

Санкт-Петербургский государственный университет

Кафедра системного программирования

Программная инженерия

Лунев Артем Евгеньевич

Развитие платежной системы для
взаимодействия с различными
реализациями blockchain

Курсовая работа

Научный руководитель:
ст. преп. Кириленко Я. А.

Консультант:
Ф.П. Долголев

Санкт-Петербург

2019

Оглавление

Введение	3
1. Цели и задачи	4
1.1 Цель работы	4
1.2 Поставленные задачи	4
2. Обзор существующих решений	5
3. Обзор проекта	6
4. Реализация	8
4.1 База данных	8
4.2 Депозитные счета	9
Заключение	11
Список литературы	12

Введение

В современных условиях криптовалюты начинают обретать все большую и большую популярность. Это происходит за счет безопасности обмена, высокой скорости проведения платежей, анонимности сделок, децентрализованности.

Из-за этих, а также многих других преимуществ появляются люди, заинтересованные в удобном сервисе для выполнения финансовых операций в различных криптовалютах, в особенности построенных на технологии блокчейн. Многие интернет магазины, биржи и предприниматели уже сейчас работают с криптовалютами и испытывают потребность в ее приеме и отправке.

1. Цели и задачи

1.1. Цель работы

Цель данного проекта – разработка системы, позволяющей производить финансовые операции с различными криптовалютами на основе blockchain.

1.2. Поставленные задачи

Для выполнения обозначенной цели были выделены следующие задачи:

- Изучить предметную область
- Сделать обзор существующих решений
- Разработать схему базы данных и реализовать работу с БД
- Добавить возможность заведения постоянного депозитного счета с множеством адресов

2. Обзор существующих решений

В данной таблице приведены сведения о наиболее популярных платежных системах. Были выделены 3 критерия: количество поддерживаемых валют, предоставление кастодиальных услуг, открытость исходного кода. Кастодиальные сервисы хранят валюту на своих кошельках с возможностью вывода. Некастодиальные, наоборот, хранят средства на кошельках клиентов.

сервис	валюты	кастодиальный	open-source
Bitpay	2	+	-
Coinsbank	4	+	-
Coingate	50+	+	-
Coinbase commerce	4	-	-
Gocoin	6	-	+

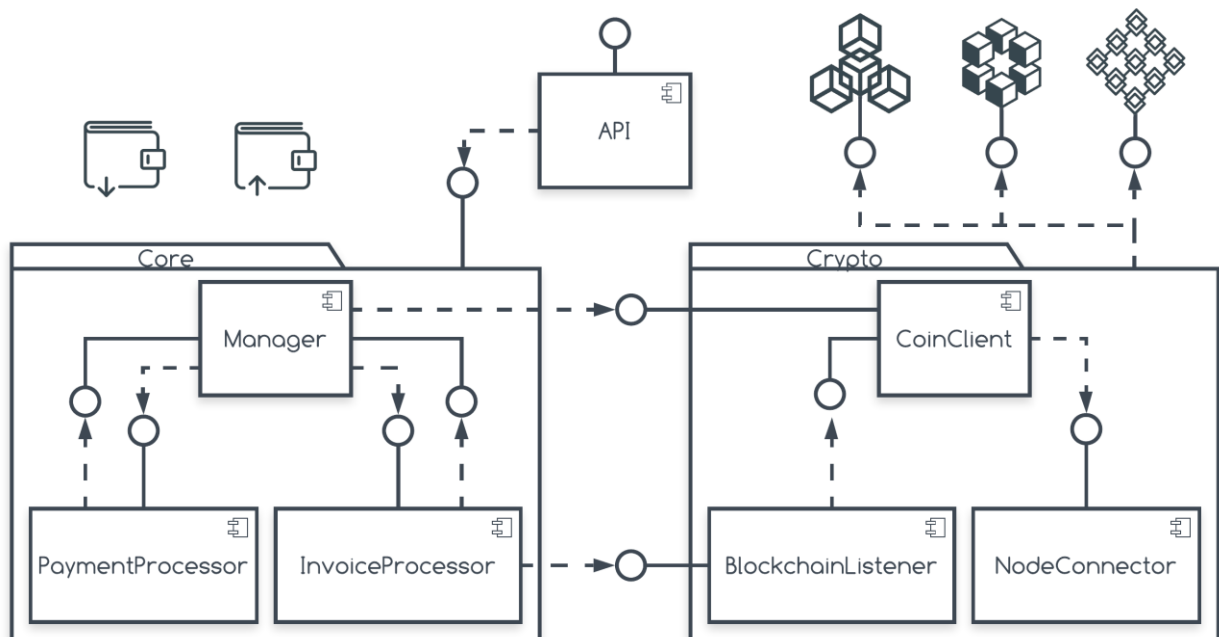
Одна из проблем криптовалютных платежных систем – это то, что большинство из них закрытые и стоят довольно дорого, поэтому подходят не всем. Из-за этого многие предприятия создают свои системы.

