|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Информационная безопасность

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К курсовому проекту***

***НА ТЕМУ:***

***Сервис регистрации авторского права на технологии blockchain***

Студент ИУ8-31 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Безверженко И.В.

(Подпись, дата)

Руководитель курсового проекта **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Бородин А.А

(Подпись, дата)

*2019 г.*

**Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой ИУ8

Басараб М. А.

« \_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 2019 г.

**ЗАДАНИЕ**

**на выполнение курсового проекта**

по дисциплине Технологии и методы программирования

***Студент группы ИУ8-31:***

Безверженко Илья Вадимович

***Тема курсового проекта:***

Сервис регистрации авторского права на технологии blockchain.

***Направленность курсового проекта:***

исследовательская.

***Источник тематики:***

кафедра.

***График выполнения курсового проекта:***

25% к 3 нед., 50% к 9 нед., 75% к 12 нед., 100% к 15 нед.

***Техническое задание:***

Разработать сервис по регистрации, хранению и подтверждению авторского права, использующий технологию blockchain. Система будет поддерживать регистрацию и авторизацию пользователя, давать ему возможность в любой момент просматривать список зарегистрированным им прав на продукты. В проекте будет реализована пиринговая сеть для коммуникации blockchain’а на разных компьютерах, что позволит осуществлять совместную работу нескольких пользователей.

***Оформление курсового проекта:***

Расчетно-пояснительная записка на \_\_\_\_\_ листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «1» сентября 2019 г.

**Руководитель курсового проекта**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Бородин А.А

(Подпись, дата)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Безверженко И.В.

(Подпись, дата)

Оглавление

[Цель 4](#_Toc27161459)

[Основные определения 5](#_Toc27161460)

[Введение 6](#_Toc27161461)

[Требования к проекту 7](#_Toc27161462)

[Проектирование системы 8](#_Toc27161463)

[Blockchain 8](#_Toc27161464)

[Класс «Copyright» 8](#_Toc27161465)

[Класс «Block» 9](#_Toc27161466)

[Класс «Blockchain» 10](#_Toc27161467)

[P2P сеть 10](#_Toc27161468)

[GUI 11](#_Toc27161469)

[Статические объекты 13](#_Toc27161470)

[Динамические объекты 14](#_Toc27161471)

[Выбор технологий 16](#_Toc27161472)

[Выбор языка программирования 16](#_Toc27161473)

[Выбор используемых библиотек 18](#_Toc27161474)

[Описание технический решений 19](#_Toc27161475)

[Заключение 20](#_Toc27161476)

[Список используемых источников 21](#_Toc27161477)

# Цель

Цель проекта заключается в создании сервиса регистрации авторского права на технологии blockchain для определения рентабельности blockchain проектов, не связанных с криптовалютой; для того, чтобы ответить на вопросы: Почему использование технологии blockchain оправдано только в случае создания криптовалюты? С какими проблемами сталкиваются разработчики «некриптовалютных» blockchain проектов? Если их можно решить, то как? Стоит ли инвестировать в blockchain проекты?

# Основные определения

* AVTOR – название проекта.
* P2P сеть – одноранговой, децентрализованная или пиринговой сети. Часто в такой сети отсутствуют выделенные серверы, а каждый узел (peer) является как клиентом, так и выполняет функции сервера.
* «Праворегистрация», или «copyright», – единица измерения регистрация права на продукты, которая содержит в себе информацию об авторе, продукте, дате и времени регистрации.
* GUI (Графический интерфейс пользователя) — разновидность пользовательского интерфейса, в котором элементы интерфейса, представленные пользователю на дисплее, исполнены в виде графических изображений. Также называется графической оболочкой управления.
* Blockchain — выстроенная по определённым правилам непрерывная последовательная цепочка блоков (связный список), содержащих информацию. Каждый блок содержит свою собственную хеш-сумму и хеш-сумму предыдущего блока. Для изменения информации в блоке придётся редактировать и все последующие блоки. Чаще всего копии цепочек блоков хранятся на множестве разных компьютеров независимо друг от друга.
* Коллизия – это равенство значений хеш-функции на двух различных блоках данных.
* Конценсус – это набор определенных математических правил и функций, регулирующих работу сети
* Хеширование — преобразование по определённому алгоритму входного массива данных произвольной длины в выходную битовую строку фиксированной длины.
* Broadcasting – метод передачи данных в компьютерных сетях, при котором поток данных (каждый переданный пакет в случае пакетной передачи) предназначен для приёма всеми участниками сети.

