

Министерство образования Республики Беларусь  
Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет инженерно-экономический  
Кафедра экономической информатики  
Дисциплина «Средства и технологии анализа и разработки информационных  
систем»

«К ЗАЩИТЕ ДОПУСТИТЬ»

Руководитель курсового проекта

Старший преподаватель

\_\_\_\_\_.Петрович Н.О.

\_\_\_\_\_.\_\_\_\_\_.2022

## **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовому проекту

на тему:

**«ПРОГРАММНАЯ ПОДДЕРЖКА ПОИСКА ИНВЕСТИЦИОННЫХ  
ПРОЕКТОВ»**

БГУИР КП 1-40 01 02-08 015 ПЗ

Выполнил студент группы 972303  
Хомякова А.А.

\_\_\_\_\_  
(подпись студента)

Курсовой проект представлен на  
проверку \_\_\_\_\_.\_\_\_\_\_.2022

\_\_\_\_\_  
(подпись студента)

Минск 2022

## РЕФЕРАТ

ПРОГРАММНАЯ ПОДДЕРЖКА ПОИСКА ИНВЕСТИЦИОННЫХ ПРОЕКТОВ: курсовой проект / А.А. Хомякова – Минск: БГУИР, 2022, – п.з. – с.

Ключевые слова: инвестирование, краудфандинг, соискатель, выгода.

Объект исследования: процесс инвестирования стартапов.

Предмет исследования: оптимизация быстрого поиска инвестиций для стартапов и поиска бизнес-идей для инвестирования.

Цель курсового проекта: ускорение и упрощение процесса поиска средств для запуска стартапов со стороны стартаперов и поиска лучших бизнес-идей для вложения средств инвестора.

Методология проведения работы: в процессе разработки системы использованы методы анализа, классификации, обобщения данных, функциональный анализ процессов, принципы построения баз данных, моделирование системы с помощью UML-диаграмм, разработана программная и графическая составляющие программного средства.

Результаты работы: разработана онлайн платформа для краудфандинга.

Область применения результатов: бизнес-инкубаторы, фонды развития.

## Содержание

Перечень условных обозначений.....	2
Введение .....	5
1 Анализ литературных исследований и программных решений .....	7
1.1 Описание и анализ предметной области .....	7
1.2 Сравнительный обзор аналогов программного средства .....	8
1.3 Обоснование и описание выбора языка программирования, средств разработки, используемых технологий и сторонних библиотек .....	14
2 Моделирование предметной области и разработка требований к программному средству .....	24
2.1 Анализ и формализация бизнес-процессов предметной области .....	24
2.2 Анализ требований к разрабатываемому программному средству. Спецификация функциональных требований .....	25
3 Проектирование и разработка программного средства .....	34
3.1 Проектирование пользовательского интерфейса .....	24
3.2 Проектирование схемы информационной модели предметной области .....	24
3.3 Архитектурные решения .....	24
3.4 Описание динамических аспектов поведения объектов системы .....	24
3.5 Описание и разработка алгоритмов, реализующих бизнес-логику разрабатываемого программного средства .....	24
4 Тестирование и проверка работоспособности программного средства .....	36
5 Руководство по установке (развертыванию) и использованию программного средства .....	36
5.1 Руководство по установке (развертыванию) программного средства .....	24
5.2 Руководство пользователя .....	24
Заключение .....	48
Список использованных источников .....	49
Приложение А .....	50
Приложение Б .....	51
Приложение В .....	53

## ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, ТЕРМИНОВ

*Краудфандинг* (от англ. *crowdfunding*: *crowd* — «толпа», *funding* — «финансирование») — это способ коллективного финансирования проектов, при котором деньги на создание нового продукта поступают от пользователей, получающих взамен какие-либо товары или услуги, в том числе, итоговый продукт.

*Бизнес-идея/крауд-проект/крауд-кампания* — идея, которую планируется реализовать на средства, полученные с помощью краудфандинга.

*Стартап* — молодая быстро развивающаяся компания, в основе которой лежит инновационная бизнес-идея и технология.

*Краудфандинговая платформа/площадка* — специализированный интернет-ресурс, на котором размещаются крауд-проекты. Существуют как многопрофильные, так и узкоспециальные площадки, допускающие только проекты конкретных категорий.

*Стартапер/автор бизнес-идеи* — лицо, представляющее свою идею на краудфандинговой платформе.

*Акции* — бонусы, которые автор предлагает пользователям в качестве вознаграждения за вклад в проект. Эти бонусы могут быть материальными и нематериальными.

*Инвестор* — лицо или организация, размещающие капитал с целью последующего получения прибыли.

*Программа* — данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определённого алгоритма.

*Программное средство* — объект, состоящий из программ, процедур, правил, а также, если предусмотрено, сопутствующих им документации и данных, относящихся к функционированию системы обработки информации.

*БД* – База данных.

*Язык программирования* — формальная знаковая система, предназначенная для записи компьютерных программ.

*Фреймворк* — программная платформа, определяющая структуру программной системы, облегчающая разработку и объединение разных компонентов большого программного проекта.

## ВВЕДЕНИЕ

В последнее время в мире активно внедряются альтернативные способы предоставления финансовых услуг, получившие обобщённое название финансовые технологии. Они охватывают такие направления, как платежи и переводы, аутентификацию и идентификацию пользователей, инвестиционный и риск-менеджмент, управление личными финансами и кредитование, банковское дело и страхование. Одним из финтех-сервисов, получивших значительное распространение во всем мире, является краудфандинг, под которым подразумевается привлечение денежных средств посредством специализированной интернет-платформы. К настоящему моменту она имеет множество модификаций, а объёмы привлечённых средств с её использованием ежегодно увеличиваются.

Возникают новые формы привлечения финансовых ресурсов для претворения в жизнь различных бизнес-проектов, к которым можно отнести и краудфандинг. Краудфандинг представляет собой способ финансирования коммерческих и некоммерческих проектов, который реализуется посредством сбора финансовых ресурсов от широкого круга людей преимущественно с использованием глобальной компьютерной сети Интернет. Стоит отметить, что легального определения краудфандинга, а также особенностей правового регулирования данного инструмента национальным законодательством не предусматривается. Однако анализ Государственной программы инновационного развития Республики Беларусь на 2016–2020 годы позволяет сделать вывод о том, что краудфандинг, по законодательству Республики Беларусь, понимается как интернет-платформа для взаимодействия потребителей и производителей товаров и услуг, инвесторов и соискателей инвестиций[1].

Идея финансирования каких-либо проектов за счёт сбора небольших сумм денежных средств посредством привлечения большого количества людей не нова. Однако изобретение интернета и его доступность для большого количества пользователей позволили оптимизировать издержки на поиск и финансирование проектов посредством появления сайтов-посредников, размещающих информацию о проведении сбора средств и привлекающих заинтересованных лиц.

Первопричиной роста популярности краудфандинга считается мировой финансовый кризис 2008 года, который привёл к ограничениям в использовании традиционных инструментов финансирования организации

малого и среднего бизнеса, а также стартапов. Это создало недостаток финансирования, заполнить который смог краудфандинг.

Информационно-коммуникационные технологии также способствовали его распространению: широкое использование информационных систем обеспечило инфраструктуру для охвата миллионов пользователей — потенциальных инвесторов и спонсоров; использование технологий, обеспечивающих поддержку социальных сетей, позволило пользователям взаимодействовать в режиме онлайн.

В общем виде краудфандинг рассматривается как способ привлечения через интернет-платформы денежных средств физических и юридических лиц для достижения различных целей без посредничества традиционных институтов или профессиональных участников финансового рынка.

В современном мире всё чаще становится актуальным вопрос о разработке разнообразных приложений, которые сократят время пользователя на удовлетворение его запросов и потребностей в интерактивном режиме, а также внедрения их в жизнедеятельность человека.

Основной целью данной курсовой работы является создание платформы для поиска инвестиций для бизнес-идей[2].

Поставленная цель потребовала решения следующих задач:

- исследовать процесс сбора средств для запуска стартапа;
- изучить предметную область;
- детально описать стадии основного процесса предметной области;
- построить информационную модель системы;
- ознакомиться с особенностями организации потоков инвестирования;
- исследовать особенности выбора стартапов для финансирования;
- разработать программную платформу для удовлетворения потребностей всех участников бизнес-процесса;
- описать алгоритмы, реализующие бизнес-логику;
- проверить разработанный программный продукт на соответствие всем заявленным требованиям.

Объектом исследования является сбор инвестиций на реализацию бизнес-идей.

# 1 АНАЛИЗ И МОДЕЛИРОВАНИЕ СИСТЕМЫ ПОИСКА ИНВЕСТИЦИОННЫХ ПРОЕКТОВ

## 1.1 Описание и анализ предметной области

Цифровая трансформация является важнейшей тенденцией экономического развития, которая касается всех сфер и отраслей. Развитие цифровых технологий привело к появлению такого явления как краудфандинга — способ привлечения денежных средств компаниями от большой группы инвесторов через электронные системы посредников — краудфандинговые платформы.

В отличие от традиционных методов инвестирования, краудфандинговые кампании собирают средства у широкой публики. Обычно наиболее успешные проекты получают около 40 % своего дохода от людей, с которыми автор каким-либо образом связан: семья, друзья, коллеги по работе, знакомые. Затем кампания получает поддержку от незнакомых людей. Общая схема прогресса кампании изображена на графике (рисунок 1.1).

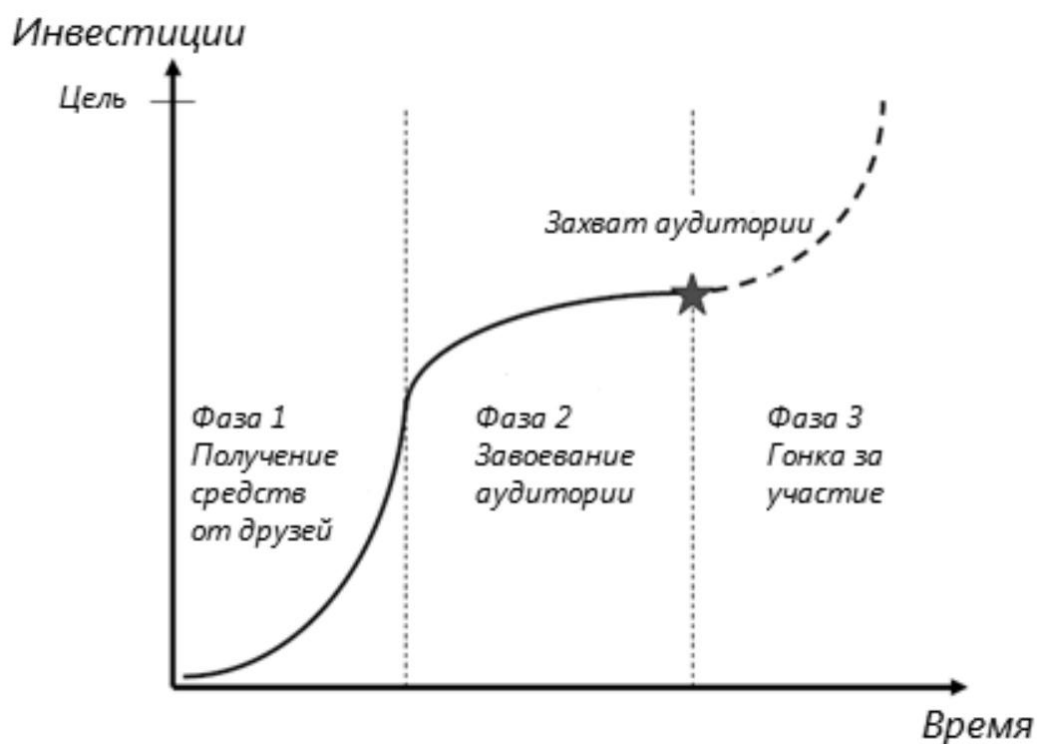


Рисунок 1.1 – График прогресса краудфандинговой кампании

На графике представлена кривая, отражающая процесс накопления вкладов во времени. Как видим, данный процесс имеет три фазы. Первая фаза имеет название «Получение средств от друзей». Она характеризуется стремительным и довольно значимым потоком инвестиций, который в сумме примерно достигает половины целевой. Такое быстрое накопление обусловлено тем, что на данном этапе инвестируют в первую очередь люди, непосредственно связанные с проектом и его автором. Однако при завершении начальной фазы наступает следующая, от прохождения которой зависит, достигнет ли проект конечной цели.

Первая фаза имеет название «Получение средств от друзей». Она характеризуется стремительным и довольно значимым потоком инвестиций, который в сумме примерно достигает половины целевой. Такое быстрое накопление обусловлено тем, что на данном этапе инвестируют в первую очередь люди, непосредственно связанные с проектом и его автором. Однако при завершении начальной фазы наступает следующая, от прохождения которой зависит, достигнет ли проект конечной цели.

Вторая фаза — «Завоевание аудитории» характеризуется замедлением темпа роста накопления инвестиций. Чтобы проект прошёл данную стадию, требуется стимулировать получение инвестиций путём мотивации и вовлечения в проект других инвесторов через общение. Большое значение здесь имеет общение потенциальных спонсоров между собой. Автор проекта должен понимать, что его ответственность за «выживание» проекта на данной фазе очень высока. Только немногие проекты достигают точки «Захват аудитории», которая даёт толчок цепной реакции и вызывает быстрый рост получения вложений. Инвесторами на данном этапе обычно являются люди, не имевшие прямой связи с проектом, но которые узнали и заинтересовались им.

После точки «Захват аудитории» начинается фаза «Гонка за участие». На данной фазе заинтересованные инвесторы становятся более решительными и спешат принять участие в проекте, пока его срок не истёк и цель ещё не достигнута. Если проект завершён успешно, т. е. необходимая сумма собрана за установленный срок, она отправляется автору, при этом краудфандинговая платформа может взять небольшой процент с этой суммы. Если проект не успел собрать нужную сумму в срок, средства возвращаются спонсорам[3].

В модели краудфандинга в общем смысле присутствуют три участника: автор проекта, потенциальные инвесторы и краудфандинговая платформа. Первые — это авторы идеи или проекта, который необходимо



профинансировать для осуществления его реализации. Эти люди хотят воспользоваться краудфандингом, чтобы получить прямой доступ к рынку и принять финансовую поддержку от действительно заинтересованных личностей. Вторые — это потенциальные инвесторы, которые решили поддержать финансово предложенные проекты, несут риск и чаще всего ожидают определённое вознаграждение. Выбрав наиболее обещающий и интересный проект, они становятся соучастниками реализации проекта.

