Introduction

# Idea

The idea is to have a platform where users can exchange currency pairs that are not available on the foreign exchange market. For example, the MAD (Moroccan Dirham) is only available for exchange in very limited quantities per year, you can only exchange around $4000 per year from MAD to another currency (EUR, USD, ...). Which is very limiting. The idea is to connect users who want to buy MAD with those who want to sell EUR, USD or any other currency for MAD, by using internal bank transfers only, which is fully legal.

# Example

* User A wants to get 5000 MAD into his Moroccan bank account and he has 450€ (1€ ~= 10.80 MAD).
* User B is willing to get 450€ in his EUR bank account in exchange for sending 5000 MAD to a Moroccan bank account.

In this scenario, the user A will send 450€ to user B's bank account and user B will send 5000 MAD to user A's bank account. In the end no MAD has left the country and there was just a simple internal exchange within the boundaries of each jurisdiction. This could facilitate business and it will be an alternative and cheap (if not free) way for sending money abroad.

# Scenarios

* A user (requester) posts a trade specifying the amount of money he wants to receive
* Other users (senders) can respond to his trade and make offers of exchange rates they are willing to send the money for
* Once the requester accepts a demand, he provides his bank details on where he wants to receive the money
* The exchange is then conducted depending on the chosen scenario
  + Scenario 1: Full Escrow
    - The requester makes a deposit into Tabdyl's bank account
    - The sender sends the equivalent sum to Tabdyl's bank account
    - Tabdyl's platform will then proceed to transfer the sums between the parties
  + Scenario 2: Partial Escrow
    - The requester makes a deposit into Tabdyl's bank account. This money is then held until confirmation
    - The sender sends the equivalent sum to the requester bank account
    - Once the requester receives the money, he confirms the reception
    - Tabdyl's platform will then proceed to send the held money to the sender's bank account he provided
  + Scenario 3: No Escrow
    - The requester sends a deposit to the sender’s bank account
    - The sender sends the equivalent sum to the requester bank account
    - Once the requester receives the money, he confirms the reception
    - Once the sender receives the equivalent money, he confirms the reception

# Details

Each scenario has its peculiarities:

* In the Full Escrow Scenario, it’s the platform that guarantees the exchange. There will be no need for confirmation and thus no claims can be opened nor documents are needed. It is the safer option.
* In the Partial Escrow Scenario, the requester can open a claim if it took more 5 working days and he still didn’t receive the money. The sender can also open a claim if the he actually sent the money and the requester still didn’t confirm it after 5 working days. In either case an investigation can be opened.
* In the No Escrow Scenario, the requester needs to provide a screenshot or a scan of his bank wire to Tabdyl. After which, Tabdyl will give the green light to the sender to send his money too, and he also should provide a document to prove the bank wire. They both can open a claim if they haven’t received their money after 5 working days.

Problems and Solutions

**Problem: The transfers are going to be slow**

The problem is that in all of these scenarios the transfer is not in real-time, the waiting period could take days or weeks (in case of No Escrow Scenario) which may be a deterrent to people in need of fast transfers.

**Solution: Full Escrow Scenario + Payable amount for insurance**

A faster (but payable) solution should be provided in which Tabdyl’s platform will proceed to exchanges instantaneously without waiting to receive the money from either party (a solution only applicable to Full Escrow Scenario). The transfer will thus be fast and will only be constrained by the parties’ banks’ accounts processing times.

**Problem: How to investigate usurpers**

In the Partial Escrow and No Escrow Scenarios, there could be usurpers who will not send the money and claim they did.

**Solution: Investigation involving original documents**

Conduct and investigation, in which Tabdyl will ask the party on the wrong side to provide a bank statement proving the bank wire was made, or the document provided by the bank who made the transfer if it was made with cash. The process should be further strengthened.

**Problem: The platform could be used for money laundering**

Some people may use the platform to legitimize their source of revenues. Especially if you allow cash for transfers.

**Solution: Implement a lightweight version of AML policies**

Limit the amounts of cash transfers a person could do per month. Implement a very lightweight version for Anti-Money Laundering policies that will make sure the money comes from legit sources.

**Problem: The platform needs a huge number of users to be operable**

The platform could not work without the contribution of users, and it needs a lot of them to be able to function.

**Solution: Create fake users**

Create fake users that will all be you in the end, you will be the one fulfilling the transactions if the first months. You will pay from your own pockets (you will not lose money, at least not too much, you will just get money in currencies you are not in need of).

