# Overview

Thank you for the interest in betPawa.

As agreed, I am sending you the test task which will assess your readiness to work with our technology stack.

All technologies are used on a daily basis within our organization and this will prepare you for everyday software development participation.

# Requirements

The task consists of a wallet server and a wallet client.   
The wallet server will keep track of a users monetary balance in the system.   
There should be separate balance for different currencies.  
The client will emulate users depositing and withdrawing funds.

# Database

Must include the database schema that is needed for the server application.

# Technologies

The following technologies **must** be used

* Java 8+
* Spring 5.x
* gRPC
* MySQL 5.x
* Gradle
* JUnit

Extra points for using these technologies:

* Docker
* Hibernate
* Spring Boot 2.x

# Wallet Server

The wallet server must expose the interface described below via gRPC.

## Interfaces

## **Deposit**

Deposit funds to the users wallet.

## Input

* User id
* Amount
* Currency (allowed values are EUR, USD, GBP)

## Output

* No output needed

## Errors

* Unknown currency

## **Withdraw**

Withdraw funds from the user wallet.

## Input

* User id
* Amount
* Currency (allowed values are EUR, USD, GBP)

## Output

* No output needed

## Errors

* Unknown currency, insufficient funds

## **Balance**

Get the users current balance.

## Input

* User id

## Output

* The balance of the users account for each currency

## Integration Test

1. Make a withdrawal of USD 200 for user with id 1. Must return "insufficient\_funds".
2. Make a deposit of USD 100 to user with id 1.
3. Check that all balances are correct
4. Make a withdrawal of USD 200 for user with id 1. Must return "insufficient\_funds".
5. Make a deposit of EUR 100 to user with id 1.
6. Check that all balances are correct
7. Make a withdrawal of USD 200 for user with id 1. Must return "insufficient\_funds".
8. Make a deposit of USD 100 to user with id 1.
9. Check that all balances are correct
10. Make a withdrawal of USD 200 for user with id 1. Must return "ok".
11. Check that all balances are correct
12. Make a withdrawal of USD 200 for user with id 1. Must return "insufficient\_funds".

# Wallet Client

The wallet client will emulate a number of users concurrently using the wallet.   
The wallet client must connect to the wallet server over gRPC.

The client eliminating users doing rounds (a sequence of events). Whenever a round is needed it is picked at random from the following list of available rounds

Round A

* Deposit 100 USD
* Withdraw 200 USD
* Deposit 100 EUR
* Get Balance
* Withdraw 100 USD
* Get Balance
* Withdraw 100 USD

Round B

* Withdraw 100 GBP
* Deposit 300 GPB
* Withdraw 100 GBP
* Withdraw 100 GBP
* Withdraw 100 GBP

Round C

* Get Balance
* Deposit 100 USD
* Deposit 100 USD
* Withdraw 100 USD
* Depsoit 100 USD
* Get Balance
* Withdraw 200 USD
* Get Balance

## The wallet client CLI parameters

1. Users (number of concurrent users emulated)
2. Concurrent\_threads\_per\_user (number of concurrent requests a user will make)
3. Rounds\_per\_thread (number of rounds each thread is executing)

Make sure the client exits when all rounds has been executed.

# Reporting

Together with your source code please include the following:

* Instructions on how to run the client and the server
* Explanation of important choices in your solution
* Estimate on how many transactions your wallet can handle per second on your development machine (and an explanation on how you reached that number)

# Deadline

The project must be submitted for review at the most three weeks after the task has been assigned.

# Submission and Follow Up

Please submit the task to [valentina.frei@betpawa.com](mailto:valentina.frei@betpawa.com) AND aleksei.makagonov@betpawa.com

After submission I will contact you and schedule a follow up interview. In the interview we will review you solution and discuss various choices. The interview will be scheduled within one week of submission.

Thank you.