Существуют также дополнительные участники, которые оказывают услуги основным акторам на платной основе или безвозмездно. Анализ функционирования платформ в различных странах позволил составить перечень предоставляемых услуг, которые представлены на рисунке 1.2.

Перечень услуг, предоставляемых основным участникам краудфандинга				
Наименование оказываемой услуги	Используемая бизнес-модель краудфандинга	Актор предоставляемой услуги	Вид услуги по характеру предоставления	Провайдер услуги
Перевод денежных средств лиц, поддержавших крауд-проект, на транзитные или условные (в т. ч. эскроу) счета, зачисление денег на счет инициатора проекта или их возврат спонсорам в случае, если не удалось собрать необходимую сумму	Любая бизнес-модель краудфандинга	Краудфандинговая платформа	Платная, удерживается комиссия при перечислении денег инициаторам проекта	Банки, платежные системы (агрегаторы, поставщики платежных услуг, провайдеры платежных сервисов), эскроу-агенты
Организация и проведение краудкампании	Наградной краудфандинг	Инициаторы краудкампании	Платная, комиссия удерживается в случае успешного сбора средств	Маркетинговые, краудфандинговые и PR-агентства
Предоставление отчетов и аналитических инструментов, позволяющих проводить мониторинг и анализ эффективности сбора денежных средств	Любая, чаще – наградной краудфандинг	Инициаторы краудкампании	Зачастую услуга предоставляется бесплатно или условно бесплатно, а также в одном пакете с платными услугами	IT-компании
Юридические консультации (о соблюдении авторских прав, законности проводимых сделок)	Любая бизнес-модель краудфандинга	Любой базовый участник экосистемы	Платная	Юридические фирмы, индивидуальные предприниматели
Рекомендация краудфандинговой платформы в качестве посредника субъектам малого и среднего бизнеса (в т. ч. стартапам)	Любая бизнес-модель краудфандинга	Краудфандинговая платформа	Бесплатная	Венчурные фонды, банки, государственные органы исполнительной власти
Проведение процедуры дью дилидженс	Любая, чаще – краудинвестинг	Краудфандинговая платформа	Платная	Провайдер услуг дью дилидженс
Регистрация и передача ценных бумаг от эмитента инвесторам	Краудинвестинг	Краудфандинговая платформа	Платная	Агент по передаче и регистрации ценных бумаг

Рисунок 1.2 – Перечень услуг

Важную роль в экосистеме краудфандинга играют и традиционные участники финансового рынка, а также местные органы власти, которые направляют обратившихся к ним за финансированием субъектов малого и среднего бизнеса на краудфандинговые платформы. Целью венчурных

компаний обычно выступает определение спроса на предлагаемые товары или услуги в бизнес-модели предзаказа, а также изучение наличия интереса у инвесторов к разрабатываемой продукции. В случае несоответствия требованиям, которые предъявляют кредитные институты, а также при низком уровне кредитоспособности банки направляют субъекты малого и среднего бизнеса на краудлендинговую платформу. Исполнительные органы власти могут заключать договор с платформой о софинансировании проектов при условии, если их инициаторы сумеют привлечь фиксированную долю финансовых ресурсов с помощью краудфандинга.

Также важной составляющей в экосистеме краудфандинга является государственное регулирование как самого процесса привлечения денежных средств, так и деятельности всех основных и дополнительных акторов, что особенно необходимо в краудинвестинге и краудлендинге, поскольку затрагивает финансовые интересы участников, предоставляющих свои средства в пользование[4].

Преимуществ запуска проекта на специальной платформе много. Ниже приведены самые весомые аргументы в пользу использования стороннего сервиса для поиска инвестиций:

- выходя за рамки собственных ресурсов, вы расширяете аудиторию проекта. Пользователи крауд-платформ — это миллионы человек, уже знакомых с принципом народного финансирования. Обращаясь к ним за поддержкой, вы значительно увеличиваете охват людей, готовых поддержать вас информационно и финансово;

- также специализированные крауд-платформы предоставляют расширенные возможности оплаты. Акционеры могут оплатить покупку любой акции несколькими способами: банковская карта, со счета мобильного телефона, через различные финансовые сервисы. При сборе средств на своём сайте автору придётся заняться этим вопросом самостоятельно. Обеспечение удобных вариантов перечисления денег — залог успешного сбора средств. Определённо, на крауд-платформах он будет работать более точно и прозрачно, чем универсальные платёжные системы, добавленные на свой сайт;

- также важный фактор в пользу платформ — это встроенная система сбора статистики проекта и информации об инвесторах. Удобный способ сбора данных по всей базе ваших сторонников. Данные могут содержать почту для удобной коммуникации. В случае с запуском крауд-кампании на собственном сайте вы почувствуете недостаток в автоматизированном сборе данных, как только у вас появится первый инвестор. В итоге ручной сбор

информации может занять недели. Особенно, если количество инвесторов измеряется сотнями;

– размещение проекта на крауд-платформе является дополнительной гарантией его надёжности. Это связано с тем, что до старта крауд-кампания проходит проверку, а с её автором платформа заключает договор.

Для определения цены краудфандинга рассчитывается размер финансовой цели. Финансовая цель — это сумма средств, которые необходимо собрать в течение установленного срока на краудфандинговой платформе. Финансовая цель должна включать необходимые для реализации проекта средства, комиссию краудфандинговой платформы, сбор платёжной системы, налог. То есть на краудфандинговой платформе нужно собрать такую сумму средств, чтобы после вычета из неё комиссии краудфандинговой платформы, сбора платёжной системы и налога осталась сумма средств, которые необходимы непосредственно на реализацию проекта. Схема финансирования представлена на рисунке 1.3.



Рисунок 1.3 – Схема организации краудфандингового финансирования

Проанализировав суть краудфандинга возникает вопрос: как представить проект на краудфандинговой платформе и оценить этот метод финансирования. Рассмотрим методические основы экономической оценки организации краудфандинга в целях использования его как источника финансирования некоего проекта. Предлагаемая методика включает следующие этапы:

- составление бизнес-плана;
- оценка рациональности использования краудфандинга;

- представление проекта на краудфандинговой платформе;
- оценка выгоды использования краудфандинга в сравнении с альтернативным источником финансирования;
- оценка риска при организации краудфандинговой кампании[5].

Таким образом, привлечение денежных средств посредством использования краудфандинговых площадок стремительно набирает популярность среди белорусов, поэтому для достижения наибольшего прогресса в данной области необходимо на законодательном уровне разработать механизм регулирования общественных отношений в сфере краудфандинга в целом, а не только его отдельных видов.

## **1.2 Разработка функциональной модели бизнес-процессов платформы для поиска инвестиций**

Исходя из задания данного курсового проекта, целью моделирования системы является описание её функциональности для дальнейшего использования созданной модели при разработке информационной модели.

После проведённого анализа предметной области представим функциональную модель работы платформы для поиска бизнес-идей для их финансирования. Детали анализа данного процесса создавались на основе рассмотрения бизнес-процесса со всех заинтересованных сторон, а именно: инвестор, соискатель инвестиций.

Методология IDEF0 предписывает построение иерархической системы диаграмм — единичных описаний фрагментов системы. Сначала проводится описание системы в целом и её взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция — система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности.

Каждая IDEF0-диаграмма содержит блоки и дуги. Блоки изображают функции моделируемой системы. Дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними.

Функциональные блоки на диаграммах изображаются прямоугольниками, означаями поименованные процессы, функции или задачи, которые происходят в течение определённого времени и имеют распознаваемые результаты. Имя работы должно быть выражено отглагольным существительным, обозначающим действие.

IDEF0 требует, чтобы в диаграмме было не менее двух и не более шести блоков. Эти ограничения поддерживают сложность диаграмм и модели на уровне, доступном для чтения, понимания и использования[6].

Каждая сторона блока имеет особое, вполне определённое назначение. Левая сторона блока предназначена для входов, верхняя – для управления, правая – для выходов, нижняя — для механизмов. Такое обозначение отражает определённые системные принципы: входы преобразуются в выходы, управление ограничивает или предписывает условия выполнения преобразований, механизмы показывают, что и как выполняет функция.

Итак, на рисунке 1.4 представлена контекстная диаграмма верхнего уровня процесса «Найти финансирование для стартапа». Входная информация, в данном случае — это бизнес-идея и данные от пользователей системы. В результате процесса на выходе получаем отчёт о сборе денежных средств.

В качестве ресурсов для поиска инвестиционных проектов выступает программное обеспечение, создание которого является целью данного проекта, а также человеческие ресурсы инвестора и соискателя средств для развития стартапа.

В управлении процессом поиска участвуют сроки поиска финансирования и правовые акты, существующие в рамках использования создаваемой интернет-платформы.

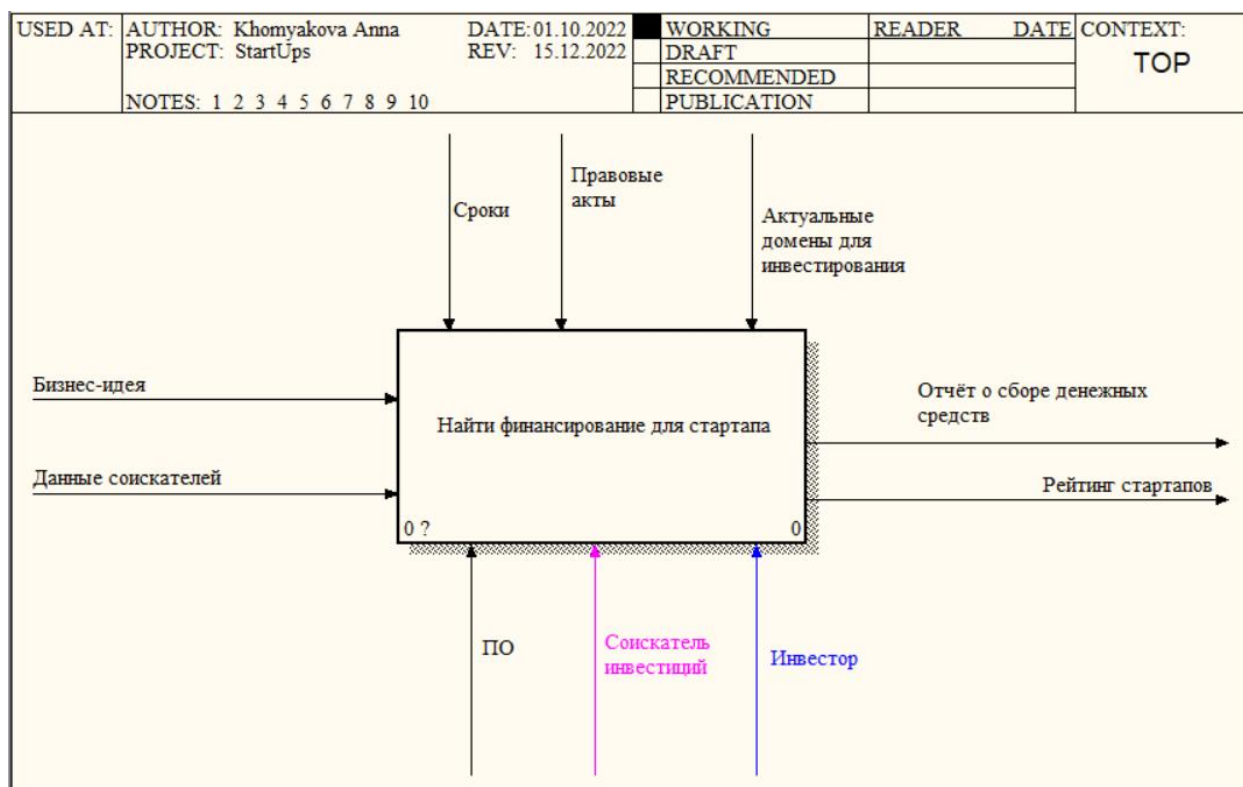


Рисунок 1.4 – Контекстная диаграмма верхнего уровня

Декомпозиция контекстной диаграммы, показанная на рисунке 1.5, описывает разделение основного процесса на компоненты-задачи. Процесс поиска инвестирования в общем виде можно разделить на следующие этапы:

- пройти регистрацию в системе;
- выбрать бизнес-идею для инвестирования;
- провести анализ результатов.

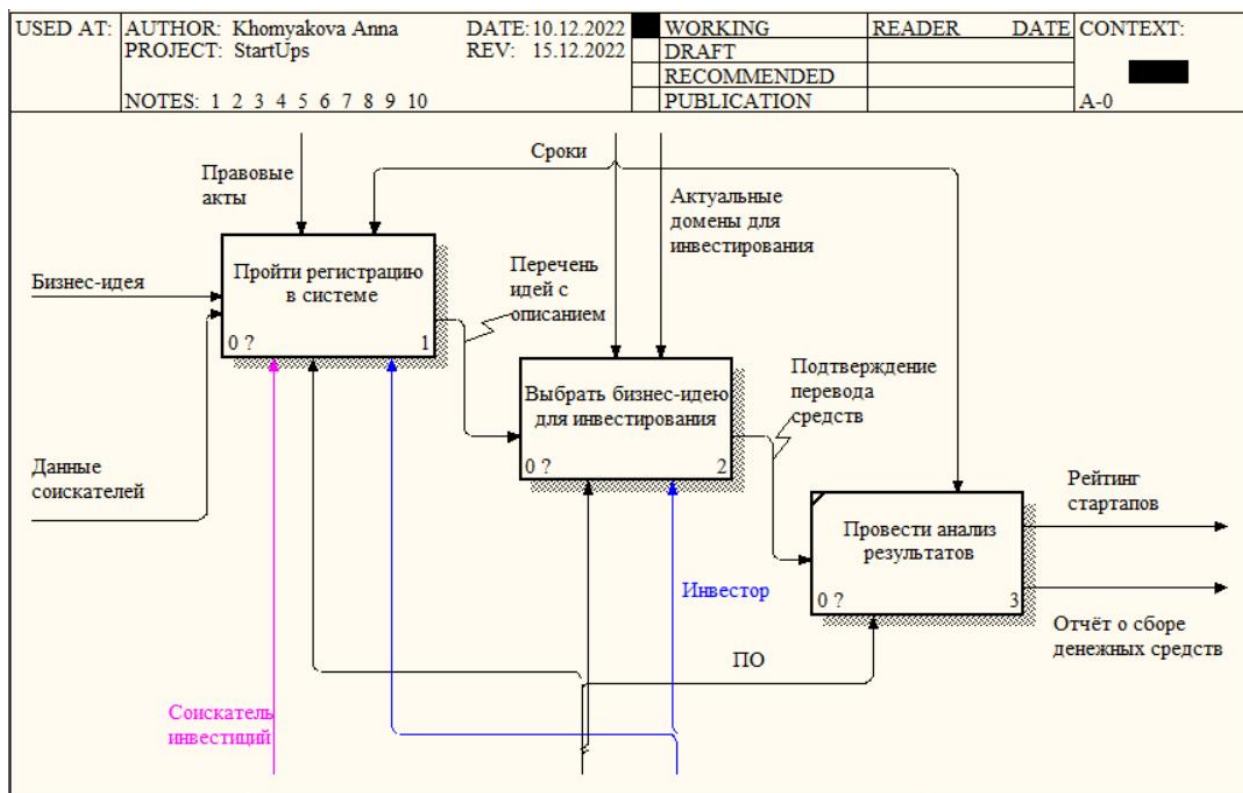


Рисунок 1.5 – Декомпозиция контекстной диаграммы верхнего уровня

При прохождении регистрации ограничителями будут выступать правовые акты стран, жители которых взаимодействуют с системой. После успешной регистрации всех сторон выходным результатом является перечень бизнес-идей, а после того, как инвесторы сделают свой выбор, получим подтверждение о переводе денежных средств.

