**QuickByte Food Delivery App User Manual**

**\*\*By:\*\***

* ***Sergiu Medrea***
* ***Mircea Răuțoiu***

**Table of Contents**

1. Introduction
2. System Requirements
3. Installation
4. Starting the Application
5. Main Menu
6. Enitities management
7. Error Handling
8. Support and Contact Information

**1. Introduction**

Welcome to QuickByte, an innovative and efficient food delivery application developed using the Spring Framework. QuickByte is designed to streamline your food ordering experience, offering a range of features and functionalities to make ordering, managing, and delivering food a breeze.

**Developed by:**

* Mircea Răuțoiu
* Sergiu Medrea

In this user manual, you will find comprehensive guidance on how to navigate through the application's diverse features. QuickByte not only ensures a seamless food delivery process but also incorporates advanced design patterns and frameworks to enhance reliability and performance.

If you have any questions, require assistance, or want to share your feedback, feel free to reach out to our dedicated support team.

Enjoy the convenience of QuickByte as you explore the various functionalities tailored to make your food delivery experience exceptional.

**2. System Requirements**

Before using QuickByte, ensure that your system meets the following requirements:

* Java Runtime Environment (JRE) 8 or higher
* PostgreSQL Database
* Internet connection for database connectivity

**3. Installation**

1. Clone the QuickByte repository from GitHub.
2. Set up the PostgreSQL database and configure the connection details in the application properties.
3. Build the project using your preferred IDE or build tool.

**4. Starting the Application**

Run the application using the provided executable JAR file or by executing the main class. The application will start, and you'll be presented with the main menu.

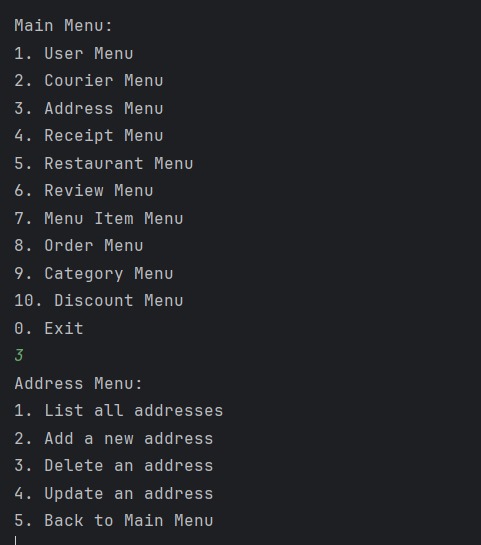
**5. Main Menu**

The main menu provides access to various functionalities(**CRUD** operations) for every entity individually. Choose an option by entering the corresponding number.



**Example for Address Management:**

* **List all addresses (Option 1):** View all existing addresses.
* **Add a new address (Option 2):** Add a new address to the system.
* **Delete an address (Option 3):** Remove an existing address.
* **Update an address (Option 4):** Modify details of an existing address.



**6. Entities Management**

The Entities Management functionality in QuickByte allows you to interact with entitiy-related operations through the RESTful API. Below is a example with user related available operations:

**6.1 List All Users**

* **API Endpoint:** **/api/v1/user**
* **Description:** Retrieve a list of all users in the system.
* **How to Access:** Select option 1 in the User Menu.