Данный проект является open-source. Его можно менять под свои нужды и использовать как полноценную платежную систему, не изобретая вновь и вновь одно и то же.

3. Обзор проекта

Данный проект является продолжением курсовой Скаредова Сергея Антоновича “Унификация взаимодействия с различными реализациями blockchain в качестве платежных систем”.

Структура сервиса разделена на два функциональных модуля. Модуль Core отвечает за финансовые операции в контексте платежей и выставленных счетов (invoice). Модуль Crypto осуществляет взаимодействие с различными реализациями blockchain. Такая архитектура позволит легко масштабировать систему, добавляя поддержку новых криптовалют во второй блок, в то время как первый будет взаимодействовать с реализацией для каждой валюты по одному принципу.



PaymentProcessor хранит и создает объекты Payment (платежи), а также отслеживает их статус. Отправка транзакции, соответствующей платежу происходит в модуле Crypto. InvoiceProcessor отвечает за создание объектов Invoice (выставленные счета), отслеживание их статуса и следит за

количеством полученных средств. Manager координирует работу всей системы.

BlockchainListener отслеживает начисления на выставленные счета в сети. NodeConnector устанавливает соединение с узлами сети. CoinClient занимается отправкой платежей и отслеживанием балансов.

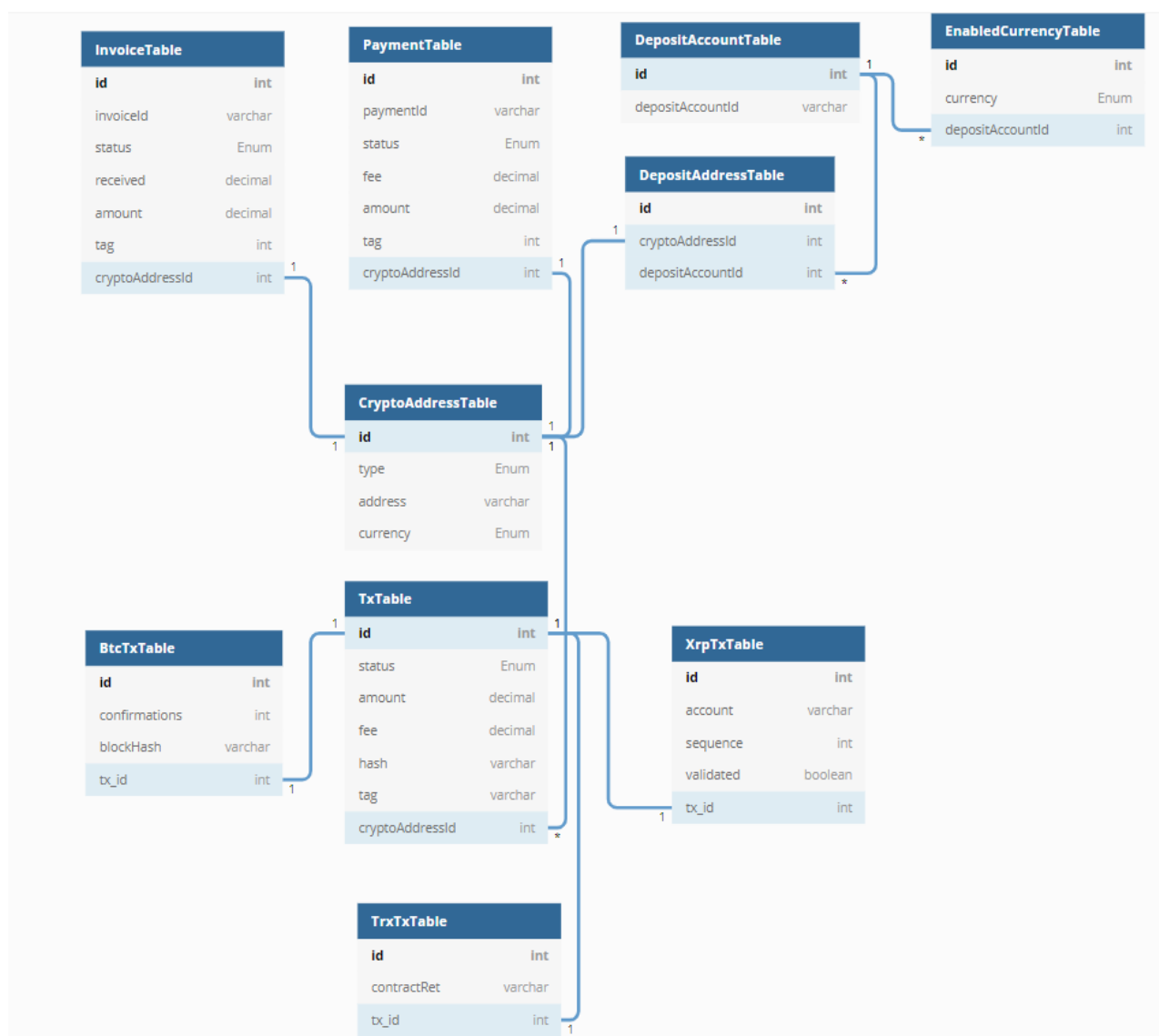
Разработка велась на kotlin из-за его совместимостью с java, компактности в написании кода, большого набора библиотек, в том числе предоставляемых blockchain-платформами.