Проведём декомпозицию блока «Пройти регистрацию в системе» для более детального изучения процесса (рисунок 1.6).

На данном этапе выполняется следующие процессы:

- осуществить регистрацию соискателей инвестиций;
- осуществить регистрацию стартапа;
- осуществить регистрацию соискателей бизнес-идей;



- получить доступ к бизнес-идеям.

В ходе данного этапа каждый процесс важен для полноценной работы с платформой поиска бизнес-идей, поэтому корректное создание учётных данных важно для высокой производительности.

Регистрация контролируется сроками активации платной подписки и правовыми актами использования собственных данных.

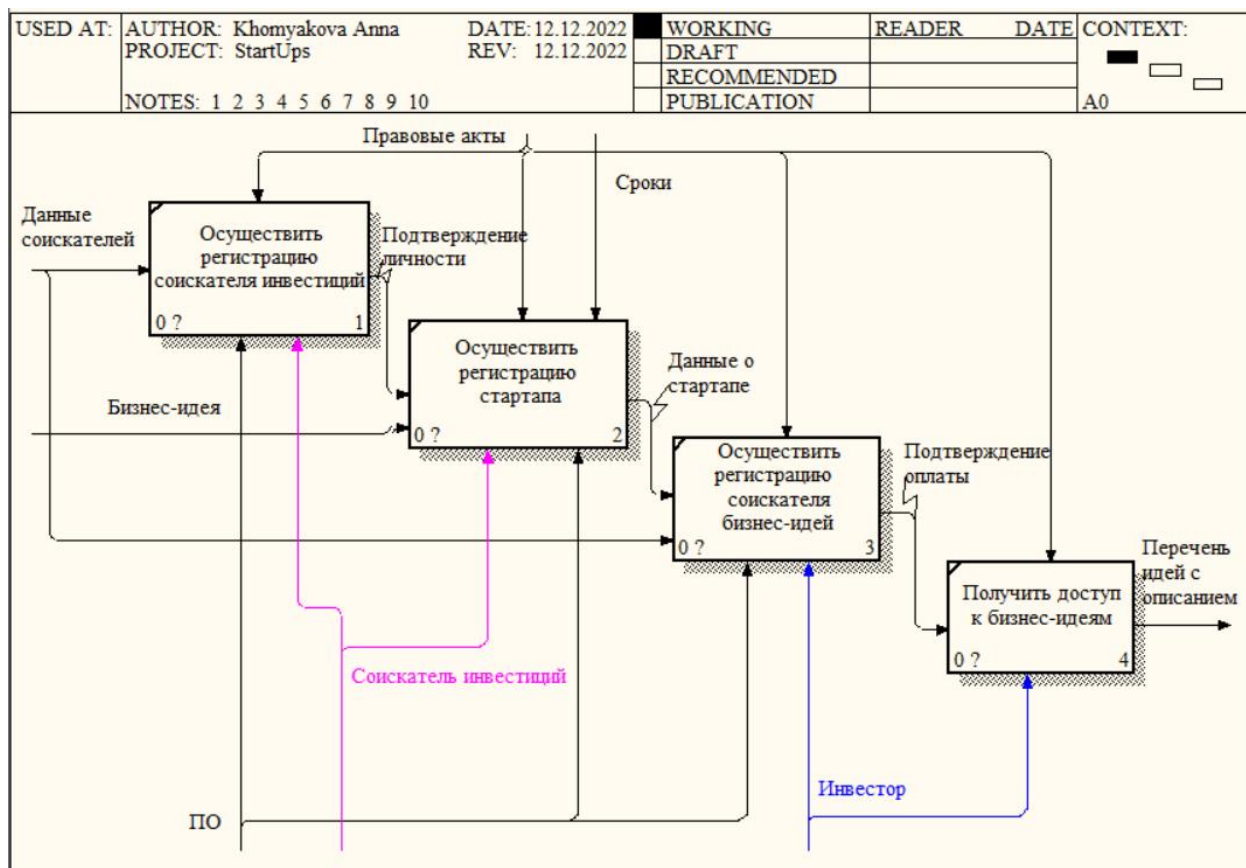


Рисунок 1.6 – Декомпозиция блока «Пройти регистрацию в системе»

Также проведём декомпозицию блока «Осуществить регистрацию соискателя бизнес-идей» (рисунок 1.7).

На данном этапе выполняется следующие этапы:

- создание личного кабинета;
- ввести данные банковского счёта;
- оплатить подписку.

После того, как ПО получит на вход вводимые персональные данные от инвестора, система анализирует корректность вводимых данных и в случае отсутствия ошибок, создаёт личный кабинет клиента платформы.

Затем происходит оплата платной подписки для доступа ко всем бизнес-идеям. Причём обработка персональных банковских данных должна производиться в соответствии с правовыми актами и должна быть защищена от кражи данных.

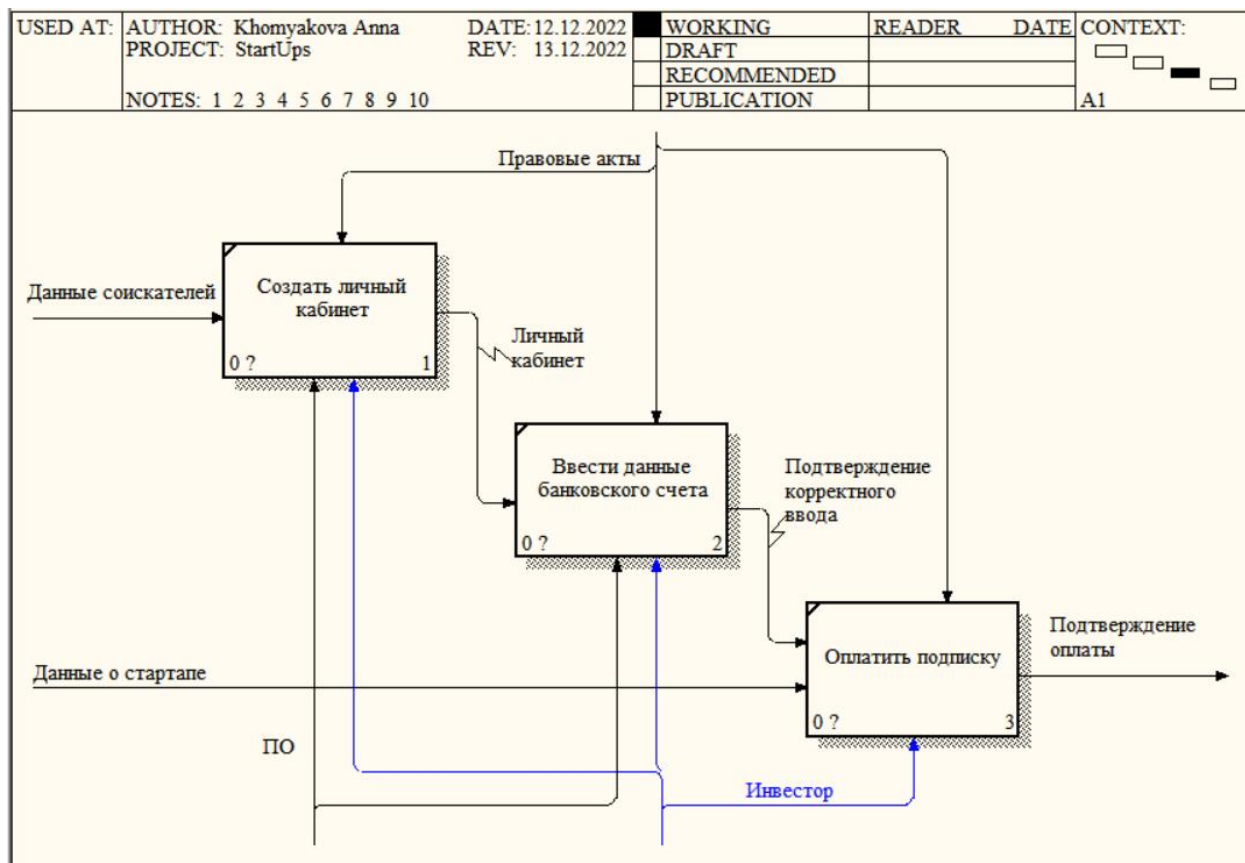


Рисунок 1.8 – Декомпозиция блока «Осуществить регистрацию соискателя бизнес-идей»

Проведём декомпозицию блока «Выбрать бизнес-идею» (рисунок 1.9). Данный процесс является ключевым в отношении сторон, участвующих в краудфандинговой системе. Рассмотрим его более детально. Данная декомпозиция будет включать следующие этапы:

- отсортировать идеи по критерию;
- провести сравнение идей;
- ознакомиться с описанием и комментариями;
- инвестировать в стартап.

После того, как программное обеспечение отсортирует бизнес-идеи по критерию, заданному соискателем, происходит этап их сравнения в виде



таблиц. После процессов сортировки и сравнения происходит анализ комментариев инвестором для выявления конкурентоспособных лидеров.

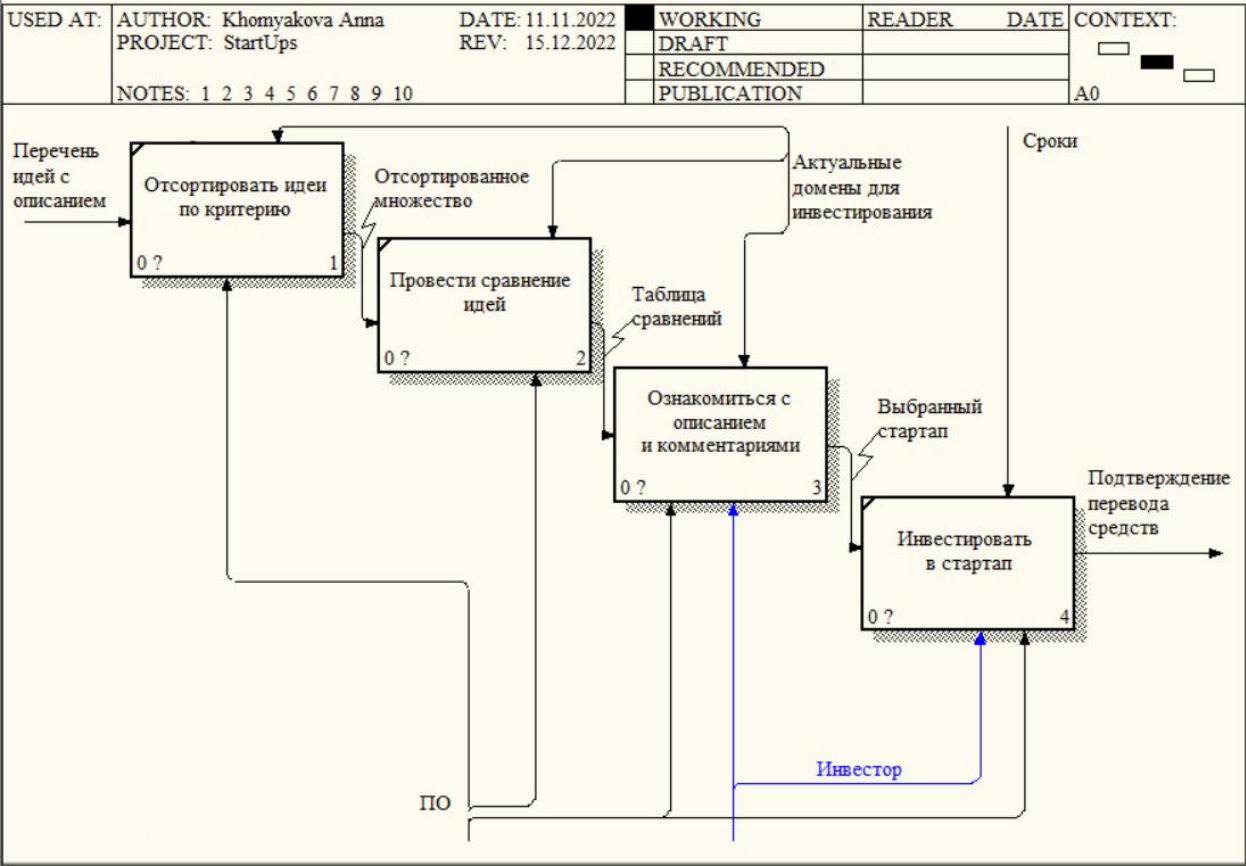


Рисунок 1.9 – Декомпозиция блока «Выбрать бизнес-идею»

Таким образом модель TO-BE диаграммы IDEF0 даёт чёткое представление о необходимости внедрения программного обеспечения в процесс сбора инвестиций для развития бизнес-идей.

## **2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СИСТЕМЫ ПОДДЕРЖКИ ИНВЕСТИЦИОННЫХ ПРОЕКТОВ**

### **2.1 Постановка задачи на проектирование системы поддержки инвестиционных проектов**

Для достижения цели были поставлены и решены следующие задачи:

- описать будущий процесс поиска инвестирования бизнес-идей с помощью IDEF0;
- разработать UML-диаграммы для анализа аспектов создаваемой системы;
- отобразить информационную модель системы поддержки инвестиционных проектов;
- визуализировать алгоритм процесса будущего программного средства с помощью схемы;
- подготовить инструкцию для развёртывания и использования системы.