# Введение

Проект AVTOR является простым и наглядным примером blockchain проекта, никак не связанного с криптовалютой.

Это сервис регистрации авторского права. Он работает следующим образом: богемный человек создал какой-нибудь продукт (это может быть картина, музыка, роман, текс песни, код программы, революционный проект на blockchain’е, что угодно). Он хочет зарегистрировать свое авторское право на него: запускает нашу программу, авторизуется и добавляет свой продукт в цепь блоков. Сервис записывает время регистрации продукта. Отныне технология blockchain гарантирует его авторство.

На первый взгляд – все замечательно. Теперь творческому человеку не придется стоять в огромных очередях для подачи заявки в патентное бюро, платить немалую пошлину, заполнять бланки (которые крайне сложно правильно заполнить без юридической помощи). Сама процедура не быстрая, зачастую длится от нескольких месяцев до года. Тут сервис регистрации права на технологии blockchain – куда более удобная, простая, быстрая и дешевая альтернатива. Это означает, что такой сервис будет очень успешным стартапом, который станет популярным во всех творческих кругах по всему миру, принесет своим создателям горы зеленых бумажек. Но почему же так до сих пор не произошло? Значит не все так просто, как кажется на первый взгляд. Но в чем же проблема? В этом нам и предстоит разобраться в рамках нашей курсовой работы.

# Требования к проекту

* Возможность авторизовываться в сервисе
* Возможность создавать новый профиль в сервисе (регистрироваться)
* Легкий способ регистрировать права на продукт в сервисе
* Простота проверки актуальности продукта, т.е. был ли он уже зарегистрирован в цепи блоков
* Возможность посмотреть список своих «праворегистраций», которые были добавлены в blockchain, и узнать подробную информацию о них
* Реализация P2P сети, которая позволяет программам, запущенным на разных PC одновременно, обмениваться актуальными данными

# Проектирование системы

Для того, чтобы реализовать данный проект, было решено разделить его на три основные части: blockchain; P2P сеть и GUI.

## Blockchain

В данном случае под пунктом blockchain подразумевается совокупность объектов (классов) и их методов, которые формируют технологию blockchain. Первая структура – это «Copyright», или «праворегистраци». Следующая структура – это «Block», который состоит из некоторых своих параметров (их мы рассмотри чуть позже) и нескольких «праворегистраций», максимум три (в нашем случае). Финальным классом являет, собственно, сам «Blockchain», «цепь блоков». Он содержит в себе все созданные блоки. Теперь давайте поподробнее остановимся на каждой из этих структур.

### Класс «Copyright»

Класс «праворегистрации» создается, когда пользователь регистрирует свои права на какой-то продукт. Он содержит в себе имя автора (его логин), хэш продукта, хэш веса продукта и дату регистрации.

Хэширование используется для того, что цепь весила мало. Если blockchain будет хранить все зарегистрированные продукты, то цепь не поместится ни на одном компьютере. Благодаря хэшированию пользователь может зарегистрировать как маленький txt-файл с своим стихом, так и серию снятого им сериала, которая в хорошем разрешении может весить десятки гигабайт.

Так как хэш функции подвержены коллизии, то есть хэши от разных файлов могут совпасть, мы используем еще и хэш веса файла. Ведь вероятность того, что оба хэша разных продуктов совпадут, ниже, чем вероятность аналогичной ситуации при использовании только одного хэша.

### Класс «Block»

Класс «блок» объединяет в себе несколько «праворегистраций». Как только в блоке их накапливает три штуки, блок начинается закрываться.

Это работает следующим образом: блок содержит хэш предыдущего блока, который формируется на основе всех его параметров. То есть если мы изменим какую-нибудь информацию в закрытом блоке, то его хэш не сойдется с хэшом, записанным в потомке. Отсюда мы сразу узнаем, что blockchain был изменен злоумышленником.

