide the development process.

User Registration:

- As a new user, I want to register inside the web application using my Facebook account in order to access the service.

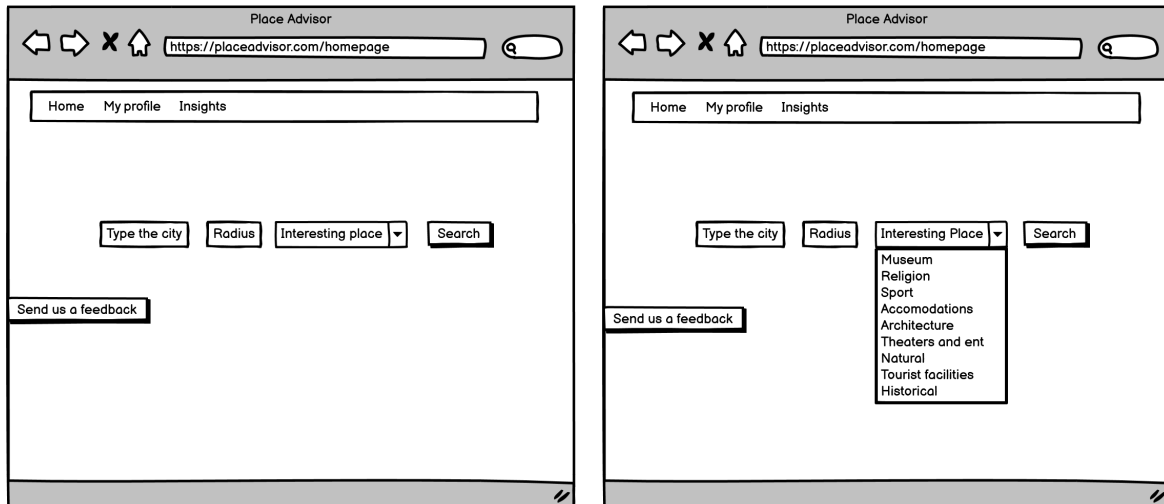


- As a registered user, I want to be able to log in with my Facebook credentials to save time and avoid creating a new account.



Points of Interest Discovery:

- As a user, I want to search for points of interest in a specific area or city to discover new places to visit.



- As a user, I want to view detailed information about a point of interest, including its address, historical facts, and other relevant details, in order to make informed decisions about visiting it.
- As a user, I want to read reviews and ratings of points of interest from other users to get insights into their experiences.
- As a user, I want to leave my own reviews and ratings for points of interest to share my opinions and help others make informed choices.