С технической стороны требовалась реализация следующих функций:

- регистрация пользователей в соответствии с их ролью;
- возможность администрирования платформы;
- сохранение конфиденциальности данных;
- регистрация бизнес-идей;
- рекламная кампания бизнес-идей;
- поиск эффективных бизнес-идей для финансирования;
- подписка на еженедельные обновления;
- просмотр данных о разработчиках, стартапах, новостях
- реализация платной подписки на платформу с дополнительными возможностями;
- ведение учёта комментариев и лайков;
- создание резюме для участия в развитии стартапа;
- возможность принять или отклонить резюме участника разработки;
- сравнение, фильтрация и сортировка бизнес-идей;
- удаление бизнес-идей по истечении срока давности;
- при обнаружении проблем, заполнение формы обратной связи;
- инвестирование средств в проект и отслеживание прогресса;
- создание графических отчётов;
- создание stripe-аккаунта для оплаты.

## **2.2 Модели представления системы и её описание**

В данном разделе будет продемонстрировано моделирование информационной системы с помощью стандарта UML, который использует графические обозначения для создания абстрактной модели системы и предназначен для определения, визуализации, проектирования и

документирования в основном программных систем. UML позволяет описать систему практически со всех возможных точек зрения и разные аспекты поведения системы.

Для данного дипломного проекта были построены следующие диаграммы: диаграмма вариантов использования, последовательности, состояний, развёртывания и компонентов.

**2.2.1** Диаграмма вариантов использования состоит из актёров, для которых система производит действие и собственно действия Use Case, которое описывает то, что актёр хочет получить от системы.

В данной диаграмме вариантов использования в роли актёров выступают администратор краудфандинговой платформы, стартапер (владелец бизнес-идеи), инвестор, разработчик. К приватным операциям администрирования принадлежит управление новостным каналом, форумом, настройка срока, после которого стартап исчезает из рейтинга, также отслеживание аккаунтов с платной подпиской, одобрение регистрации идей. стартапер создаёт идеи, имеет возможность продвижения идей (только при наличии платной подписки), может нанять разработчика или отклонить его запрос, получить отчёт на почту, просмотреть графический отчёт о финансировании, также ему доступны операции для выбора идей. Общими кейсами для стартапера, разработчика и инвестора является ведение обсуждений на форуме, написание комментариев. Разработчик имеет возможность публикации резюме, поиска позиций на стартапах, подачи на стартап, также может оценить идею вместе с инвестором. Данная диаграмма представлена в приложении Б на рисунке Б.1.

**2.2.2** Для моделирования взаимодействия объектов во времени в языке UML используются диаграммы последовательностей. Для демонстрации диаграммы последовательностей рассмотрим диаграмму, представленную в приложении Б на рисунке Б.2.

Действие начинается с того, что клиент, в нашем случае инвестор, отправляет запрос на получение информации об оплате. Ответом на запрос, клиенту приходит форма для ввода конфиденциальных данных, затем происходит получение токена от API аккаунта оплаты, после того, как токен прошёл проверку на стороне Stripe API, уникальный идентификатор отправляется на сервер, где сохраняется под конкретным клиентом. После чего, инвестор уведомляется об успешной оплате.

**2.2.3** Диаграмма компонентов показывает разбиение программной системы на структурные компонент, взаимодействие и связи между компонентами. В приложении Б на рисунке Б.3 приведена диаграмма компонентов данного приложения. Ключевые компоненты фреймворка Ruby on Rails имеют логичную и удобную структуру, которая позволяет легко расширять приложение.

**2.2.4** Диаграмма развёртывания предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе её исполнения. На диаграмме развёртывания (приложение Б рисунок Б.4) можно увидеть, что для развёртывания разрабатываемой системы необходимы 3 устройства: браузер клиента, пользовательский сервер и сервер для базы данных.

**2.2.5** Диаграмма состояний предназначена для отображения состояний объектов системы, имеющих сложную модель поведения. Она показывает пространство состояний системы или её элементов, события, которые влекут переход из одного состояния в другое, действия, которые происходят при изменении состояния. Объекты меняют своё состояние в ответ на происходящие события и течением времени. В данной диаграмме пользователя представлен процесс оплаты и регистрации стартапов. Диаграмма состояний данной системы приведена в приложении Б на рисунке Б.5.

**2.2.6** Диаграмма классов отображает основные классы, необходимые для разработки программы. Диаграмма классов данной системы приведена в приложении Б на рисунке Б.6.

## 2.2 Проектирование схемы информационной модели предметной области

Информационная модель, разработанная в данном проекте представлена на рисунке 2.1.



Рисунок 2.1 – Информационная модель системы поиска стартапов

Информационная модель описывает бизнес процесс на уровне работы с информацией. Данная модель является основанием для создания базы данных. Хорошо продуманная база данных – это прежде всего набор поименованных таблиц. Каждая из которых в свою очередь содержит ряд полей, обладающих определёнными свойствами. Поля образуют структуру

базы данных – её основу. Наполнение базы данных информацией происходит путем создания записей. Запись – это строка в таблице, содержащая в себе значения всех необходимых для заполнения полей.

Схема содержит 10 таблиц: users, comments, transactions, resumes, topics, likes, ideas, posts, subscribers\_subscriptions, ideas\_resumes.

Сущность «users» содержит информацию о пользовательских аккаунтах в системе и включает в себя поля идентификационный номер аккаунта (primary key), роль аккаунта (администратор, инвестор, стартапер, разработчик), пароль в зашифрованном виде, статус подписки, уникальный идентификационный номер, который привязан к аккаунту оплаты.

Сущность «comments» содержит информацию о комментариях и включает в себя поля идентификационный номер комментария (primary key), содержимое комментария, идентификационный номер пользователя, идентификационный номер сущности, к которой он относится, также информацию о времени создания и обновления.

Сущность «transactions» содержит информацию о вложениях денежных средств инвесторами и включает в себя поля идентификационный номер транзакции (primary key), идентификационный номер идеи и инвестора, количество денежных средств для перевода, также информацию о времени создания и обновления.

Сущность «resumes» содержит информацию о созданных соискателями резюме, с их навыками, образованием и другой информацией.

Сущность «topics» содержит информацию для обсуждения различных тем на форуме и содержит идентификационный номер пользователя, тему, заголовок и описание.

Сущность «likes» содержит информацию о реакциях, включает идентификационный номер сущности для реакции, идентификационный номер пользователя и статус реакции.

Сущность «ideas» содержит информацию о стартапах и включает в себя поля идентификационный номер идеи (primary key), название, описание, цель сбора денежных средств, текущее количество денежных средств, номер пользователя, статус активности.

Сущность «posts» содержит информацию о новостных постах и включает в себя поля идентификационный номер новости (primary key), заголовок, описание, рейтинг, номер пользователя, также информацию о времени создания и обновления.

Сущность «subscribers\_subscriptions» содержит информацию о подписках и помогает реализовать отношение многие-ко-многим в базе

данных. Включает в себя поля идентификационный номер подписки (primary key), номер пользователя, идеи, также информацию о времени создания и обновления.

Сущность «ideas\_resumes» содержит информацию о подачах разработчиков на позиции в стартапах и помогает реализовать отношение многие-ко-многим. Включает в себя поля идентификационный номер вакансии (primary key), номер разработчика и его резюме, также информацию о времени создания и обновления.

База данных, разработанная для приложения находится в третьей нормальной форме, так как у неё нет повторяющихся строк, все атрибуты простые, все значения скалярные, каждая из таблиц имеет первичный ключ, все атрибуты описывают первичный ключ целиком, а не какую-то часть первичного ключа, а также база данных не имеет зависимостей одних не ключевых атрибутов от других, все атрибуты зависят от первичного ключа.

## 2.3 Архитектурные решения

В качестве языка программирования серверной части системы поиска финансирования стартапов был выбран Ruby и фреймворк Ruby on Rails.

Ruby — интерпретируемый, полностью объектно-ориентированный язык программирования с чёткой динамической типизацией, который был представлен в 1995 году разработчиком Юкихиро Мацумото. Он сочетает в себе Perl-подобный синтаксис с объектно-ориентированным подходом. Также некоторые черты заимствованы из языков программирования Python, Lisp, Dylan и CLU. Кроссплатформенная реализация интерпретатора языка Ruby распространяется на условиях открытого программного обеспечения[7].

Рассмотри характеристики Ruby:

- высокоуровневый — удобный для разработчиков, с сильной абстракцией и использованием конструкций натурального или человеческого языка;
- динамичный — типы данных определяются на этапе выполнения программы, это увеличивает скорость разработки;
- интерпретируемый — код на Ruby обрабатывается интерпретатором в момент выполнения без предварительной компиляции, это обеспечивает независимость от платформ и уменьшает размер исполняемых программ;
- язык программирования общего назначения — в отличие от предметно-ориентированных, языки программирования общего назначения не создаются для применения в специфичных областях.

Ruby on Rails – это фреймворк для разработки веб-приложений на языке программирования Ruby. На RoR были созданы такие проекты, как Redmine, Twitter, Shopify, Basecamp, GitHub, Kickstarter, Airbnb и другие. Rails также быстро адаптируется к новым тенденциям в веб-технологиях. Например, Rails одним из первых полностью реализовал архитектурный стиль REST структурирования веб-приложений.

Одним из плюсов фреймворка Ruby on Rails считается скорость разработки. Многолетний опыт показывает, что решение задач на Rails происходит в полтора раза быстрее по сравнению с другими фреймворками подобного плана. Ruby очень прост как инструмент, а также существует огромное количество готовых решений для различного рода задач.

При разработке масштабного проекта одним из самых важных моментов является кеширование. Ruby on Rails в основной комплектации имеет штатные средства кеширования данных. То есть у вас сразу будут в наличии инструменты для кеширования данных на проекте, и можно легко кешировать отдельные блоки кода или даже целые страницы.

Сейчас нередко в сети публикуются статьи о взломах различных ресурсов. Разработчики фреймворка Ruby on Rails очень серьёзно отнеслись к проблеме защиты данных. В RoR изначально присутствует шифрование паролей, данных кредитных карт и других личных данных пользователя, также исключены SQL инъекции и XSS атаки. Все входные параметры экранируются по умолчанию.

Философия Ruby on Rails включает два важных ведущих принципа:

- Don't Repeat Yourself — принцип разработки программного обеспечения, нацеленный на снижение повторения информации различного рода, особенно в системах со множеством слоёв абстрагирования. Принцип DRY формулируется как: «Каждая часть знания должна иметь единственное, непротиворечивое и авторитетное представление в рамках системы»;

- Convention Over Configuration — принцип проектирования программного обеспечения, заключающийся в том, что рассматриваемые аспекты нуждаются в конфигурации тогда и только тогда, когда этот аспект не удовлетворяет некоторой спецификации. Принцип как правило, применяется, чтобы сократить количество требуемой конфигурации без потери гибкости.

Фреймворк работает по принципу MVC. Взаимодействуя с приложением Rails, браузер посылает запрос, веб-сервер принимает его и передаёт контроллеру Rails, отвечающему за дальнейшую обработку. Иногда контроллер сразу отображает представление — шаблон, который



преобразуется в разметку HTML и возвращается браузеру. В динамических сайтах гораздо чаще контроллер взаимодействует с моделью — объектом Ruby, который представляет элемент сайта и отвечает за связь с базой данных. После вызова модели контроллер отображает представление и возвращает браузеру готовую веб-страницу с разметкой HTML. Подробная схема работы MVC в Rails представлена на рисунке 2.2.

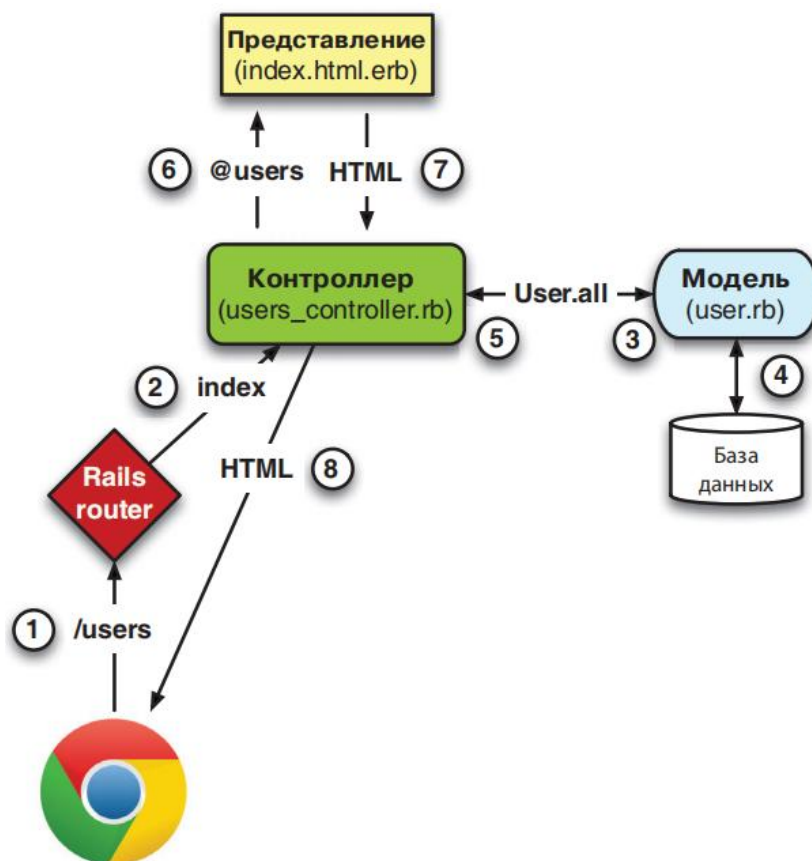


Рисунок 2.2 – Схема работы MVC в RoR

Для хранения объектов модели в реляционной СУБД по умолчанию в Rails использована библиотека ActiveRecord. Представление создает пользовательский интерфейс для отображения полученных от контроллера данных. Представление также передаёт запросы пользователя на манипуляцию данными в контроллер, как правило, представление не изменяет непосредственно модель.

Контроллер в Rails — это набор логики запускаемой после получения HTTP запроса сервером — отвечает за вызов методов модели и запускает формирование представления.