После того, как в блок были добавлены три «праворегистраци», мы ищем корень дерева Меркле. Вкратце, это алгоритм, использующий хэширование, целью которого является отследить внесенные злоумышленником изменения в «праворегистрации». На вход он получает хэши всех «праворегистраций» в блоке. На выходе выдает один хэш – корень дерева Меркле.

Затем начинается процесс майнинга. О нем, думаю, слышал абсолютно каждый. Если у обычного обывателя спросить, что он знает о майнинге, зачастую получишь ответ: «Вычислительную работу твоего компьютера эксплуатируют в каких-то непонятных целях, а за это ты получаешь крипту». И это не далеко от правды. Программа использует вашу вычислительную мощность для подбора параметра Nonce, который есть у каждого закрытого блока. Правильный, подобранный Nonce характеризуется тем, что вместе и ним хэш закрывающегося блока начинает удовлетворять какому-то выбранному условию, которое может быть абсолютно любое (к примеру, хэш начинается фразой «helloworld» или заканчивается пятью нулями).

После того, как нужный параметр Nonce был подобран, создается новый блок, а в него помешается хэш предыдущего блока, который мы так усердно майнили.

### Класс «Blockchain»

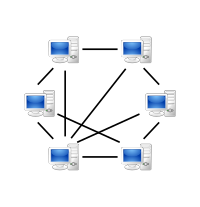
Класс «цепь блоков» является финальным. Его структура крайне проста: это просто последовательность взаимосвязанных между собой блоков.

Главная цепь этого класса – проверка истинности данных в цепи. Как уже писалось ранее, это происходит при сравнении хэша блока и хэша, который был ранее записан в потомка.

В данной курсовой я используют функцию хэширования SHA256 из библиотеки PicoSHA2.

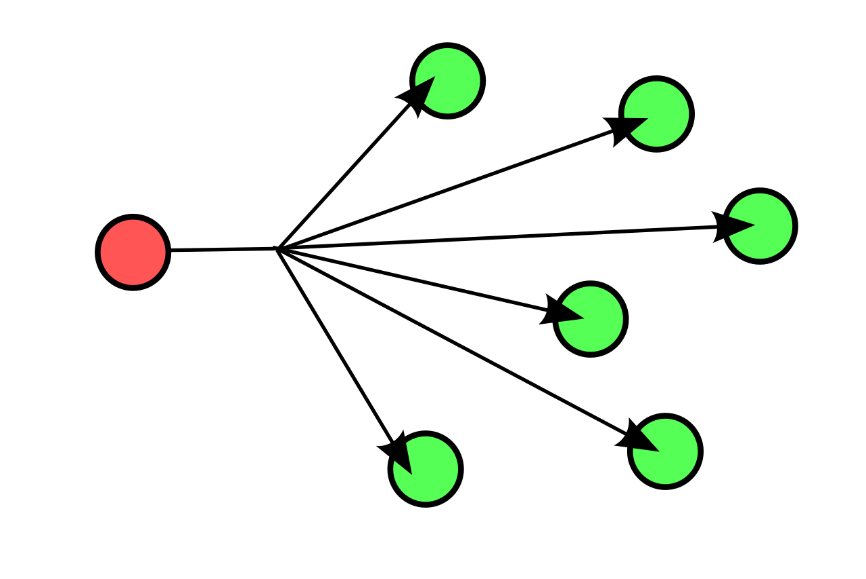
## P2P сеть

Реализация одноранговой, децентрализованная или пиринговой сети.



***Рис.1*** *Схема P2P сети.*

Каждый пользователь должен отправлять и принимать данные одновременно. Так как реализовывалось все в локальной сети, то использовался протокол udp метод broadcast из библиотеки Boost для C++.



***Рис.2*** *Схема широковещания (bloadcasting).*

Был написан класс «Buffer», в котором хранятся данные для отправки в сеть (после отправки они удаляются) и данные, полученные из сети (которые обрабатываются программой, а затем удаляются). Клиент и сервер работают параллельно.