4. Реализация

4.1. База данных

В качестве инструмента работы с базой данных был выбран ORM exposed. Технология ORM позволяет проектировать работу с базой в терминах классов, а не таблиц данных. Также она дает независимость от вида базы данных и избавляет от работы с sql. Exposed был выбран из-за его легковесности, простоты написания кода и описания таблиц и поддержки основных реляционных баз данных.

Далее была разработана база данных со следующей схемой.



Необходимо было хранить информацию о платежах, инвойсах, депозитных счетах, принятых и отправленных транзакциях.

В данном проекте образовалась следующая иерархия (перенесенная на базу данных): у платежей, инвойсов и депозитных счетов имеются соответствующие им адреса. На адреса поступают транзакции. Транзакции могут относиться к разным валютам, из-за чего отдельно есть таблицы, где хранится специфическая для каждой валюты информация.

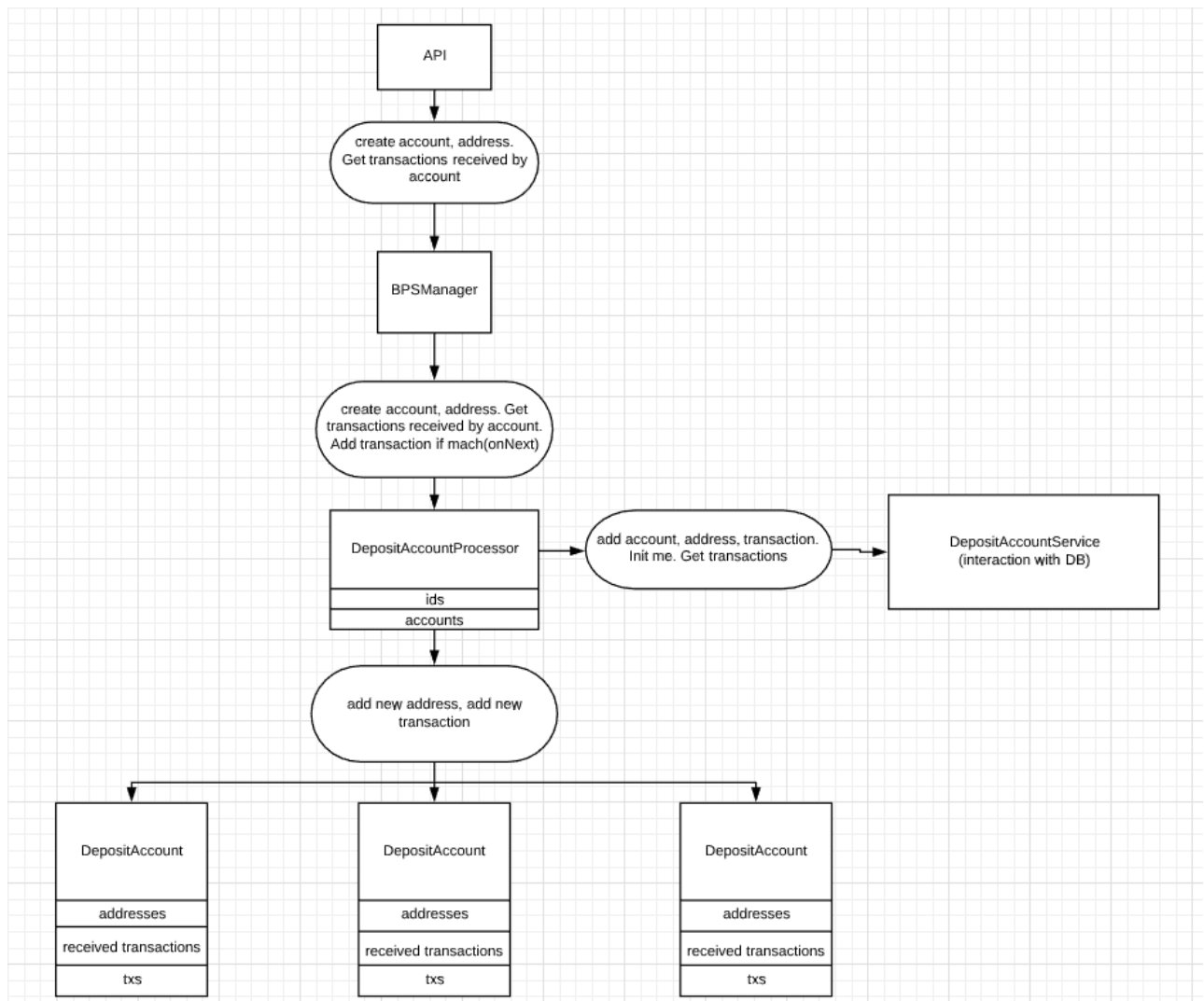
Так как у депозитных счетов может быть множество адресов, была создана таблица `DepositAddressTable`, являющаяся связующей между таблицей адресов (`CryptoAddressTable`) и таблицей счетов.

Далее были реализованы классы, отвечающие за работу с каждой таблицей. Только они имеют доступ к соответствующей таблице и могут с ней работать. После этого через них была реализована работа с базой данных.

4.2. Депозитные счета

Возможность заведения постоянного счета с множеством адресов, на которые непрерывно могут поступать депозиты является полезной функциональностью в некоторых ситуациях, например обработка депозитов на биржах, или же реализация многократной оплаты подписки на те или иные услуги.