Контроллером в Ruby on Rails является класс, наследованный от ActionController::Base. Открытые методы контроллера являются так называемыми действиями. Действие часто соответствует отдельному представлению. Например, по запросу пользователя admin/list будет вызван метод list класса AdminController и затем использовано представление list.html.erb[8].

В Ruby on Rails представление описывается при помощи шаблонов ERB. Они представляют собой файлы HTML с дополнительными включениями фрагментов кода Ruby (Embedded Ruby). Вывод, сгенерированный встроенным кодом Ruby, включается в текст шаблона, после чего получившаяся страница HTML возвращается пользователю. Также в данном курсовом проекте использовался шаблонизатор Bootstrap, собственные таблицы стилей и JavaScript для динамических элементов.

Структура проекта представлена на рисунке 2.3.

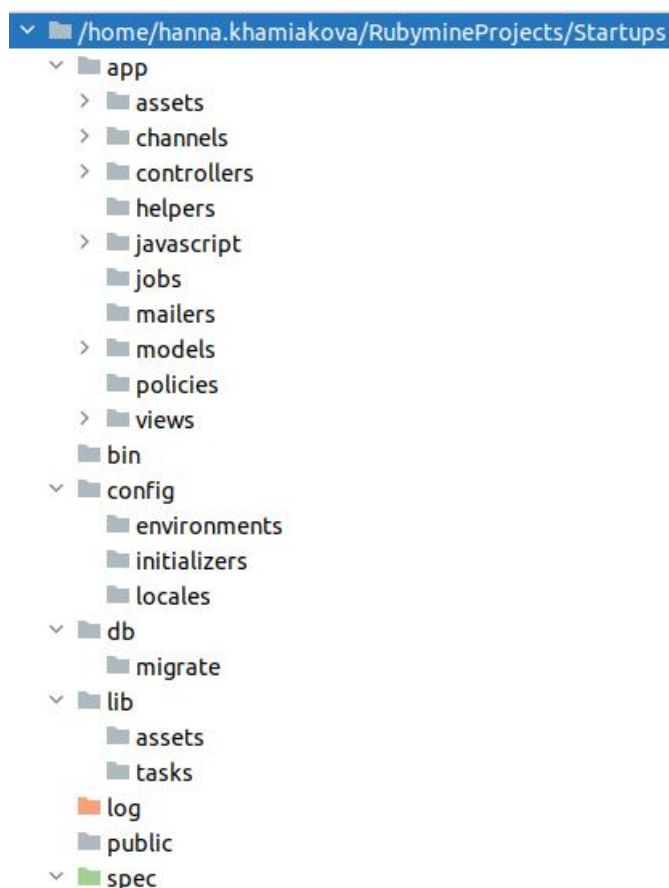


Рисунок 2.3 – Информационная модель системы поиска стартапов

В качестве СУБД для данного проекта была выбрана PostgreSQL не просто реляционная, а объектно-реляционная СУБД. Это даёт ему некоторые

преимущества над другими SQL базами данных с открытым исходным кодом, такими как MySQL, MariaDB и Firebird.

Фундаментальная характеристика объектно-реляционной базы данных — это поддержка пользовательских объектов и их поведения, включая типы данных, функции, операции, домены и индексы. Это делает PostgreSQL невероятно гибким и надёжным. Среди прочего, он умеет создавать, хранить и извлекать сложные структуры данных. У PostgreSQL множество возможностей. Созданный с использованием объектно-реляционной модели, он поддерживает сложные структуры и широкий спектр встроенных и определяемых пользователем типов данных. Он обеспечивает расширенную ёмкость данных и заслужил доверие бережным отношением к целостности данных.

Для выполнения отложенных задач, таких как еженедельные уведомления на почте, удаление стартапов по истечении срока был использован Sidekiq и фреймворк Active Job.

Active Job - это фреймворк для объявления заданий и их запуска на разных бэкендах для очередей. Эти задания могут быть чем угодно, от регулярно запланированных чисток до списаний с карт или рассылок. В общем, всем, что может быть выделено в небольшие работающие части и запускаться параллельно.

Sidekiq — одна из наиболее широко используемых инфраструктур фоновых задач, которые можно реализовать в приложении Rails. Она основана на системе хранения пар ключ-значение в оперативной памяти Redis, отличающейся гибкостью и высокой производительностью. Sidekiq использует Redis как хранилище для управления задачами, чтобы обрабатывать тысячи задач в секунду[9]. На рисунке 2.4 представлены Active Jobs, созданные в данном проекте.

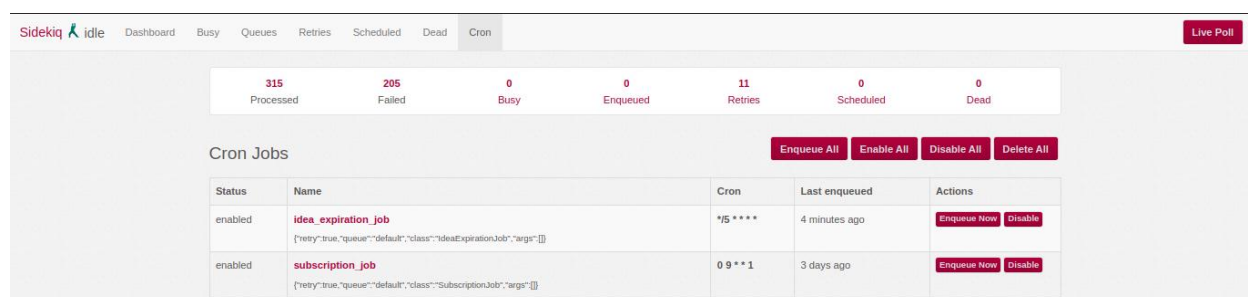


Рисунок 2.4 – Активные триггеры выполнения операций

Для осуществления платной подписки и перевода платежей будет использован Stripe API. Функции подписки являются простыми и мощными: Stripe поддерживает рынки, отчёты понятны и понятны, средства разработки отличаются расширенной настройкой, расширенные функции отчётности, которые помогают администраторам платформы отслеживать свои продажи

платных аккаунтов. На рисунке 2.5 представлено отображение продукта-подписки, созданного в аккаунте Stripe для подключения системы.

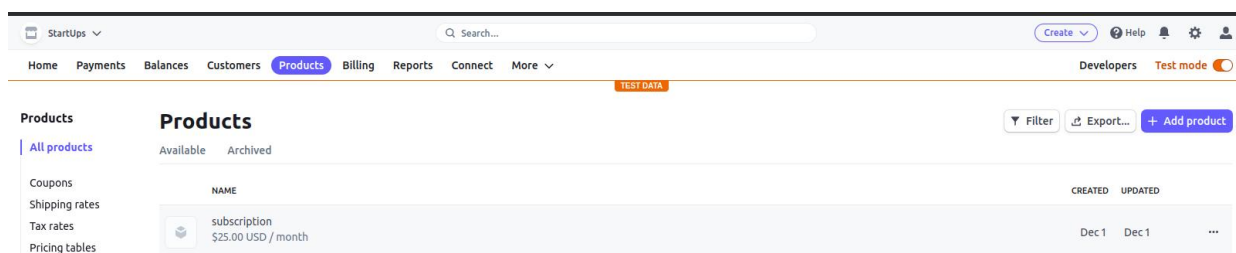


Рисунок 2.5 – Продукт-подписка в аккаунте Stripe

Таким образом все технологии, выбранные для реализации проекта, являются актуальными, развивающимися и позволяют облегчить процесс разработки, сделать платформу более гибкой и способной к расширению в будущем.

## 2.4 Описание и разработка алгоритмов, реализующих бизнес-логику разрабатываемого программного средства

Алгоритмы, реализующие важные бизнес-функции приложения дают более подробную картину эффективности работы. Схемы данного подраздела представлены в приложении В.

Рассмотрит алгоритм позиции на стартапе разработчиком (рисунок В1). Для начала необходимо создать резюме, заполнив все необходимые данные об опыте работы, в случае, если данные получены успешно, можно переходить к оплате подписки. Если подписка прошла безуспешно, пользователь получит уведомление об ошибке, иначе разработчик получит доступ ко всем бизнес-идеям и сможет осуществлять поиск позиций. Отправив резюме, остаётся ждать ответа от владельца идеи. Если он готов взять разработчика, кандидата принимают на позицию, иначе разработчик продолжает поиск.

Алгоритм получения отчётности представлен на рисунке В2 и состоит в том, чтобы нажатием на кнопку «Оформить подписку» клиент смог получать уведомления с отчётом о прогрессе сбора денежных средств на продвижение стартапа. Оформление подписки возможно только в платной версии, поэтому при отсутствии подписки, система выведет ошибку. После того, как подписка успешно оформлена, пользователь может просмотреть отчёт на почте, скачать его и открыть.

### 3 ТЕСТИРОВАНИЕ И ПРОВЕРКА РАБОТОСПОСОБНОСТИ ПРОГРАММНОЙ ПОДДЕРЖКИ ИНВЕСТИЦИОННЫХ ПРОЕКТОВ

Тестирование – процесс анализа программного средства и сопутствующей документации с целью поиска дефектов и повышения качества продукта.

Для проверки уровня базовых пользовательских требований будем использовать тестирование на основе тест-кейсов.

Тест-кейс – набор входных данных, условий выполнения и ожидаемых результатов, разработанный с целью проверки того или иного свойства или поведения программного средства.

Оформленные основные тест-кейсы для данного программного приложения представлены в таблице 3.1.

Таблица 3.1 – Тест-кейсы для проверки базовых пользовательских требований

№	Заглавие тест-кейса	Шаги тест-кейса	Ожидаемый результат	Статус тест-кейса
1.	Зарегистрировать нового пользователя в системе	1. Открыть форму регистрации. 2. Заполнить поля формы регистрации необходимыми данными.	1. Форма регистрации открылась, доступны все поля для ввода. 2. При вводе корректных данных пользователь зарегистрирован. Иначе – сообщение об ошибке.	Passed
2.	Прикрепить резюме	1. Открыть вкладку с идеями. 2. Выбрать позицию. 3. Прикрепить.	1. Открылась вкладка с вакансиями. 2. Подтверждение. 3. Отклик отправлен.	Passed

Рисунок 3.1 – Таблица с тест-кейсами для проверки базовых пользовательских операций

Разработанная система покрыта RSpec тестами, также применяется gem Capybara.

RSpec – это модуль модульного тестирования для языка программирования Ruby. RSpec отличается от традиционных платформ xUnit, таких как JUnit, потому что RSpec – это инструмент разработки, управляемый поведением. Это означает, что тесты, написанные на RSpec, фокусируются на «поведении» тестируемого приложения. RSpec делает упор не на том, как работает приложение, а на том, как оно ведёт себя, другими словами, на самом деле, что делает приложение[10].

Подтверждение успешности всех сценариев представлено на рисунке 3.2.



```
hanna.khamiakova@wsa-241-74a:~/RubymineProjects/onlineer$ bundle exec rspec
..Capybara starting Puma...
* Version 5.6.5 , codename: Birdie's Version
* Min threads: 0, max threads: 4
* Listening on http://127.0.0.1:35755
DEBUGGER: Attaching after process 60773 fork to child process 60799
.....
Finished in 3.97 seconds (files took 0.72983 seconds to load)
33 examples, 0 failures
```

Рисунок 3.2 – Выполнение тестовых сценариев

Таким образом, разработанные тест-сценарии помогли избежать потери данных и некорректной работы приложения при возникновении исключительных ситуаций.

## 4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ ПО ИСПОЛЬЗОВАНИЮ ПРОГРАММНОГО СРЕДСТВА ДЛЯ ПОИСКА ИНВЕСТИЦИЙ

Для того, чтобы в полной мере освоить принцип работы разработанной краудфандинговой платформы и введения в эксплуатацию разработанного веб-приложения, рассмотрим руководство для пользователей всех ролей.

На рисунке 4.1 представлена главная страница приложения, где размещена ключевая информация о проекте, ссылки для регистрации, авторизации, посещения форума, новостного канала и отправки обратной связи.

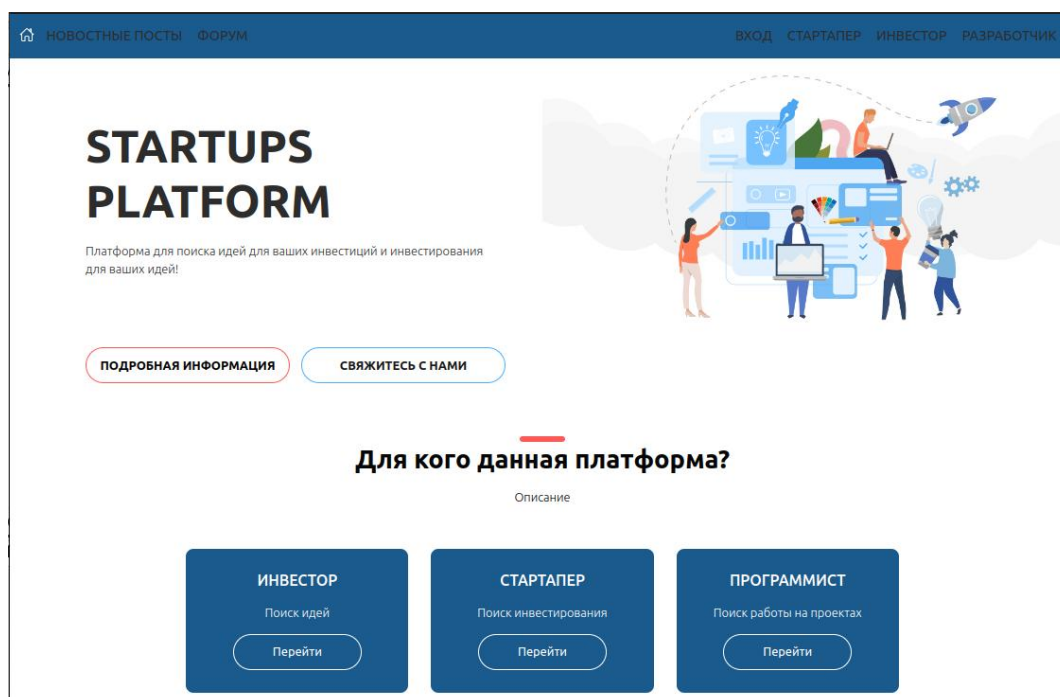


Рисунок 4.1 – Главная страница

Для осуществления входа необходимо выбрать соответствующую ссылку на панели задач. Система автоматически определит какой ролью обладает пользователь. Также происходит валидация данных. На рисунке 4.2 представлен вход в систему в качестве инвестора.