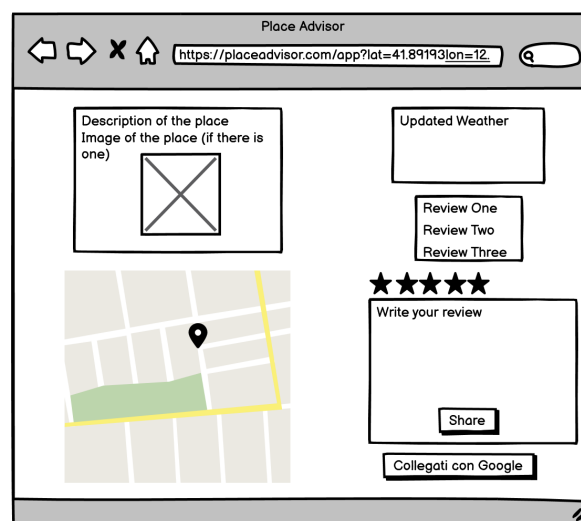
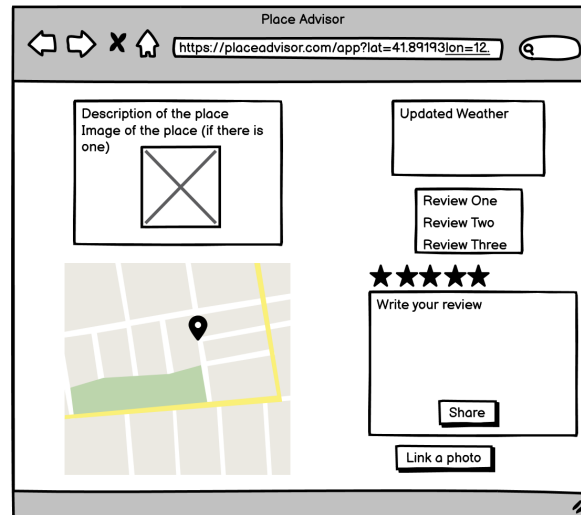
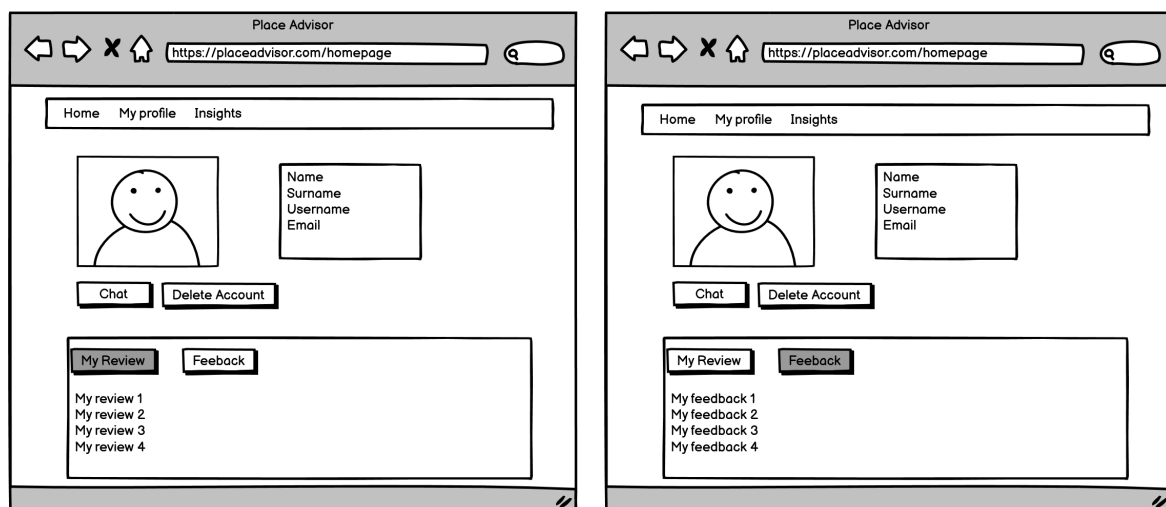


Photo Integration and User Profiles:

- As a user, I want to login with Google and upload and attach photos to points of interest to visually share my experiences with others.



- As a user, I want to access my user profile page to see all my details and view the list of the feedbacks and reviews I have submitted.

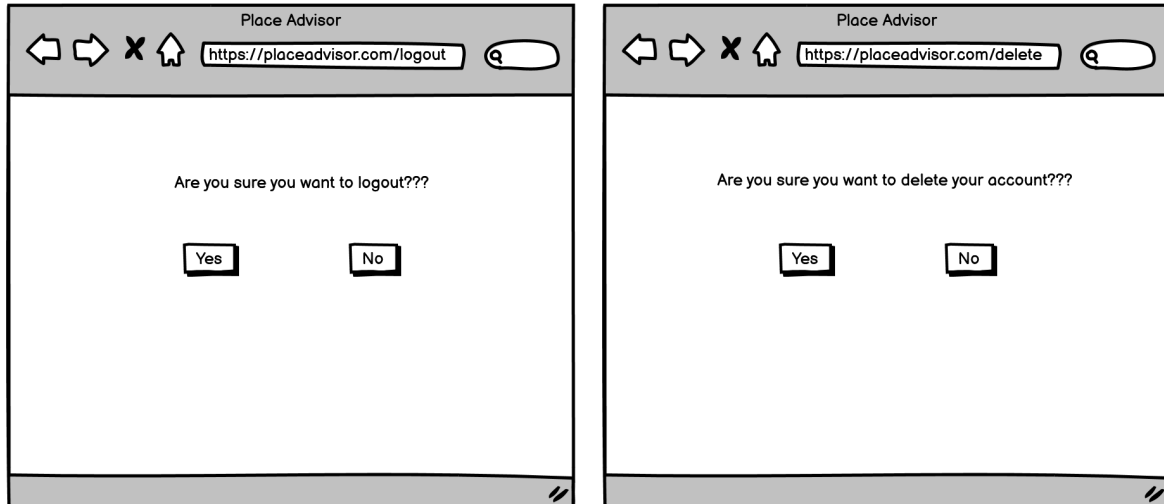


User Account Management:

- As a registered user, I want to have the ability to log out of my PlaceAdvisor account and delete my account to ensure the security and privacy of my data and to have control over my account status.

