Design Choices:

1. Use a RESTful API architecture for the microservice, which allows for easy communication between the client and the server.

2. Use a database to store user information and loan data. I'm using Sqlite3 in this case because it's a development environment.

3. Use JWT (JSON Web Token) authentication to secure the API endpoints.

4. Use HTTPS protocol to ensure secure communication.

5. Use Django to simplify the development process and make the code more readable.

Assumptions:

1. Assume that the lending platform allows users to apply for loans, view loan details, and make payments.

2. Assume that users will provide personal information, such as full name and phone number to apply for a loan.

3. Assume that the lending platform will require users to have a mobile wallet to make payments.

4. Assume that the lending platform will need to comply with relevant regulations and laws, such as KYC (Know Your Customer) and AML (Anti-Money Laundering) regulations.

Pitfalls and Improvements:

1. Pitfall: The lending platform may experience scalability issues as the number of users and loan applications increase.
   Improvement: Use a cloud-based hosting service like AWS or Azure, which can scale automatically to handle increased traffic.

2. Pitfall: The lending platform may face security risks, such as data breaches and unauthorized access.

   Improvement: Implement security measures like encryption, two-factor authentication, and regular security audits to mitigate the risks.

3. Pitfall: The lending platform may face compliance issues if it fails to meet regulatory requirements.

Improvement: Stay up-to-date with the latest regulations and laws and implement appropriate measures to ensure compliance.

Test Cases:

1. User creates an account and logs in.

2. Admin creates loan products.

3. Customer creates and updates profile which includes full name and phone number.

4. Customer creates a mobile wallet.

5. Customer applies for a loan.
   Expected outcome: User's loan application is submitted successfully.

6. User's application is rejected due to incomplete or incorrect information.
   Expected outcome: User is notified of the rejection and provided with instructions on how to correct their information.

7. User views their loan details.
   Expected outcome: User can view their loan details, including loan amount, interest rate, and repayment schedule.

8. User makes a payment.
   Expected outcome: User's payment is processed successfully, and their loan balance is updated accordingly.