Рассмотрим также исключительные ситуации: если аккаунт не был зарегистрирован ранее, пользователю необходимо нажать на «Зарегистрироваться» внизу панели входа и пройти успешную регистрацию с подтверждением почты и пароля.

The screenshot shows a login form titled "Вход в аккаунт" (Login). At the top, there is a navigation bar with links: "НОВОСТНЫЕ ПОСТЫ" (News Posts) and "ФОРУМ" (Forum) on the left, and "ВХОД" (Login), "СТАРТАПЕР" (Startup), "ИНВЕСТОР" (Investor), and "РАЗРАБОТЧИК" (Developer) on the right. The login form itself has two input fields: the first for email, containing "invest@gmail.com", and the second for password, masked with "\*\*\*\*\*". Below the password field is a checkbox labeled "Запомнить меня" (Remember me). A green button labeled "Войти" (Login) is positioned below the checkbox. At the bottom of the form, there are two links: "Зарегистрироваться" (Register) and "Забыли пароль?" (Forgot password?).

Рисунок 4.2 – Вход в аккаунт

Панель регистрации представлена на рисунке 4.3. Минимальное количество символов в пароле равно 6, поэтому ввод меньшего количества символов приведёт к появлению соответствующего уведомления.

The screenshot shows a registration form titled "Регистрация" (Registration). It contains several input fields: "Email", "Имя" (Name), "Фамилия" (Surname), "Пароль (6 символов минимум)" (Password (minimum 6 symbols)), and "Подтвердите пароль" (Confirm password). Below these fields is a larger text area labeled "Обо мне" (About me). At the bottom of the form, there is a green button labeled "Зарегистрироваться" (Register) and a blue link labeled "Войти" (Login).

Рисунок 4.3 – Регистрация пользователей

Если пользователь забыл пароль от своего аккаунта, нажав на ссылку «Забыли пароль?», есть возможность получить инструкции о восстановлении пароля на почту. На рисунке 4.4 отображена панель сброса пароля.





Рисунок 4.4 – Сброс пароля

Также в верхней части находится навигационная панель, на которой находится логотип, при нажатии на который мы можем перейти на главную страницу.

После успешной авторизации пользователь попадает в свой кабинет (рисунок 4.5), где отображается основная информация о пользователе-инвесторе, а также информация о проведенных денежных средствах на стартапы и комментарии.

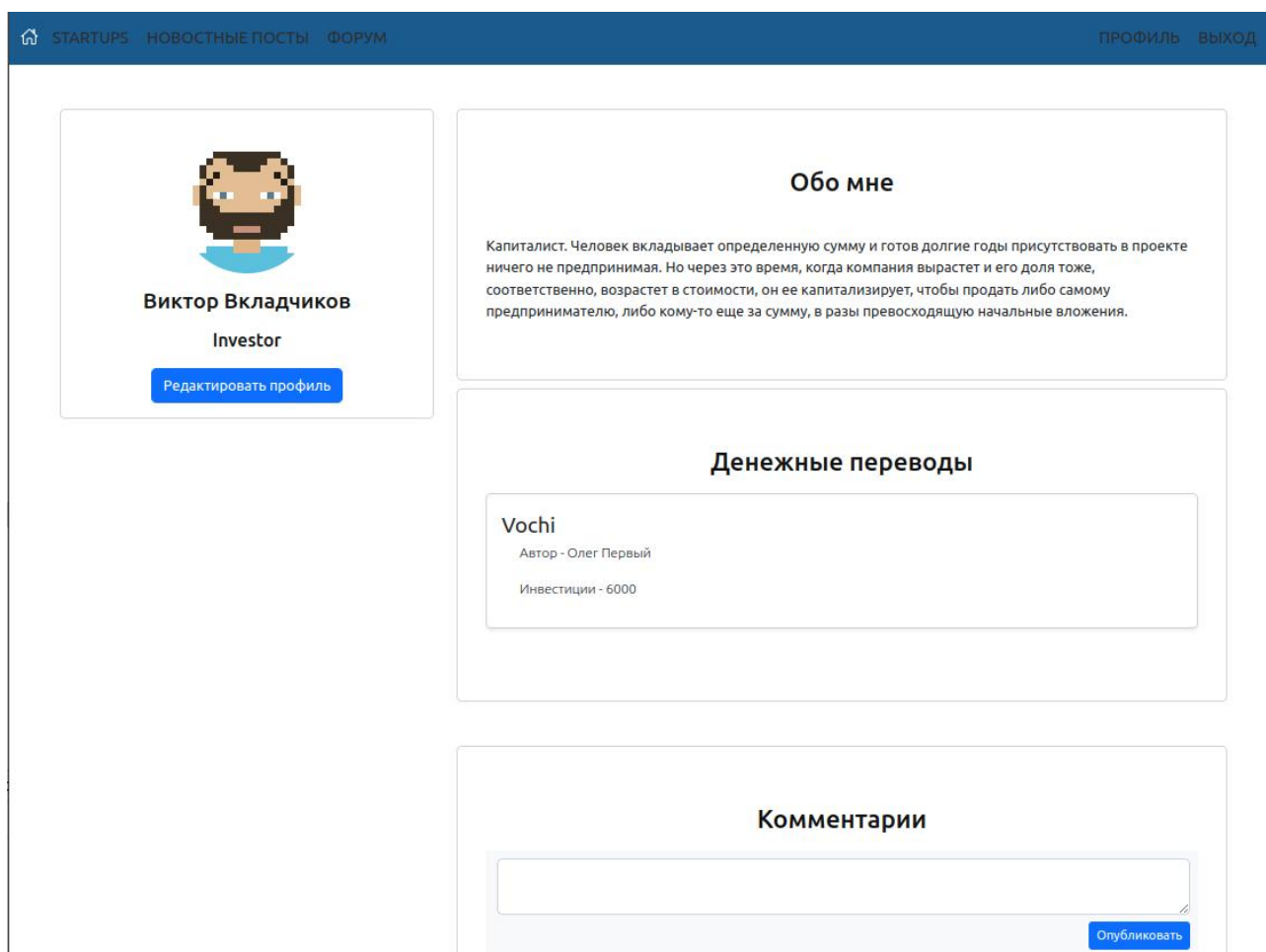


Рисунок 4.5 – Личный кабинет инвестора

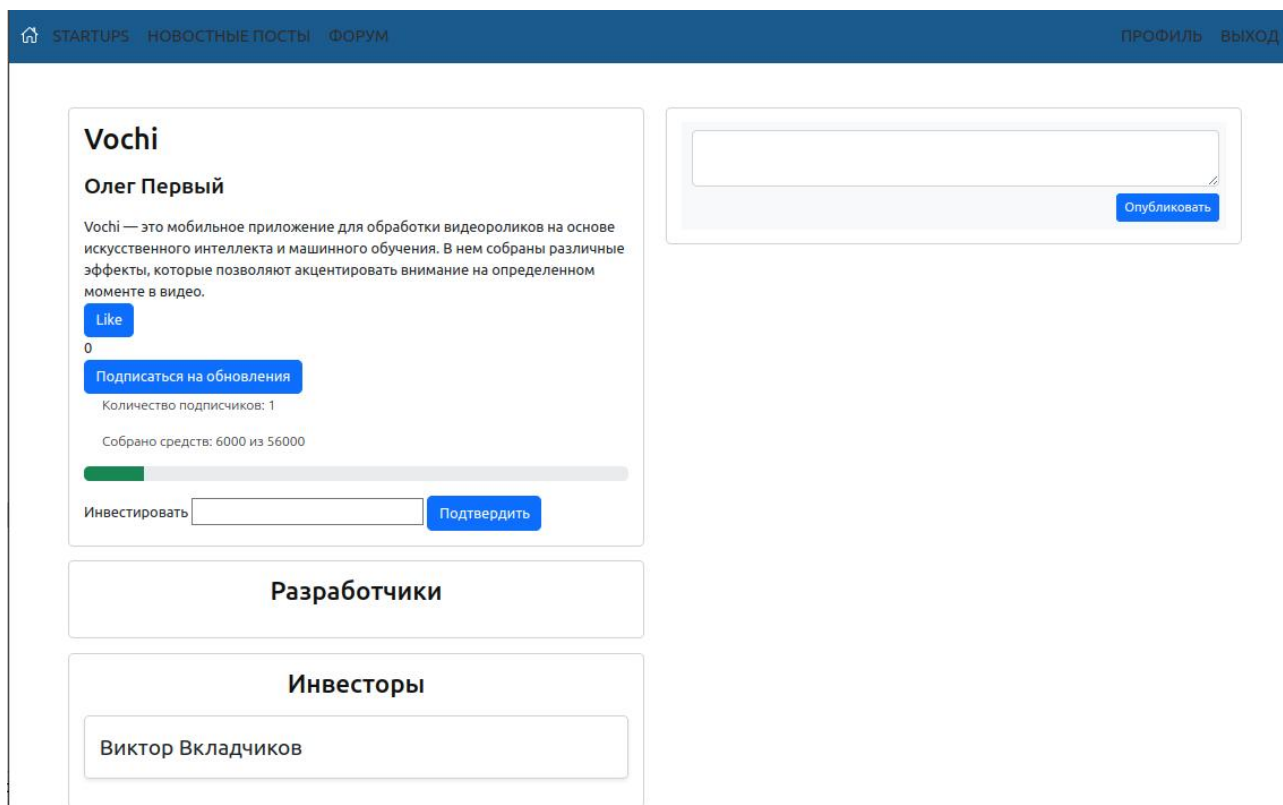


Рисунок 4.6 – Сведения об инвестициях пользователя

На рисунке 4.6 представлена информация о стартапе, в который инвестировал соискатель стартапа. Здесь присутствует описание идеи, рейтинг, оценка, возможность осуществить подписку на обновления, прогресс сбора денежных средств. Также размещена информация о разработчиках, инвесторах и комментариях.

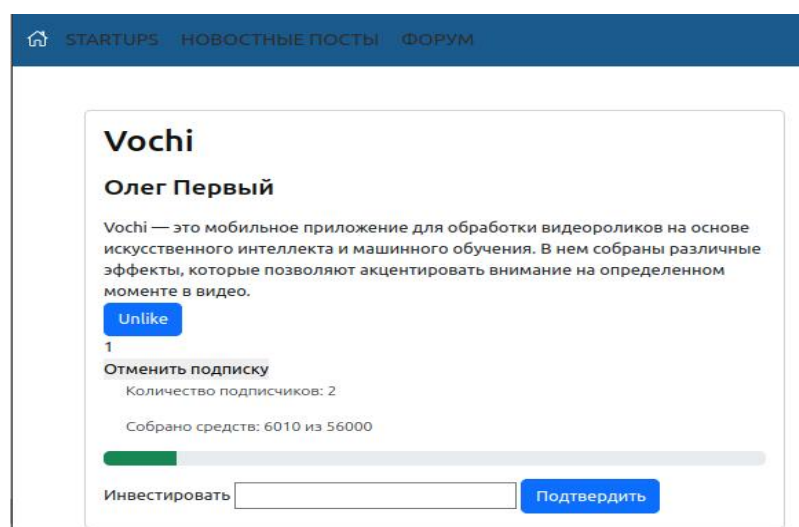


Рисунок 4.7 – Результат перевода денежных средств

На рисунке 4.7 видно, что инвестор перевёл 10 денежных единиц и оформил подписку на обновления. Поэтому в личном кабинете инвестора появилась информация о новом денежном переводе (рисунок 4.8).

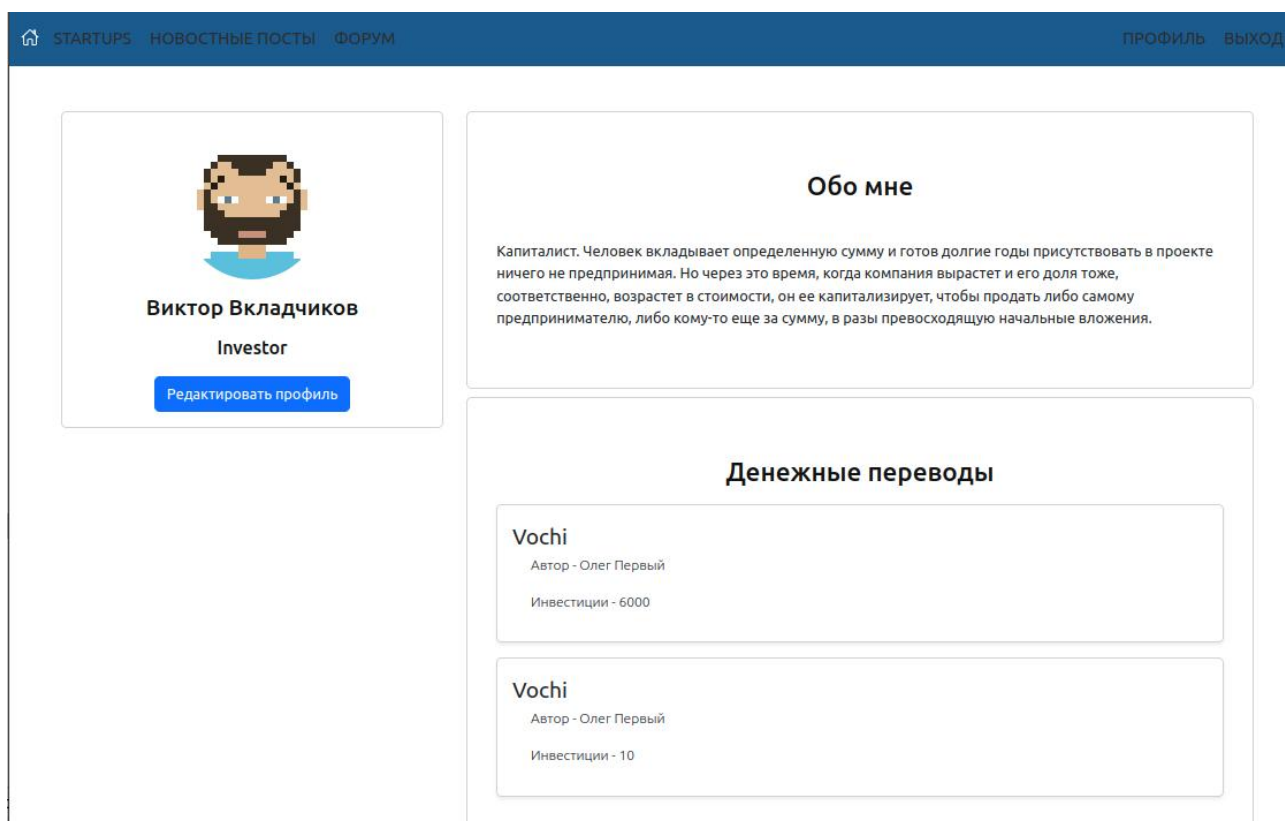


Рисунок 4.8 – Результат перевода денежных средств в аккаунте инвестора

Оставим комментарий и реакцию от лица инвестора Виктора, результат представлен на рисунке 4.9.