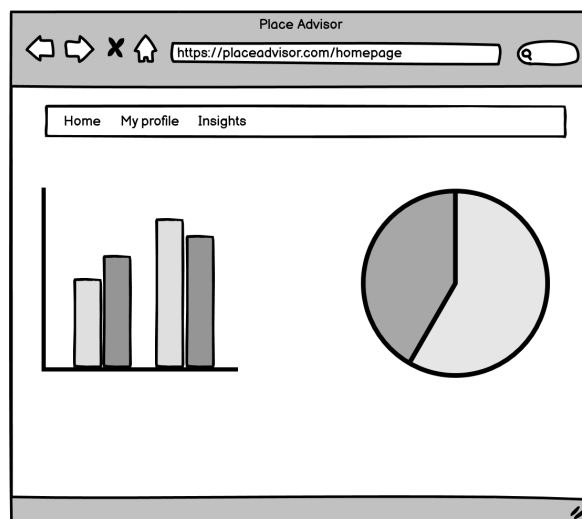
User Feedback and Suggestions:

- As a user, I want to provide feedback or suggestions about the app's features and functionality to improve the quality of service.



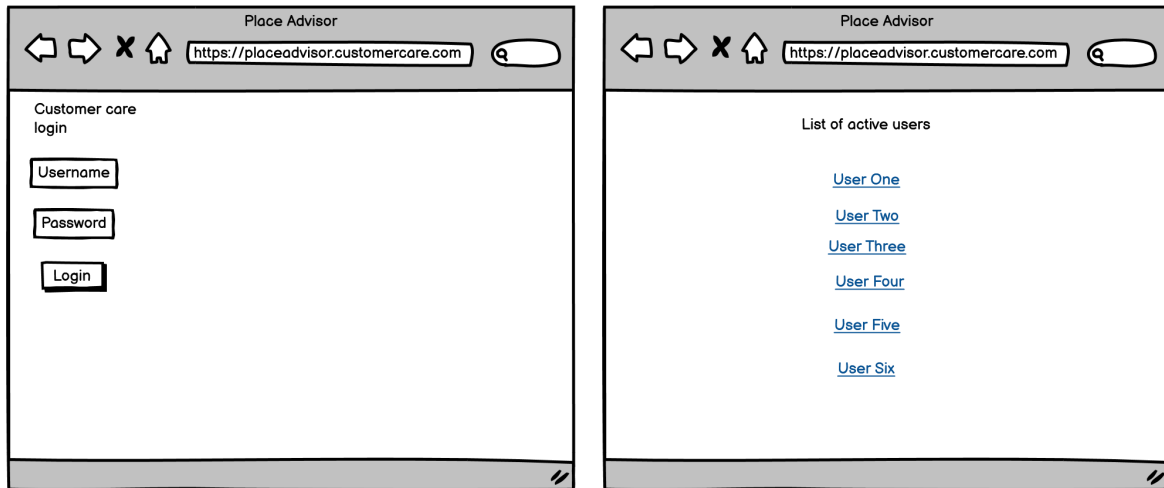
Analytics and Insights:

- As a user, I want access to analytics and insights about points of interest to gain a better understanding of popular destinations and trends.

