Advanced Software Engineering

Course Project – RESTful API's – Fall 2024

Dr. Amjad AbuHassan

RentItOut: Peer-to-Peer Rental Platform for Everyday Items

Application requirements

- I do not want you to be forced to use any programming language or technology. What is most important is that you always motivate your choices. You will have many occasions to do so, make use of the documentation and presentations.

Expose an API

- The backend exposes a dataset in the form of an API. One option is to use a **REST API**.
    - You can also consider other modern API methods, for instance **GraphQL**.

Project requirements

In the case of the project development process, you have a bit more requirements to follow

Project planning and version control

- Track your interaction activities for each of the phases of the project development. For this reason, you must use a *git* repository.
    - All the members of the team can work independently and merge the code every time a conflict occurs.
    - You are required to use at least a `main` branch and many development branches you need. One suggestion is to use the [git workflow] branching model, but you are not forced to use it. Instead, you are asked to use [pull requests] every time you want to merge the code with the `main` branch, to discuss and review potential changes with your collaborators.

Documentation (Wiki)

- I need a documentation of the whole overview, since you should have organized everything in *api's*. Anything useful to better understand your code will be appreciated. Also, mention the use of specific technologies and their motivation.
- You have then to maintain a wiki for the project. Every GitHub project has a parallel repository where you can store *Markdown* files to compose the wiki. I am expecting you

to describe your project, its architecture, and how to use it. In particular, I want to understand the motivation for all of your choices.

- You can use the wiki also to document the usage for the API of the backend.
    - As an alternative, you can use specific API documentation tools, such as [Swagger].
- The testing was part of you job. Report everything useful to describe the way you planned and performed the testing of your code.

Application Description:

In this project, you will develop a backend API for **RentItOut**, a platform where users can rent out everyday items they own but don't use frequently, such as tools, sports equipment, electronics, party supplies, and more. The goal is to create a circular economy that encourages sharing and reduces the need for people to purchase items they only need occasionally. This project will focus on creating a robust backend, you can use one of these frameworks (Spring Boot, Node JS, Django).

Core Features:

1. **Item Listings for Rent:** Users can list infrequently used items, such as tools, gear, electronics, and vehicles, across various categories like home improvement and transportation.
2. **Rental Management and Pricing:** How will the platform manage rental durations, pricing models, and flexible periods?
3. **Trust, Safety, and Verification:** Consider how user verification, reviews, and ratings can be used to foster safety and reliability in transactions. What are potential ways to ensure items are returned in good condition (e.g., security deposits, damage protection)?.
4. **Logistics: Delivery and Pickup:** Brainstorm logistics options, including in-person pickups or delivery services. Consider how a location-based system (like maps) could help facilitate item exchanges.
5. **Revenue Model and Insurance:** How will the platform generate income, and how can insurance or damage protection be integrated?
6. **User Experience and Recommendations:** How can the platform enhance the user experience?

External API Integration:

In addition to the main features, Integrate with external APIs that provide data from authoritative sources, (this enhances the platform's data accuracy and comprehensiveness).

Mandatory Features:

- CRUD Operations: The API must implement basic CRUD operations on the MySQL database to enable the creation, retrieval, update, and deletion of application data applications.
- API Documentation: The API must be fully documented using tools such as Swagger or Postman to enable other developers to understand how to use the API.

Other features

- **User Privacy and Data Security:** Implement stringent privacy and security measures to protect users' data.
- **Roles**
- **Error Handling and Logging:** The API must implement error handling and logging to help with debugging and troubleshooting.

Important Notes:

For the API project, there are some mandatory features that you must include in your API. These features will ensure that your API is functional. It is your responsibility to analyze the project requirements and identify the mandatory features. The details of these features will depend on your analysis and imagination. In addition to the mandatory features, you have the flexibility to express your project ideas and add additional features to the API. We encourage you to think creatively and add features that would make your API stand out from others.

You can also think about how your API might integrate with other applications or platforms to create a more comprehensive platform experience. Overall, this project offers a range of backend API development challenges and can be customized to fit your specific needs and interests of while providing them with the opportunity to learn how to integrate external APIs to enhance your platform.