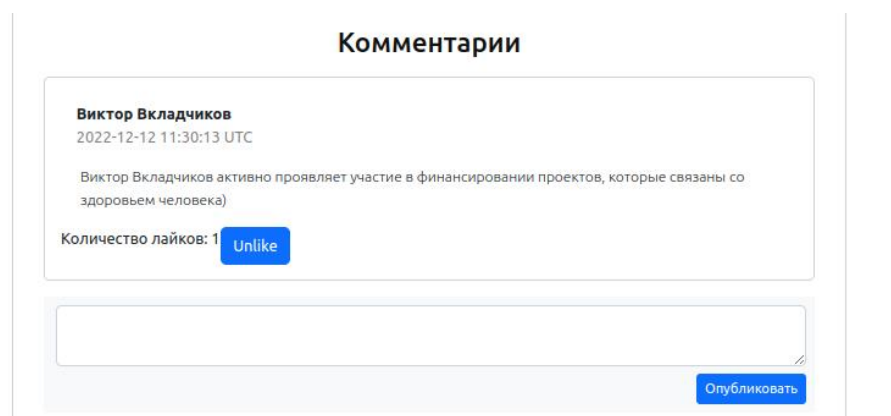


Рисунок 4.9 – Публикация комментария и реакции

Все зарегистрированные пользователи имеют доступ к форуму. Для того, чтобы создать новую тему для обсуждения, необходимо нажать на кнопку вверху страницы и заполнить поля темы обсуждения, описания (рисунок 4.10).

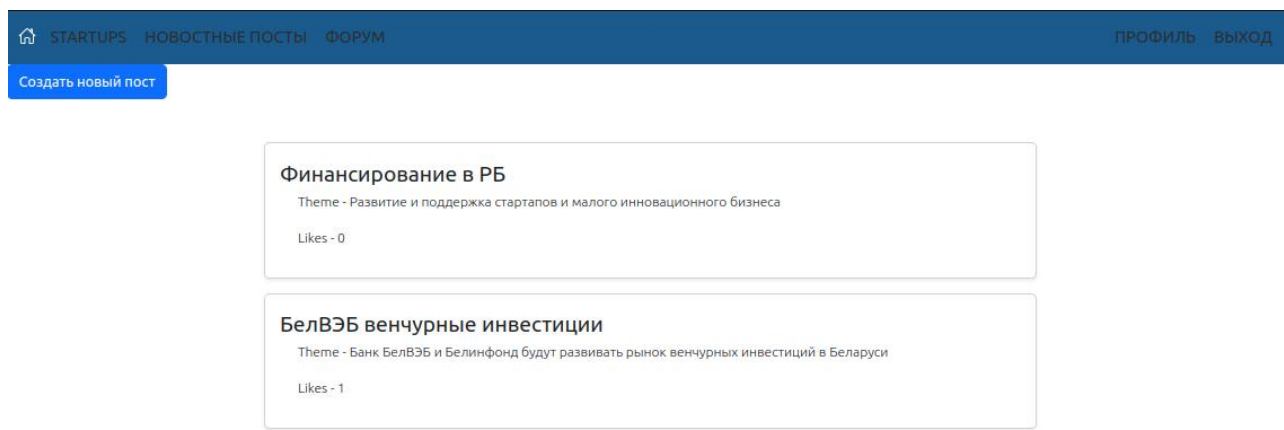


Рисунок 4.10 – Форум для обсуждения

Перейдём на страницу идей (рисунок 4.11).

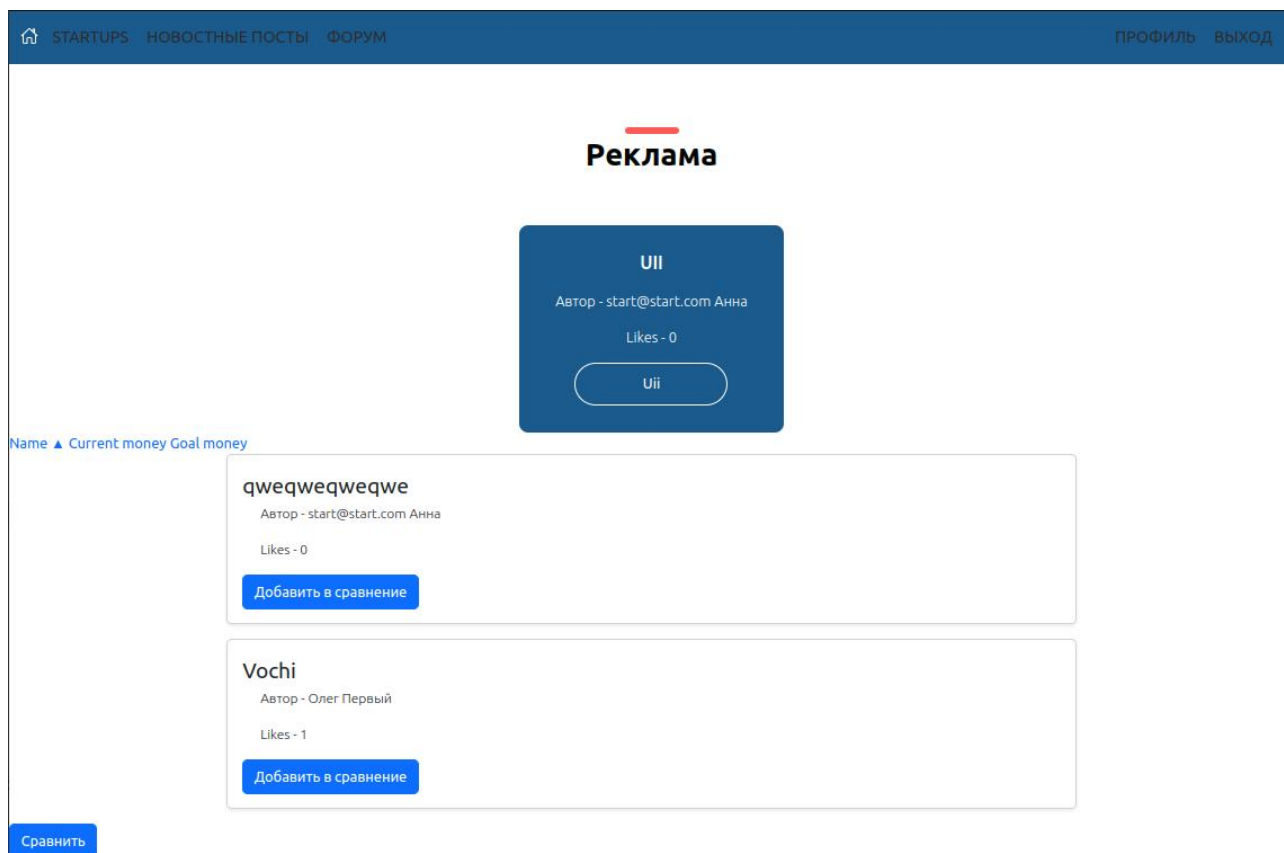


Рисунок 4.11 – Страница с бизнес-идеями

Вверху отображается контекстная реклама стартапов, создатели которых оформили платную подписку. При обновлении страницы меняется идея и её детали.

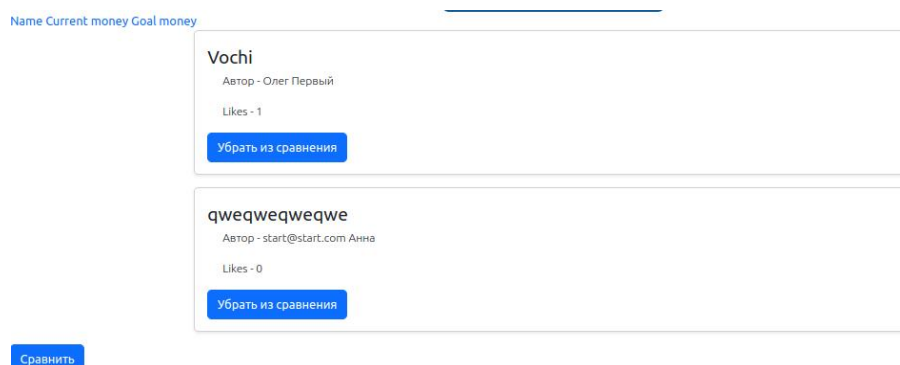


Рисунок 4.12– Выбор стартапов для сравнения

На рисунке 4.12 отображено взаимодействие с идеями: сортировка по критериям и сравнение. Добавим две идеи для сравнения.

qweqweqweqwe	Vochi
Автор: start@start.com Анна	Автор: Олег Первый
Разработчики: 1	Разработчики: 0
Инвесторы: 1	Инвесторы: 4
Идея создана: December 09, 2022 11:38	Идея создана: December 08, 2022 11:59
Средства: 13336246	Средства: 6010
Цель: 12312312	Цель: 56000
Текущий статус: Running	Текущий статус: Running
Likes: 0	Likes: 1
Комментарии: 0	Комментарии: 0

Рисунок 4.13 – Сравнение бизнес-идей

Сравнение стартапов происходит в табличной форме для удобного просмотра, результат представлен на рисунке 4.13.

Произведём вход от лица администратора, пользователей с такой ролью может быть не более одного, что обеспечивает конфиденциальность данных всех пользователей системы.

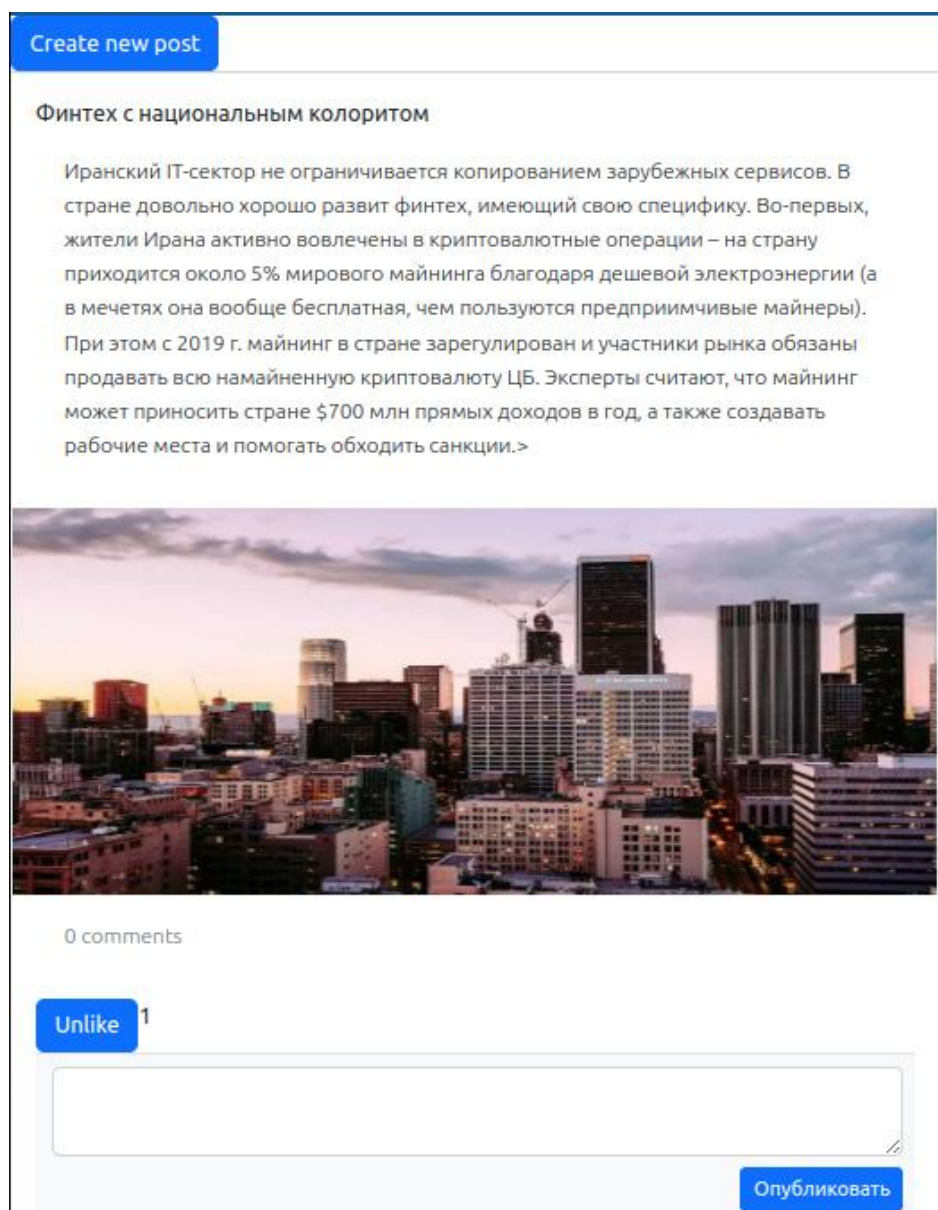


Рисунок 4.14 – Новостная лента

Правами ведения новостной ленты обладает только администратор, но комментировать посты могут все зарегистрированные пользователи (рисунок 4.14).