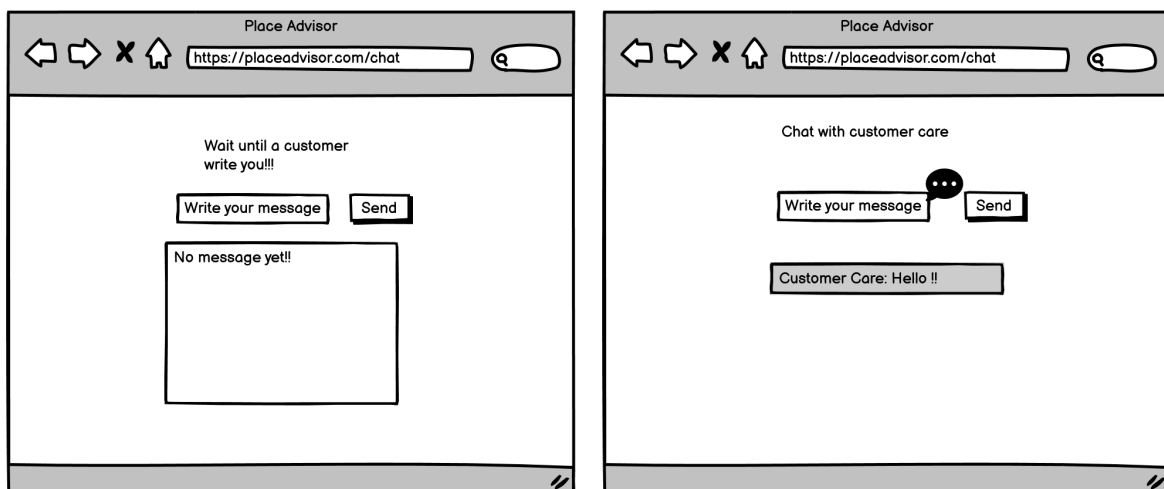


Customer Support:

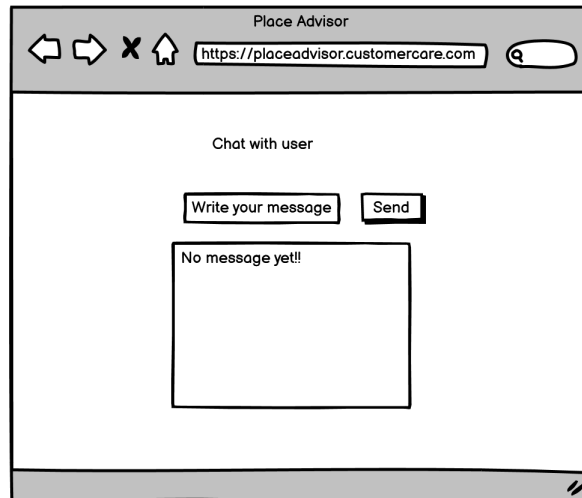
- As a Customer Care representative, I want to log in inside the web application to find if there are some users that need help.



- As a user, I want to initiate a real-time chat with support representatives to get assistance with app-related issues.



- As Customer Care, I want to chat with users of the web application to assist the users and respond to their questions.



API Documentation and Security:

- As a User/Developer, I want to access API documentation to understand how to use and integrate with the PlaceAdvisor API effectively.
- As a user, I want my authentication session to persist securely to have a seamless experience and not re-login frequently.
- As a security-conscious user, I want to access PlaceAdvisor via a secure HTTPS connection to protect my data and ensure secure communication.

Infrastructure and Aesthetics:

- As a developer, I want to integrate Docker Compose to simplify the deployment and scaling of our application components and ensure consistent development and production environments.
- As a user, I want a dashboard with Kibana and Elasticsearch for insights into user and POI data to make data-driven decisions and gain valuable insights.
- As a user, I want the pages of the PlaceAdvisor website to have an improved aesthetic design to have a visually pleasing and engaging experience while using the platform.

4 Function Points

Function Points (FP) serve as a valuable metric for measuring the functionality and size of the PlaceAdvisor web application. They provide insights into the complexity and scope of the system, aiding in project estimation and resource allocation.

4.1 Complexity Tables for Function Points

In this section, we present complexity tables for Function Points in the context of PlaceAdvisor. These tables define the relationships between data element types (DET) and data element records (RET) for Internal Logical Files (ILF) and External Interface Files (EIF), as well as the complexity of data functions associated with External Inputs (EI), External Outputs (EO), and External Inquiries (EQ).

RET/DET	1-19 DET	20-50 DET	51+ DET
0-1 RET	Low 6/4	Low 6/4	Med. 9/7
2-5 RET	Low 6/4	Med. 10/7	High 15/10
6+ RET	Med. 10/7	High 15/10	High 15/10

Table 1: Complexity Table for ILF/EIF

FTR/DET	1-4 DET	5-15 DET	16+ DET
0-1 FTR	Low 1	Low 2	Med. 3
2 FTR	Low 3	Med. 4	High 5
3+ FTR	Med. 5	High 6	High 7

Table 2: Complexity Table for EI

FTR/DET	1-4 DET	5-15 DET	16+ DET
0-1 FTR	Low 2/1	Low 3/2	Med. 4/3
2 FTR	Low 4/3	Med. 5/4	High 6/5
3+ FTR	Med. 5	High 6	High 7