**Problem: The users could bypass the platform**

Once two users have completed a transaction and they built some kind of trust between them, they may start to do exchanges without using the platform. Hence defying the purpose of monetizing the platform and taking a percentage once the platform has matured

**Solution: Retention and rewards**

Provide a reward system to enhance retention rates and make the users always find using the platform is advantageous to them, even after monetizing it. Like givebacks depending on the number of transactions completed.

Functions

# Overview

The user should be able to:

* Account
  + Create an account
  + Login / Logout
  + Profile (show, update)
  + Settings (show, update)
  + Dashboard
    - Wallet
    - Search alerts
    - Ongoing trades
    - Transactions history
* Trading
  + Trade (post, deactivate, delete)
  + Offer (make, withdraw)
* Search
  + Search for trades
  + Apply filters to trades
  + Respond to a trade
* Messaging
  + Send a message to a user
  + Show messages
* Reviews
  + Give review to a user
  + List reviews of a user
* Notifications
  + Show notifications of events

# Details

**Bank accounts**

A user can have many bank accounts.

**Trade / Offer dynamics**

Once a trade has been posted, there are two ways for a match to occur:

* A sender (user) accepts the trade with the proposed\_exchange\_rate
* A user will make an offer with another exchange\_rate and the requester (user) accepts it

If the sender accepts the trade as it is, he will have to choose one of the payment\_methods proposed in the in the trade. As such, the transaction will take on the exchange\_rate of the trade and the chosen payment\_method.

If the sender makes an offer and the requester accepts it, the created Transaction will take on the details of the offer.

**Wallet**

A wallet service, to which the user could send money so as to be ready when he needs to make a transfer. It will make the exchange faster, since it’ll bypass the part where he needs to send the money to the platform’s bank account and wait for it to be credited. That way, the money will be ready to use, and it could be interesting for the platform too, investment-wise.

**Tiers**

A tiered account service, for each tier, a corresponding KYC data must be completed. For example:

* Tier 1: Provide personal information with no supporting documents. 1000€ max transfers
* Tier 2: Provide ID document to support the information. 2000€/month transfers
* Tier 3: Provide the proof of address and a second ID document. 5000€/month transfers

**Partial fulfillment**

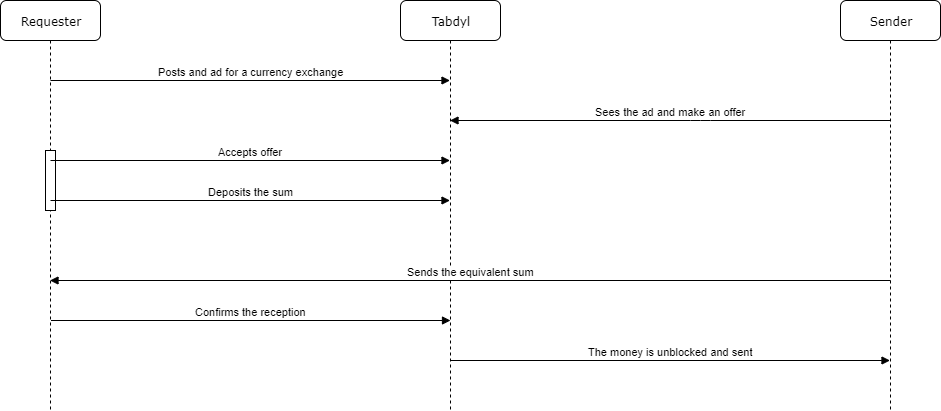
Another feature that could interesting to add is partial trade fulfillment. It is when an offer can fulfill only a part of the trade amount. For example, A trade for 1000€ is opened and someone is interested in it but only partially, he can only exchange the equivalent of 400€. So, he makes an offer, and if accepted and completed, the trade automatically updates itself for the remaining amount, and so on, until all of it is exchanged. This will need to give the trade poster an option to activate this option or not, and to giver the offer poster an option to specify the amount he’s willing to exchange.