## GUI

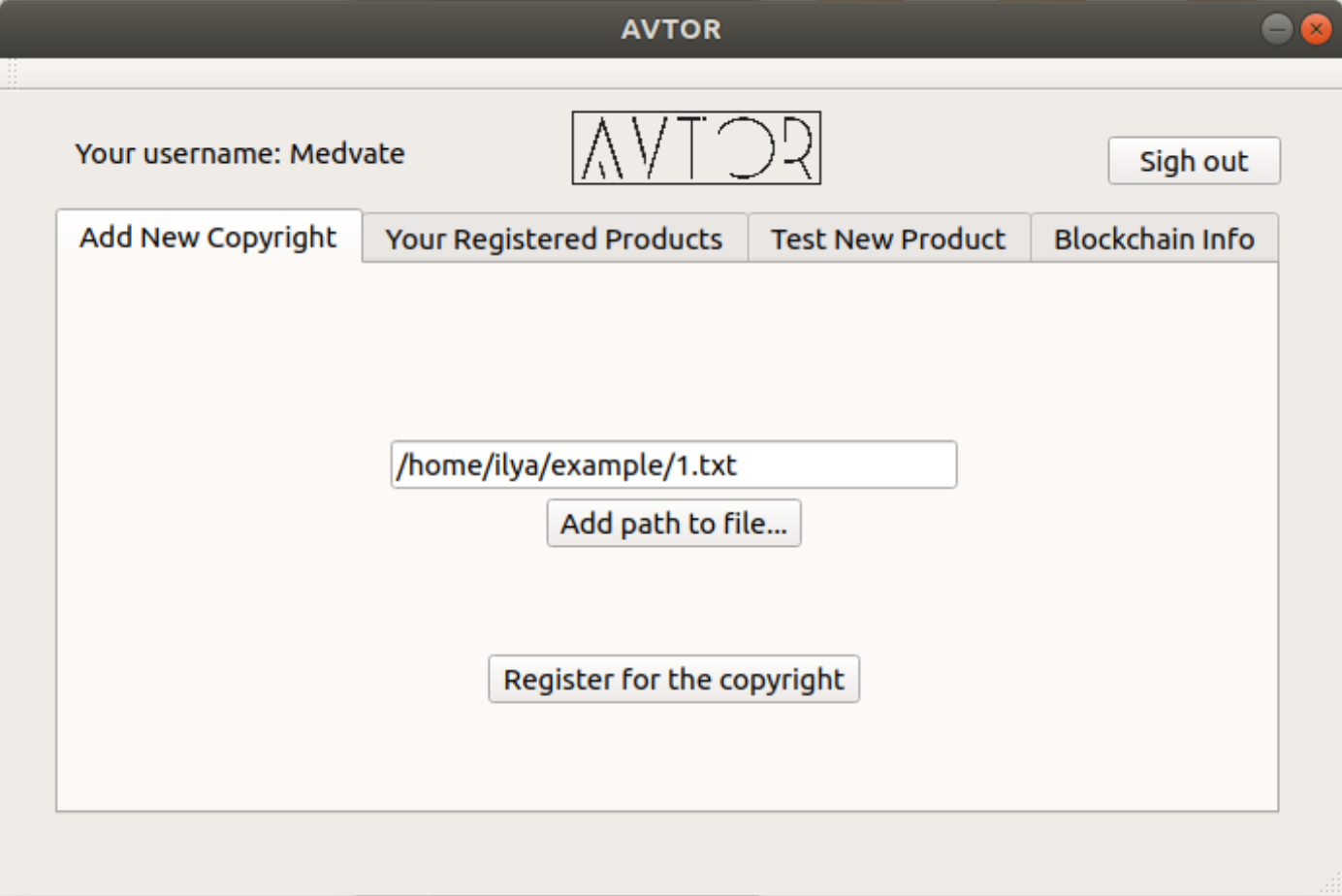
Графический пользовательский интерфейс было решено делать в Qt Creator по следующим причинам: интуитивно понятный процесс создания GUI; это IDE для разработчиков на C++ (наш проект был написан именно на этом языке программирования); его кроссплатформенность.