Table 3: Complexity Table for EO/EQ

4.2 Internal Logical Files (ILF)

ILF represents data maintained within the application. Here are the ILFs identified for PlaceAdvisor:

- **User**
 - DET (Data Element Type): 7
 - RET (Record Element Type): 2 (reviews, feedbacks)
- **Review**
 - DET (Data Element Type): 8
 - RET (Record Element Type): 1 (reviews)
- **City**
 - DET (Data Element Type): 3
 - RET (Record Element Type): 0

4.3 External Interface Files (EIF)

EIF represents data shared with external applications or services. Here are the EIFs identified for PlaceAdvisor:

- **Place**
 - DET (Data Element Type): 25
 - RET (Record Element Type): 2
- **Facebook Profile**
 - DET (Data Element Type): 4
 - RET (Record Element Type): 0

4.4 Calculation of Function Points

We calculate function points based on the user stories and requirements previously outlined in this report. Each user story is mapped to one or more function point categories, and a complexity factor (Low, Average, or High) is assigned based on the complexity of the story.

The function points are calculated as follows:

User Registration and Authentication:

- Login: 1 EQ * Low Complexity = 1 FP

- User Registration with Facebook Login: 1 EI * Low Complexity = 2 FPs

Homepage:

- Search for places: 1 EQ * Low Complexity = 1 FPs
- Profile information: 1 EQ * Medium Complexity = 5 FPs
- Kibana dashboard: 1 EQ * High Complexity = 6 FPs
- Login with Google navigation link: 1 EQ4 * Low Complexity = 1 FPs

List POI after search:

- Place (for each POI): 1 EQ * Low Complexity = 1 FPs

POI Details Page:

- Google Profile (login with Google): 1 EQ * Low Complexity = 1 FPs
- Review, User (review and photo): 1 EI * Medium Complexity = 4 FPs

User Info Page:

- User, Reviews, Cities (Kibana dashboard): 1 EQ * High Complexity = 6 FPs
- User (see reviews link): 1 EQ * Low Complexity = 1 FPs
- User (see feedback link): 1 EQ * Low Complexity = 1 FPs
- Reviews, User (delete review): 1 EO * Medium Complexity = 3 FPs
- Reviews, User (delete account): 1 EO * Medium Complexity = 5 FPs

Kibana Dashboard Insight Page:

- User (user profile navigation link): 1 EQ * Low Complexity = 1 FPs

User Chat:

- User: 1 EQ * Low Complexity = 1 FP

Support:

- Login: 1 EI * Low Complexity = 1 FPs
- User: 1 EQ * Low Complexity = 1 FPs

4.5 Total Unadjusted Function Points (UFP)

The Unadjusted Function Points (UFP) is the sum of the calculated function points in each category:

- EIs: 7 FPs
- EOs: 8 FPs
- EQs: 27 FPs
- ILFs: 18 FPs
- EIFs: 8 FPs

Total UFP: 68 FPs

5 REST Interface

The PlaceAdvisor web application provides a RESTful API that allows users to interact with the platform programmatically. Below, we outline the available resources, their descriptions, supported HTTP methods, and required authorization.

5.1 API Tags

The API is organized into several tags, each representing a specific area or function of the application:

- **Root:** Start and general information.
- **User:** Obtain user information.
- **Home:** Manage user access and access the primary APIs.
- **APIs:** Access various APIs.
- **Reviews:** Manage user reviews.
- **Feedback:** Manage user feedback.
- **Error:** Handling errors and providing feedback.
- **RefreshToken:** Token refresh.

5.2 Security Schemes

The PlaceAdvisor API employs several security schemes to ensure secure access. These schemes include:

- **googleOAuth:** OAuth2 authentication for Google.
- **facebookOAuth:** OAuth2 authentication for Facebook.
- **fbcookieAuth:** Authentication using Facebook access tokens stored in cookies.
- **GoogleAccessToken:** Authentication using Google access tokens stored in cookies.
- **GoogleIdToken:** Authentication using Google ID tokens stored in cookies.
- **JWT:** JSON Web Token (JWT) for authentication.
- **JWT_refresh:** JWT for token refresh.

