Staff Engineering - Level 6 | AstroPay

Objective:

The goal of this test is to assess the candidate's ability to understand, design and implement a technical solution to a problem that is representative to the reality the candidate will be faced with on AstroPay.

Problem:

As AstroPay expanded, we now have a lot of services, each one of them with a somewhat clear scope and responsibilities. We have one for deposits (top-ups), one for users, one for card payments, etc. While this is a robust architecture, we are now facing the emerging challenge of aggregating and organizing user activities from different sources.

This assignment's goal is to come up with a strategy, and design and implement a robust and scalable solution for handling user activities, including the data layer and structures that can support hundreds of thousands of users.

Expected Outcome:

The candidate must create an endpoint or set of endpoints that can be plugged to a frontend allowing a user to browse his activities and potentially support search, sort and filter in an efficient manner. The solution must specify any changes or additional requirements to other existing services.

For the purpose of this assignment, we will consider the following activity types:

- Card payment
- Deposit (top-up)
- P2P Transfer

Sample data for these transactions can be found in the Appendix A.

What we are looking for is the following:

- A good solution covering many edge cases or stating clearly detected ones that are not covered due to lack of time, along with comments about approach to handling them.
- Understanding of the inherent challenges of the problem and alignment of the solution to them.
- Components that are well separated and potentially reusable.
- Abstractions are clear and provide the possibility of extension in the future without significant refactoring.
- Details like idempotency and transactionality are considered and properly handled if relevant to the problem.

Technical requirements:

The API must be implemented in Java 11/17 using any version of Spring Boot. Libraries of your choice are permitted. Any additional technology choice must be clearly stated and justified.

Submission Guidelines:

- Create a Git repository for the project and share the repository URL.
- Include a README.md file with instructions on how to run services and test them.
- Create a dedicated folder within the repository to house diagrams and necessary documentation.

Appendix A

Here you have one example of each activity structure:

Card Payment	`{`
	`"payment_id": "8c051707aadd416d8e7e094971e395c0",`
	`"card_id": 12341234,`
	`"user_id": 113411,`
	`"payment_amount": 10,`
	`"payment_currency": "EUR",`
	`"status": "COMPLETED",`
	`"created_at": "2023-12-14 13:37:31",`
	`"merchant_name": "TFL*LONDON",`
	`"merchant_id": 12309,`
	`"mcc_code": 7399`
	`}`

Deposit (top-up)	`{`
	`"deposit_id": "071d500bf9d74c72a963d77d2b9a0107",`
	`"user_id": 113411,`
	`"deposit_amount": 12.99,`
	`"deposit_currency": "GBP",`
	`"status": "PENDING",`
	`"created_at": "2023-12-02 13:54:30",`
	`"expires_at": "2023-12-04 13:54:30",`
	`"payment_method_code": "BANK_TRANSFER",`
	`}`

AstroPay

P2P Transfer	{ "transfer_id": "7bec23e832ee41bfa0d59497ffccc553", "sender_id":"113411", "recipient_id":"113412", "transfer_amount": 15, "transfer_currency": "EUR", "status":"COMPLETED", "comment":"Dinner party at Richard's", "created_at": "2023-12-11 13:38:16" }