Когда пользователь запускает приложение, ему становится доступна панель авторизации, где он может ввести свой логин и пароль и войти в систему. Если же это новый пользователь, то он может открыть панель регистрации, после окончания которой также войдет в систему.

Когда пользователь успешно авторизуется, то откроется основное окно программы, которое состоит из четырех вкладок. Каждая из вкладок предоставляет пользователю свои возможности. Рассмотрим каждую из вкладок.

### Вкладка 1. «Add new copyright» (Рис.3)

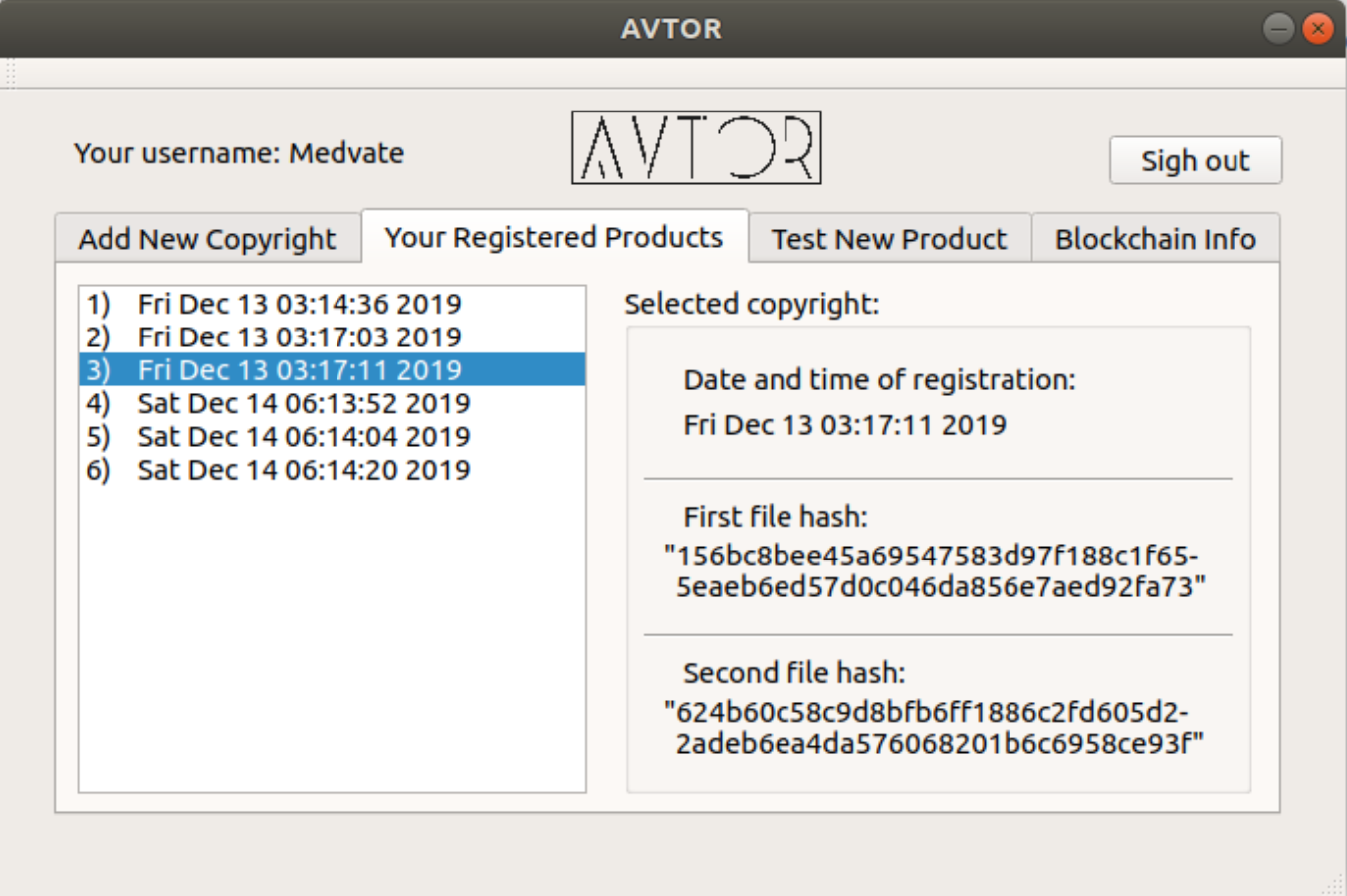
Пользователь добавляет путь до продукта и регистрирует его.