5.3 API Components

The PlaceAdvisor API defines several data schemas to structure the data sent and received through the API. These schemas include:

- **Review:** Information about user reviews.
- **Feedback:** User feedback details.
- **User:** User profile information, including reviews and feedback.
- **Ricerca:** Parameters for search queries.

5.4 API Endpoints

The PlaceAdvisor API is organized into various endpoints, each serving a specific purpose. Here are some of the key endpoints:

Root Resource (/)

- **GET:** Returns the index page (index.ejs).
- **POST:** Allows users to authenticate with Facebook.
- **Authorization:** None for GET; Facebook OAuth for POST.

Home Resource (/home)

- **GET:** Returns the homepage (HTML HOMEPAGE).
- **Authorization:** JWT token required.
- **Responses:** 200 OK, 403 Forbidden (User not authenticated).

User Resource (/info)

- **GET:** Returns user information, reviews, and feedback (HTML user_info).
- **Authorization:** JWT token required.
- **Responses:** 200 OK, 403 Forbidden (User not authenticated).

City Info Resource (/city_info)

- **GET:** Returns a list of popular places in the site (city_stat.ejs).
- **Authorization:** JWT token required.
- **Responses:** 200 OK.

API Resource (/app)

- **GET:** Returns a list of places within a certain location, category, and radius.
- **Parameters:** lat (latitude), lon (longitude), cate (category), rad (radius).
- **Authorization:** JWT token required.
- **Responses:** 200 OK, 404 Not Found.

OpenMap Resource (/openmap)

- **POST:** Executes a search for places of interest based on city, radius, and category.
- **Authorization:** JWT token required.

Details Resource (/details)

- **GET:** Returns details about a specific place.
- **Parameters:** xid (XID of the place), baseUrl (optional photo URL).
- **Authorization:** JWT token required.
- **Responses:** 200 OK, 404 Not Found.

Google Photos API Resource (/googlephotosapi)

- **GET:** Retrieves the user's Google Maps photos to upload to PlaceAdvisor.
- **Parameters:** stato (to specify if the photo should be used for feedback) and xid (XID of the place to post the photo).
- **Authorization:** JWT token required, Google OAuth cookie.
- **Responses:** 200 OK, 404 Not Found.

Logout Resource (/logout)

- **GET:** Returns the logout page.
- **POST:** Logs out the user.
- **Authorization:** JWT token required.
- **Responses:** 200 OK (GET and POST).

Reviews Resource (/reviews)

- **POST:** Posts a review.
- **Authorization:** JWT token required.
- **Responses:** 200 OK (creates or updates a review).

Elimina Resource (/elimina)

- **POST:** Deletes a review by code.
- **Authorization:** JWT token required.
- **Responses:** 200 OK (deletes a review), 404 Not Found.

New Feedback Resource (/newfeedback)

- **GET:** Displays the feedback submission form.
- **Parameters:** baseUrl (optional photo URL).
- **Authorization:** JWT token required.
- **Responses:** 200 OK (displays the form), 403 Forbidden (token expired), 404 Not Found.

Feedback Resource (/feedback)

- **POST:** Submits user feedback.
- **Authorization:** JWT token required.
- **Responses:** 200 OK (submits feedback).

Refresh Token Resource (/refreshtoken)

- **GET:** Refreshes the JWT token.
- **Authorization:** JWT token and refresh token required.

Error Resource (/error)

- **GET:** Displays an error page with a status code explanation.
- **Parameters:** statusCode (error code).
- **Responses:** 200 OK (displays the error page).

These endpoints facilitate various actions, including user authentication, data retrieval, and interaction with reviews and feedback.

5.5 API Example

Below is an example of an API request:

