PW Application

Тестовое задание для создания небольшого приложения для осуществления переводов виртуальных денег (PW) между пользователями приложения.

Задание приведено ниже на английском языке, рассматривайте его не как готовое ТЗ, а как первый материал, присланный заказчиком.

Задание предназначено для претенд на различные должности. В зависимости от претендуемой должности выполните следующее:

* Аналитик

Выясните все вопросы у заказчика (на английском языке) и составьте

1. макеты экранов приложения,
2. подробное техническое задание

* Руководитель проекта

Сделайте все, что написано выше для аналитика, плюс составьте

1. Список задач (Task Breakdown) с оценкой трудоемкости реализации каждого пункта,
2. Календарный план реализации проекта.
3. Представьте план реализации директору компании (на русском) и заказчику (на английском).

* Программист .NET

1. Уточните все непонятные вам детали приложения, сформулировав вопросы в письменном или устном виде (можно на русском, английский приветствуется).
2. Создайте серверную часть приложения
3. Создайте веб приложение, используя стандартную тему интерфейса.

* Программист Front-End

1. Уточните все непонятные вам детали приложения, сформулировав вопросы в письменном или устном виде (можно на русском, английский приветствуется).
2. Если Вы не пишете серверную часть, то мы вышлем Вам документацию на существующие веб-сервисы.
3. Создайте веб приложение.

* Разработчик мобильных приложений

1. Уточните все непонятные вам детали приложения, сформулировав вопросы в письменном или устном виде (можно на русском, английский приветствуется).
2. Если Вы не пишете серверную часть, то мы вышлем Вам документацию на существующие веб-сервисы.
3. Создайте мобильное приложение.

* Веб/мобильный дизайнер

Предложите нам варианты оформления (дизайна) приложения согласно guidelines: [UI design requirements](#_4v5b5sv7ohia). Пришлите PSD файл(ы).

Вы можете выполнить несколько заданий, чем больше Вы сделаете, тем больше вероятность успешно пройти собеседование.

# PW Application Overview

The application is for Parrot Wings (PW, “internal money”) transfer between system users.

The application will be very “polite” and will inform a user of any problems (i.e. login not successful, not enough PW to remit the transaction, etc.)

## User registration

Any person on Earth can register with the service for free, providing their Name (e.g. John Smith), valid email (e.g. [jsmith@gmail.com](mailto:jsmith@gmail.com)) and password.

When a new user registers, the System will verify, that the user has provided a unique (not previously registered in the system) email, and also provided human name and a password. These 3 fields are mandatory. Password is to be typed twice for justification. No email verification required.

On successful registration every User will be awarded with 500 (five hundred) PW starting balance.

## Logging in

Users login to the system using their email and password.

Users will be able to Log out.

No password recovery, change password, etc. functions required.

## PW

The system will allow users to perform the following operations:

1. See their Name and current PW balance always on screen
2. Create a new transaction. To make a new transaction (PW payment) a user will
   1. Choose the recipient by querying the User list by name (autocomplete).
   2. When a recipient chosen, entering the PW transaction amount. The system will check that the transaction amount is not greater than the current user balance.
   3. Committing the transaction. Once transaction succeeded, the recipient account will be credited (PW++) by the entered amount of PW, and the payee account debited (PW--) for the same amount of PW. The system shall display PW balance changes immediately to the user.
3. (Optional) Create a new transaction as a copy from a list of their existing transactions: create a handy UI for a user to browse their recent transactions, and select a transaction as a basis for a new transaction. Once old transaction selected, all its details (recipient, PW amount) will be copied to the new transaction.
4. Review a list (history) of their transactions. A list of transactions will show the most recent transactions on top of the list and display the following info for each transaction:
   1. Date/Time of the transaction
   2. Correspondent Name
   3. Transaction amount, (Debit/Credit for PW transferred)
   4. Resulting balance

(Optional) Implement filtering and/or sorting of transaction list by date, correspondent name and amount.

# Architecture requirements

1. WEB API application, server-side.

WEB API must implement RESTful service.

1. HTML front end application
2. Mobile application

HTML frontend and mobile application will implement full application functionality, and communicate with WEB API using JSON.

# Technical requirements

Use the following software:

1. Server side
   1. MS SQL Server 2012 or higher
   2. IIS
   3. Web API
   4. Entity framework 6
2. WEB HTML client
   1. HTML5
   2. Javascript or TypeScript
   3. AngularJS or ReactJS
   4. (optional) primeNG, material, bootstrap
3. Mobile application
   1. iOs or Android (at your choice, or both platforms)
   2. C# with Xamarin development environment

# UI design requirements

Customer likes Material Design approach as well as native Mac OS look'n feel.

It is necessary to design and implement unique UI look'n feel for the front-end applications, for every UI elements, and (optionally) apply it to developed frontend application.