***Рис.3***

### Вкладка 2. «Your registered products» (Рис.4)

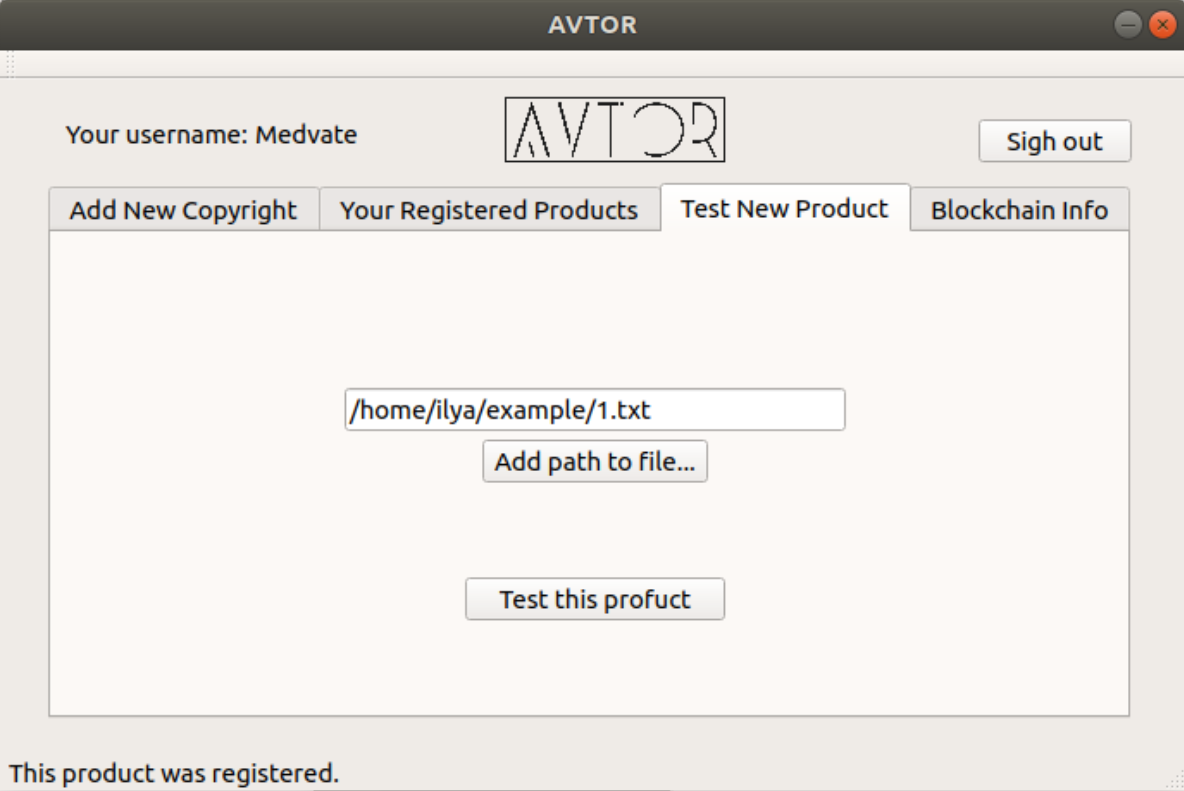
Пользователь получает список всех своих праворегистраций. Также он может посмотреть подробную информацию о каждой из них.



***Рис.4***

### Вкладка 3. «Test new product» (Рис.5)

Пользователь может указать путь до продукта и проверить, был ли он когда-то зарегистрирован.



***Рис.5***

### Вкладка 4. «Blockchain info»

Пользователь получает информацию о нынешнем состоянии цепи.