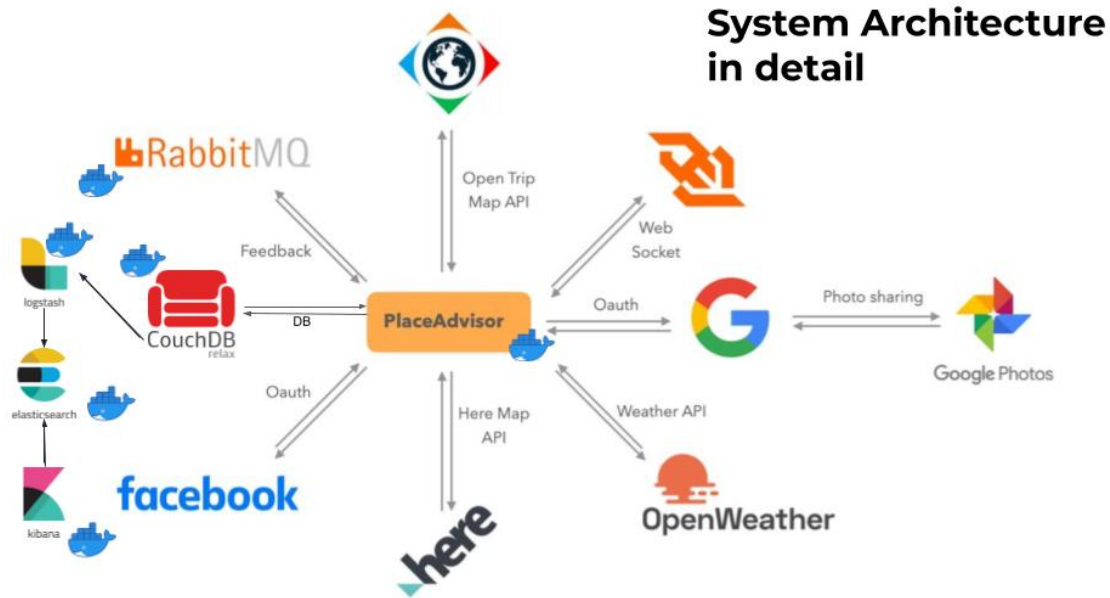
```
1 GET /info
2 Headers:
3   Authorization: Bearer <JWT Token>
```

In this example, the user requests their information using a JWT token for authentication.

The PlaceAdvisor API documentation serves as a comprehensive guide for interacting with the platform's capabilities, empowering users and developers to make the most of the application's features.

6 System Architecture

The system architecture of PlaceAdvisor is designed to be a robust and scalable solution for providing information on points of interest and facilitating user interactions. The architecture is based on microservices, with each component responsible for specific tasks and functionalities. Below is an overview of the key architectural components and technologies used:



6.1 Microservices

PlaceAdvisor employs a microservices architecture, which allows for modularity and flexibility in scaling individual components. The following microservices make up the system:

1. **Node.js Service:** This microservice is responsible for handling user registration, authentication, and user-related operations. It communicates with the CouchDB database to store user data and interacts with external services for authentication.
2. **CouchDB:** A NoSQL database used for storing user information, city data, and user reviews. It provides a scalable and reliable data storage solution for the application.
3. **RabbitMQ:** A message broker used for handling asynchronous tasks and feedback processing. It ensures reliable communication between components and allows for scaling services independently.

4. **Elasticsearch and Kibana:** Elasticsearch serves as the search and analytics engine for user reviews and places of interest. Kibana provides a user-friendly interface for data visualization and exploration.
5. **Logstash:** Logstash is responsible for collecting and processing data from couchdb and to use them in elasticsearch.
6. **Support Service:** This microservice handles user feedback and support-related operations.

6.2 Integration with External Services

PlaceAdvisor integrates with several external services to enhance its functionality and user experience. These services include:

1. **OpenTripMap API:** Used to obtain information about places of interest, including their details, locations, and categories.
2. **OpenWeatherMap API:** Provides real-time weather data for locations, enhancing the user's experience with weather information.
3. **HERE API:** Enables the identification of places on a map, allowing users to visualize the location of points of interest.
4. **Google Photos and OAuth:** Users can upload photos in their reviews. OAuth is used for secure and authorized access to Google Photos.
5. **Facebook OAuth:** Allows users to log in to PlaceAdvisor using their Facebook credentials and share posts if desired.

6.3 Messaging and Communication

To ensure efficient communication and data exchange between components for the feedback service, PlaceAdvisor relies on the RabbitMQ message broker. This allows for asynchronous processing of tasks such as feedback handling, ensuring system responsiveness and scalability.

6.4 Data Storage and Persistence

CouchDB is used as the primary database for storing user data, city information, and user reviews. It provides the required flexibility and scalability for a system that handles user-generated content.

6.5 Web Socket

Web Sockets are used to implement a chat between the user and the support. The communication happens in real-time and it is realized with the purpose of offering a service to the user which can increase the user-friendliness of the website.

6.6 Security

The system ensures secure communication using OpenSSL for HTTPS connections, protecting user data and interactions.