**6.2 Add a New User**

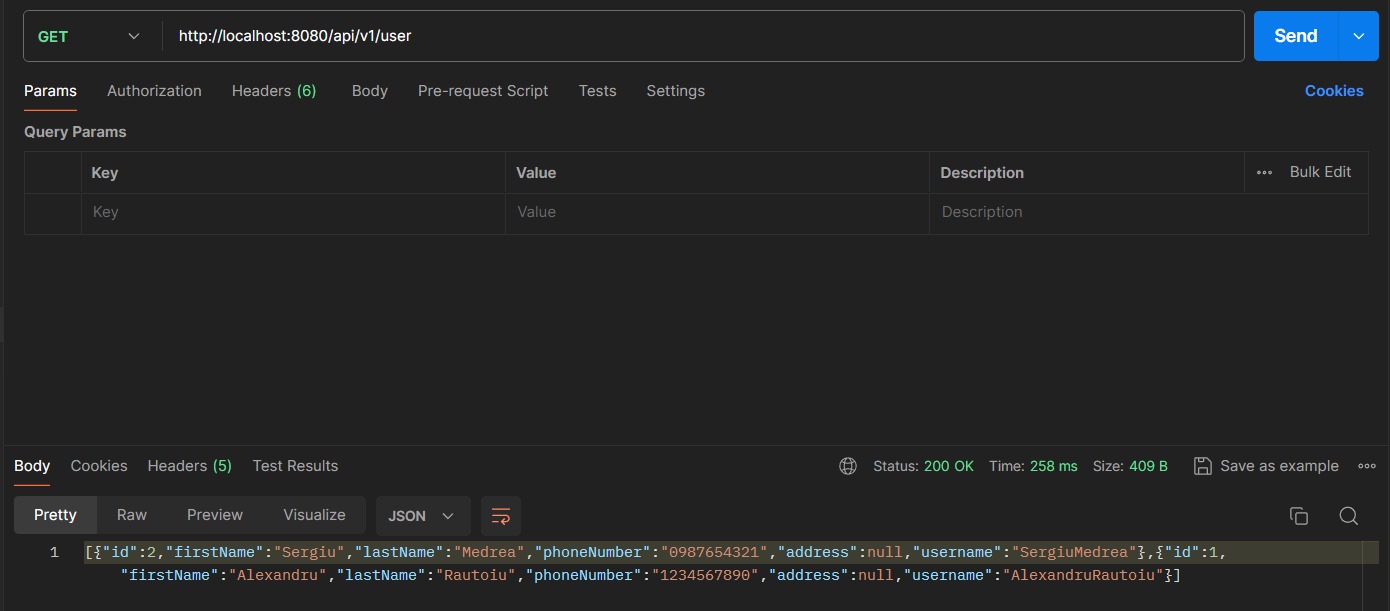
* **API Endpoint:** **/api/v1/user**
* **Description:** Add a new user to the system.
* **How to Access:** Select option 2 in the User Menu and follow the prompts.

**6.3 Delete a User**

* **API Endpoint:** **/api/v1/user/{userId}**
* **Description:** Delete an existing user from the system.
* **How to Access:** Select option 3 in the User Menu and follow the prompts.

**6.4 Update a User**

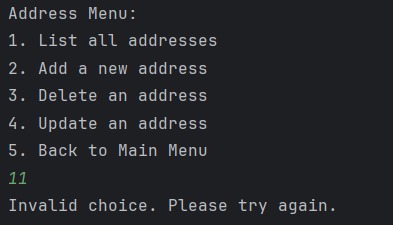
* **API Endpoint:** **/api/v1/user/{userId}**
* **Description:** Update details of an existing user in the system.
* **How to Access:** Select option 4 in the User Menu and follow the prompts.

These API endpoints correspond to the operations available in the QuickByte User Management section. Feel free to use tools like cURL or Postman to interact with the API directly or integrate it into your own applications.

This addition provides users with information on how the functionalities they interact with in the user interface correspond to the underlying API endpoints for user management. It helps users understand the connection between the actions they perform in the application and the backend API operations.

**7. Error Handling**

The application is designed to handle errors gracefully. If you encounter any issues, follow the on-screen instructions. Common errors include entering invalid input or attempting to perform an action on a non-existent entity.



**8. Support and Contact Information**

If you need assistance or have any questions, please contact our support team at support@quickbyte.com. We appreciate your feedback and are dedicated to improving the QuickByte experience.

*Thank you for choosing QuickByte for your food delivery needs. Enjoy a seamless and efficient experience with our application!*