# Выбор технологий

## Выбор языка программирования

***Рассмотрим язык программирования C++. У него много плюсов:***

* Поддерживаются различные стили и технологии программирования, включая традиционное директивное программирование, ООП, обобщённое программирование и метапрограммирование;
* Предсказуемое выполнение программ. Весь код, неявно генерируемый компилятором, определён в стандарте. Также строго определены места программы, в которых этот код выполняется;
* Автоматический вызов деструкторов объектов при их уничтожении, причём в порядке, обратном вызову конструкторов;
* Используя шаблоны, возможно создавать обобщённые контейнеры и алгоритмы для разных типов данных, а также специализировать и вычислять на этапе компиляции;
* Кроссплатформенность: стандарт языка накладывает минимальные требования на ЭВМ для запуска скомпилированных программ;
* Эффективность. Язык спроектирован так, чтобы дать программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы. Ни одна из языковых возможностей, приводящая к дополнительным накладным расходам, не является обязательной для использования — при необходимости язык позволяет обеспечить максимальную эффективность программы;
* Имеется возможность работы на низком уровне с памятью, адресами.

***Но также есть и минусы:***

* Синтаксис, провоцирующий ошибки;
* Некоторые преобразования типов не интуитивны;
* Иногда шаблоны приводят к порождению кода очень большого объёма;
* Отсутствие встроенной системы сборки мусора.

***Как альтернативный язык программирования рассмотрим Java. У него также есть некоторые плюсы:***

* Кроссплатформенность. Для запуска Java-приложения достаточно установить виртуальную Java-машину (JVM);
* Многофункциональные стандартные библиотеки;
* Встроенный сборщик мусора;
* Встроенные библиотеки для работы с сетью.

***Из недостатков можно назвать:***

* Низкое, в сравнении с другими языками, быстродействие, повышенные требования к объему оперативной памяти;
* Постоянное развитие языка вызывает наличие как устаревших, так и новых средств, имеющих одно и то же функциональное назначение;
* Java является полностью объектно-ориентированным языком. Например, C++ тоже является объектно-ориентированным, но в нем есть возможность писать программы не в объектно-ориентированном стиле, а в Java так нельзя;
* Язык реализован с использованием интерпретации байт-кода. То есть программа сначала транслируется в машинно-независимый Р-код, а потом интерпретируется некоторой программой-интерпретатором (виртуальная Java-машина, JVM).

***Также можно рассмотреть язык программирования Python. Его плюсы:***

* Python – универсальный язык, простой и интуитивно-понятный. Кривая обучения у него сравнительно пологая, на этом языке возможно быстрее писать программы;
* В Python встроен фреймворк для тестирования, входной барьер у которого очень низок. Фреймворк обеспечивает хорошее тестовое покрытие;
* Python – простой и распространенный язык, не только понятный многим программистам, но и легкий для специалистов по статистике, на нем можно написать такой инструмент, в котором будут интегрированы все этапы рабочего процесса.

***Его минусы:***

* Хотя в Python и есть «приятные» библиотеки, визуализация тут устроена гораздо сложнее, а ее результаты порой не слишком наглядны;
* Для выполнения скриптов требуется много ресурсов;
* Отступы могут производиться как за счёт пробелов, так и за счёт табуляции. При совместной разработке это вызывает проблемы.

Подведем итоги. Для нашего проекта лучше всего использовать язык программирования C++, так как этот язык является достаточно гибким, на нём можно написать оптимальную программу и качественно её визуализировать.

## Выбор используемых библиотек

Для данного проекта было решено использовать следующие библиотеки: Boost, PicoSHA2 и Qt-библиотеки.

Boost — собрание библиотек классов, использующих функциональность языка C++ и предоставляющих удобный кроссплатформенный высокоуровневый интерфейс для лаконичного кодирования различных повседневных подзадач программирования (работа с данными, алгоритмами, файлами, потоками и т. п.). Использовалась библиотека boost::filesystem для отслеживания директорий и файлов. Также P2P сеть писалась именно при использовании boost::asio.