В модуль `core` был добавлен сервис `DepositAccountProcessor`, который отвечает за работу со счетами, со следующей схемой взаимодействия.



Взаимодействие с API происходит через BPS Manager, как и в других Processor-ах. DepositAccountProcessor умеет создавать новые счета, адреса, выдавать информацию о пришедших транзакциях. Взаимодействие с базой данных происходит через специально отведенный для этого класс DepositAccountService. Также у данного Processor-а есть доступ ко всем депозитным счетам, которым он может отдавать команды добавить новый адрес или пришедшую транзакцию. Информацию о транзакциях он получает от BPS Manager-а, посредством паттерна Observer.

Заключение

В рамках курсовой работы были достигнуты следующие результаты:

- Изучена предметная область
- Проведен анализ существующих решений
- Разработана схема базы данных
- Реализована работа с базой данных
- Реализована работа с депозитными счетами

Список литературы

[1] Bitcoin: A Peer-to-Peer Electronic Cash System.
<https://bitcoin.org/bitcoin.pdf>

[2] Original Bitcoin client.
https://en.bitcoin.it/wiki/Original_Bitcoin_client/API_calls_list

[3] Iota hub. <https://docs.iota.org/docs/wallets/0.1/hub/introduction/overview>

[4] Binance exchange. <https://www.binance.com/ru/>

[5] Блокчейн изнутри: как устроен биткоин.
<https://vas3k.ru/blog/blockchain/>

[6] XRP accounts. <https://xrpl.org/list-xrp-as-an-exchange.html#accounts>

[7] 10 Best Bitcoin Payment Gateways for 2020.
<https://www.devteam.space/blog/10-best-bitcoin-payment-gateways/>

[8] Bitpay gateway. <https://bitpay.com/>