Analysis

# Sequence diagram

## Full escrow

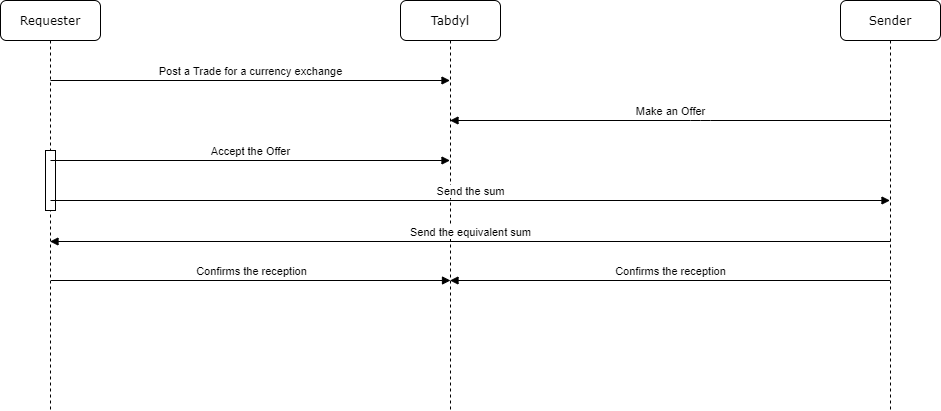
#### **Partial escrow**



Loop over offers until the Requester accepts one.

The Sender can also see the trade and accept it as it is, without making a counter-offer, it implicitly creates an offer with the same details as the trade.

## No escrow



# Class diagram

## Conception

**User** (id, username, email, password, firstname, lastname, birthdate, gender, address, city, zipCode, country, nationality, phoneNumber, idDocumentType, idDocument, photo, bankAccounts, reviews, tier, lastConnectionDate, creationDate, lastUpdateDate, deactivated)

**Bank\_Account** (user, label, country, currency, bank\_name, iban/rib, swift)

**Trade** (title, description, amount, base\_currency, target\_currency, proposed\_exchange\_rate, accepted\_payment\_methods, escrow\_mode, fulfillment\_mode, status)

**Offer** (description, currency, amount\_to\_fulfill, exchange\_rate, payment\_method, datetime, status)

**Transaction** (ad, offer, state, datetime)

**Message** (text, from, to, datetime)

**Review** (user, stars, comment, trusted)

**Notification** (user, event, description)

There will be other tables for fixed values like Payment methods, Banks names, and such.

## Diagram

TODO

User Interface

# Idea 1

The frontend will be made of two parts:

* Public: For descriptions of the platform, services, help, and signing up/in
* Private: Once the user has signed in, he can search for trades and make full use of the platform

If you chose this idea you may use the StartBootstrap template:

<https://shop.startbootstrap.com/product/pro-angular-bundle/>

It’s made of Bootstrap and already integrates Angular. It consists of two parts a simple website UI for presenting the platform’s services, and an trademin template for managing everything else. It’s a fully featured and easy to customize template.

The only downside I see to this idea, is that the trades won’t be visible for users without creating an account and login in, which may deter users that just want to see how’s the platform to make an idea of it and use it later. Making everything accessible except the exchange feature (and the surrounding features) is a great way to showcase the platform and give first time comers a glimpse into the services.

# Idea 2

The trades will be accessible on the first page, there will be no special frontend to talk about the platform, just the bare search functions and everything will be accessible except actually exchanging, messaging, and commenting

The platform will look like:

<https://localbitcoins.com/>

The trades section will provide a way to post trades whether for exchanging currencies, filters and search options to navigate through all the current offers.

The account section will be for managing your account, providing your bank details and proving your identity as well as the KYC process. Also creating alerts for trades with specific criteria.

There will also be a review system to give a rate and a feedback to users and build trust There will also be the possibility to message the user once they are matched

# Mockup

TODO

Technologies

# Common

Backend: Spring Boot + Lombok

# V0: Prototyping (Proof of Concept)

Frontend: jQuery + Bootstrap

Database: MySQL

Infrastructure: Windows + GitHub

# V1: MVP (Production-ready)

Frontend: jQuery + Bootstrap

Database: PostgreSQL

Infrastructure: Linux (on premise) + Tomcat + Apache + GitHub

# V2: Growing (Success)

Frontend: Angular + Bootstrap

Database: PostgreSQL

Infrastructure: Linux (cloud) + GitLab + Tomcat + NGINX + microservices + Redis + Docker + Kubernetes + ELK stack for logging + Prometheus/Grafana for metrics

Monetization

At first, the platform should be completely free, with no fees whatsoever, until it reaches a respectable number of transaction and notoriety. After which a small percentage should be applied to performed transactions.

Example

* 0.5% for EUR/USD to MAD transactions
* 1% for MAD to EUR/USD transactions

You may also offer prepaid packages or subscriptions.

Example

* 20€/month for up to 5 transactions per month
* 50€/month for up to 20 transactions per month

Any user doing more than 20 transactions per month could be seen as a suspicious activity. They’ll need to contact for a personalized offer and provide some documents to prove they are legit.

You may also forego all of this and just implement Google trades or something that will monetize the traffic without you managing this aspect.