Библиотека PicoSHA2 была использована для хэширования. А именно однонаправленная хэш-функция SHA256. Этот алгоритм отличается высокой скоростью шифрования, не критическим риском коллизий (но проигрывает в этом параметре некоторым другим хэш-функция, к примеру MD5). В случае майнинга есть риск централизации добычи крупными майнерами. SHA256 используют такие криптовалюты, как Bitcoin и все его форки, Peercoin, Namecoin, Terracoin, частично в Litecoin’е и других. Для нашего проекта эта функция отлично подходит, так как она надежная, зарекомендовавшая себя в blockchain субкультуре и быстрая (чтобы при тестировании проекта не приходилось ждать несколько пар майнинг одного блока).

Qt-библиотеки использовались нами для написания GUI в Qt Creator’е и только для этого.

# Описание технический решений

Во время выполнения данного проекта, как и любого другого проекта, возникали некоторые трудности. Каждую из них нам удавалось в кратчайшие сроки без особых проблем преодолеть. Каждую, кроме одной. О ней хочу написать чуть поподробнее.

Главной проблемой, ставшей перед нами, стала пиринговая сеть. Мы пробовали разные реализации, варианты, библиотеки и языки программирования. В конечном итоге удовлетворяющий нас результат был достигнут благодаря библиотеке boost::asio.

# Заключение

Во время выполнения данной курсовой работы мы поняли, что блокчейн обеспечивает неизменность занесённых в него данных, но не может обеспечить их истинность. Единственное исключение — случай, когда блокчейн не пытается заглядывать в физический мир. Тогда вся необходимая информация есть внутри блокчейна, и система может верифицировать корректность данных: например, что на адресе достаточно средств для совершения конкретной транзакции.

Именно это является главной проблемой любого проекта на blockchain’е, не связанного с криптовалютой. И пока это проблема не будет решена (что маловероятно произойдет в ближайшем будущем), такие проекты не получат общественной поддержки. Потому что они становятся не такими уж и надежным от разного рода мошенничества. А надежность в данном вопросе – главная черта данной технологии. Поэтому я считаю, что бессмысленно сейчас инвестировать деньги в блокчейн проекты: рынок криптовалют перенасышен, а другие сервисы имею большой недостаток, который я описал выше.

# Список используемых источников

1. Книга «Mastering Bitcoin»: <https://github.com/bitcoinbook/bitcoinbook> (дата обращения: 01.09.2019)
2. Документация библиотеки boost: <https://www.boost.org/doc/> (дата обращения: 11.09.2019)
3. Библиотека PicoSHA2: <https://github.com/okdshin/PicoSHA2> (дата обращения: 13.09.2019)
4. Qt Creator Manual: <https://doc.qt.io/qtcreator/> (дата обращения: 27.11.2019)
5. Понятие о boost для C++: <https://ru.wikipedia.org/wiki/Boost> (дата обращения: 13.12.2019)
6. Преимущества и недостатки хэш-функции SHA256: <https://ico-miner.info/blokchejn/algoritm-shifrovaniya-sha-256-osobennosti-preimushhestva-i-nedostatki-majning/> (дата обращения: 13.12.2019)
7. Плюсы и минусы языка программирования C++. Доступно на: <http://cpplus.my1.ru/publ/o_jazyke_c/dostoinstva_i_nedostatki_jazyka/1-1-0-6> (дата обращения: 13.12.2019)
8. Сравнение языков программирования R и Python. Доступно на: <https://habrahabr.ru/company/piter/blog/263457/> (дата обращения: 13.12.2019)
9. Недостатки Python и их описание. Доступно на: <http://python-3.ru/page/nedostatki-python-i-ih-opisanie> (дата обращения: 13.12.2019)
10. Статься о сервисах регистрации авторского права: <https://bitnewstoday.ru/news/blokcheyn-kak-tsifrovoe-dokazatelstvo-v-sude-sila-gigantov-ernst-young-i-microsoft/> (дата обращения: 13.12.2019)
11. Статься о несостоятельности блокчейн проектов (кроме крипты): <https://habr.com/ru/company/solarsecurity/blog/438028/> (дата обращения: 13.12.2019)