Администратор также может просматривать действия платных аккаунтов, пример представлен на рисунке 4.15

## Events

Filter

EVENT	ID	DATE
dev@gmail.com is a new customer	evt_1MECV6BPeaQvvxt96VAe47rz	12/12/22, 4:57:00 PM
ilia@gmail.com is a new customer	evt_1MECHDBPeaQvvxt9zepsNcNN	12/12/22, 4:42:39 PM
A Checkout Session was completed	evt_1MEC7GBPeaQvvxt9Znsob5aI	12/12/22, 4:32:22 PM
mir@gmail.com's payment for an invoice for \$25.00 USD succeeded	evt_1MEC7GBPeaQvvxt9pWahQ1i	12/12/22, 4:32:21 PM
mir@gmail.com's invoice for \$25.00 USD was paid	evt_1MEC7GBPeaQvvxt9E63EoJGk	12/12/22, 4:32:21 PM
mir@gmail.com's subscription has changed	evt_1MEC7GBPeaQvvxt9v1jCBqLA	12/12/22, 4:32:21 PM
mir@gmail.com's invoice has changed	evt_1MEC7GBPeaQvvxt9DPw8mz7x	12/12/22, 4:32:21 PM
A card payment method ending in 4242 was attached to customer cus_My8IINZ6watJif	evt_1MEC7GBPeaQvvxt9TyLxxq23	12/12/22, 4:32:21 PM
The payment pi_3MEC7DBPeaQvvxt903NymZyk for \$25.00 USD has succeeded	evt_3MEC7DBPeaQvvxt90AG0mKej	12/12/22, 4:32:21 PM
mir@gmail.com was charged \$25.00 USD	evt_3MEC7DBPeaQvvxt90BvBK014	12/12/22, 4:32:21 PM
mir@gmail.com attempted to subscribe to price_1MABaGBPeaQvvxt9f4F7mOzL	evt_1MEC7GBPeaQvvxt9oArKSoEN	12/12/22, 4:32:19 PM
A draft invoice for \$25.00 USD to mir@gmail.com was finalized	evt_1MEC7GBPeaQvvxt9Awf0vDes	12/12/22, 4:32:19 PM
A draft invoice was created	evt_1MEC7GBPeaQvvxt9XfJD29Va	12/12/22, 4:32:19 PM
mir@gmail.com's details were updated	evt_1MEC7GBPeaQvvxt9BwogMAwS	12/12/22, 4:32:19 PM
A new payment pi_3MEC7DBPeaQvvxt903NymZyk for \$25.00 USD was created	evt_3MEC7DBPeaQvvxt90T7knq80	12/12/22, 4:32:19 PM

Рисунок 4.15 – Отслеживание платных аккаунтов

Осуществим регистрацию нового стартапера – рисунок 4.16.

### Регистрация

mira@gmail.com

Мирослава

Томская

\*\*\*\*\*

\*\*\*\*\*

Развитие стартапов в сфере финтех

Зарегистрироваться

Войти

Рисунок 4.16 – Регистрация инвестора

Если данные введены корректно, то переходим в профиль соискателя инвестиций для бизнес-идеи.



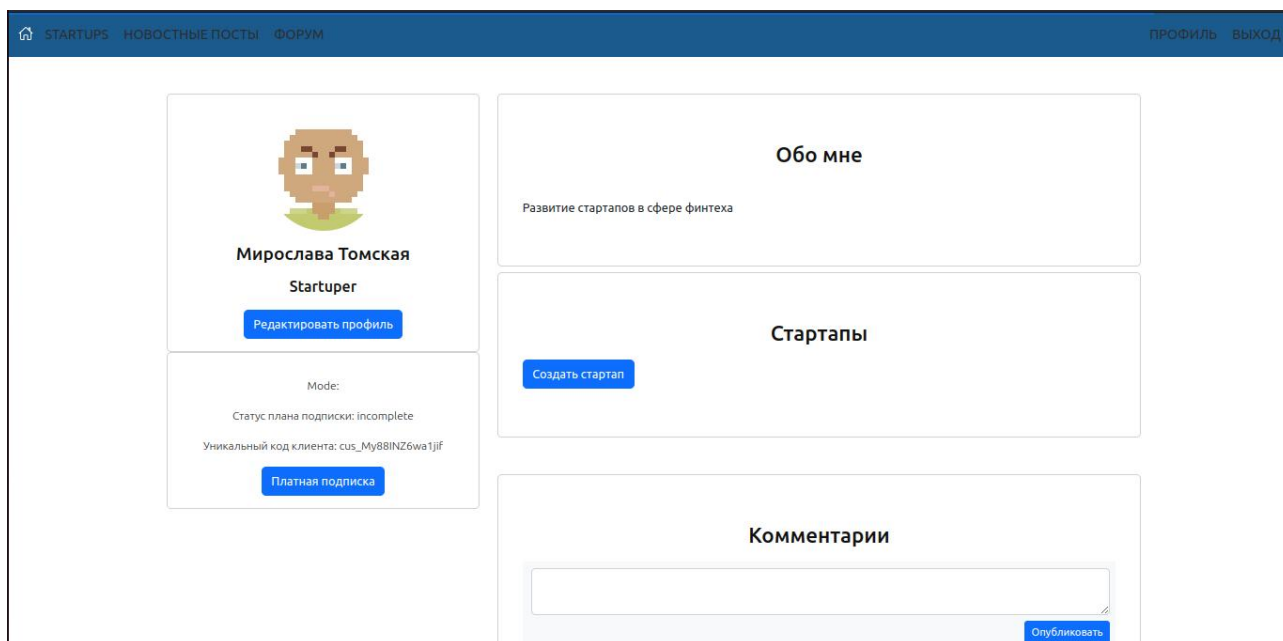


Рисунок 4.17 – Профиль стартапера

На странице профиля соискателя инвестиций (рисунок 4.17) пока нет бизнес-идей, так как не оформлена оплата подписки для регистрации идей.

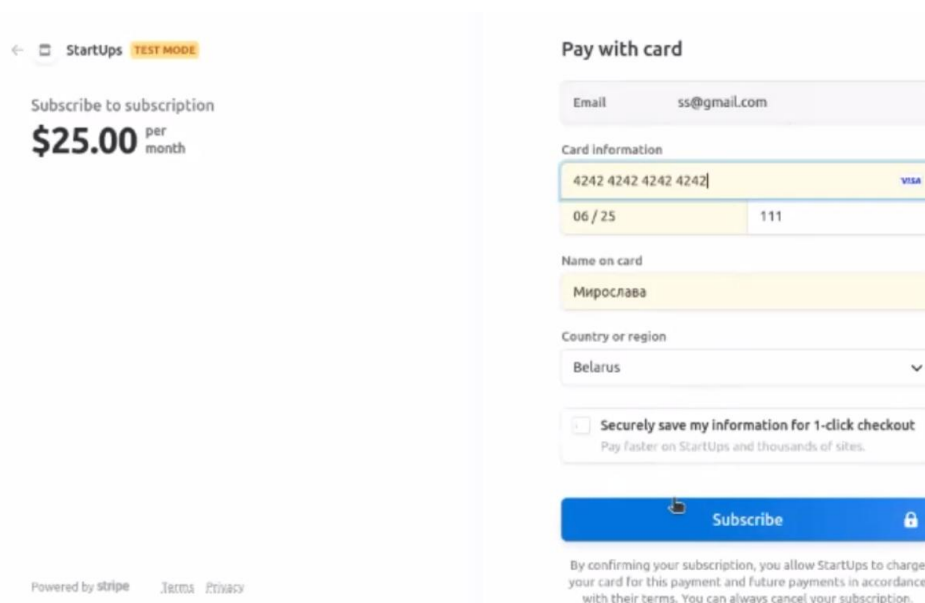


Рисунок 4.17 – Ввод данных для оплаты подписки

Введя данные банковской карты нажимаем на кнопку «Подписаться» (рисунок 4.17).

Для проверки регистрации со стороны администратора откроем историю оплаты (рисунок 4.18).



## Payments

1 selected

Deselect

Refund ctrl + enter

All Succeeded Refunded Uncaptured Failed

Date Amount Status Payment method

Clear Filters

AMOUNT	DESCRIPTION	CUSTOMER	DATE
<input type="checkbox"/> \$25.00 USD Succeeded ✓	Subscription creation	ss@gmail.com	Dec 14, 11:05 AM ...
<input type="checkbox"/> \$25.00 USD Succeeded ✓	Subscription creation	fff@gmail.com	Dec 14, 10:34 AM ...
<input checked="" type="checkbox"/> \$25.00 USD Succeeded ✓	Subscription creation	mir@gmail.com	Dec 12, 4:32 PM ...
<input type="checkbox"/> \$110.00 USD Succeeded ✓	Subscription update	mi@gmail.com	Dec 12, 2:48 PM ...
<input type="checkbox"/> \$10.00 USD Succeeded ✓	Subscription update	aaa2@gmail.com	Dec 12, 1:23 PM ...
<input type="checkbox"/> \$100.00 USD Succeeded ✓	Subscription update	qqq@gmail.com	Dec 12, 12:37 PM ...

Рисунок 4.17 – История оплаты подписки

Осуществим создание бизнес-идеи «Умные интерфейсы», результат представлен на рисунке 4.18. И откроем подробную информацию о нём (рисунок 4.19).

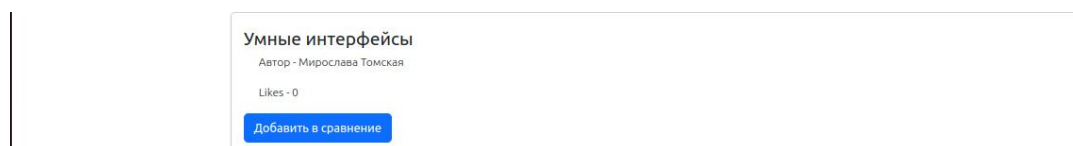


Рисунок 4.18 – Регистрация нового стартапа

STARTUPS

НОВОСТНЫЕ ПОСТЫ

ФОРУМ

ПРОФИЛЬ

ВЫХОД

Умные интерфейсы

Миромира Томская

Общий смысл концепции умного интерфейса можно описать с помощью следующей аналогии: это персональный консультант, который на 100% компетентен в банковских продуктах и находится в курсе контекста ситуации клиента и предыстории его взаимодействия с банком. Более того, консультант постоянно учится и подстраивается под манеру работы клиента с системой. Результат — выстроенная под пользователя навигация системы, шорткаты для текущих задач, актуальные и действительно полезные контекстные подсказки и, конечно, котик в чате, когда пользователю грустно. Система будет запоминать все рутинные операции, актуальные для конкретного пользователя, и создавать к ним шорткаты в интерфейсе. Благодаря этому решать такие задачи можно будет в один-два клика.

Unlike

1

Подписаться на обновления

Количество подписчиков: 0

Собрано средств: 0 из 147000

Разработчики

Инвесторы

6

4

2

0

Dec 5

Dec 7

Dec 9

Dec 11

Миромира Томская

2022-12-12 13:36:11 UTC

Обновление: MVP прошел все этапы тестирования

Количество лайков: 1

Unlike

Опубликовать

Рисунок 4.19– Форма для добавления рекрутера

40

Осуществим вход в качестве инвестора, у которого пока нет денежных переводов на счёт стартапов (рисунок 4.20).

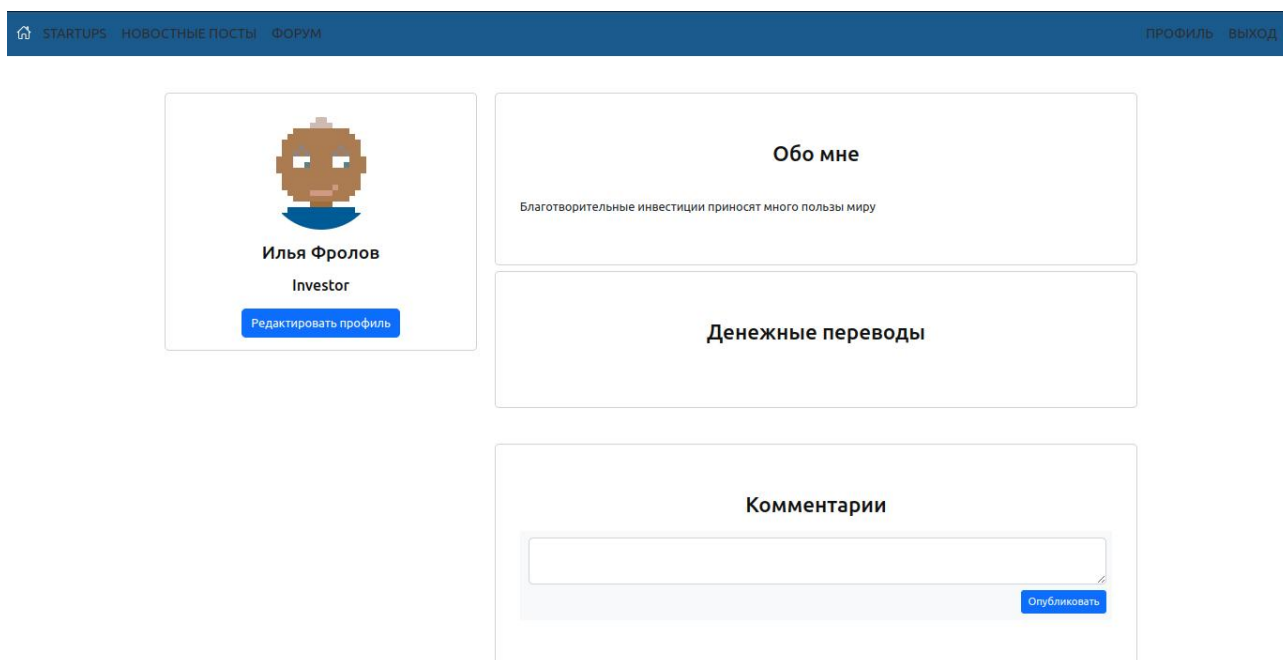


Рисунок 4.20 – Профиль инвестора до проведения денежных переводов

Теперь осуществим перевод денег для стартапа «Умные интерфейсы». На почту инвестора придёт уведомление (рисунок 4.21). А в профиле появится информация о денежном переводе (рисунок 4.22).



Рисунок 4.21 – Получение письма о переводе денежных средств

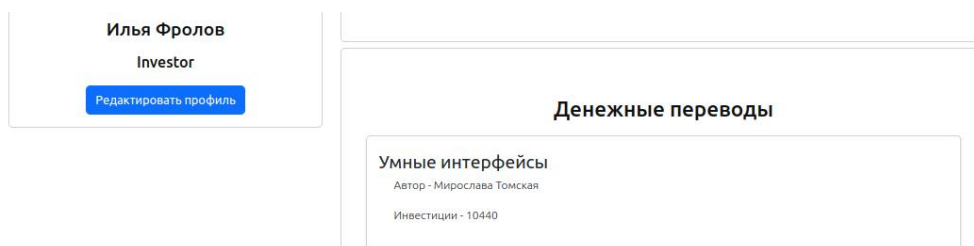


Рисунок 4.22 – Отображение пополнение счёта в профиле инвестора

Также проверим отображение пополнения счёта на странице самого стартапа, результат на рисунке 4.23.

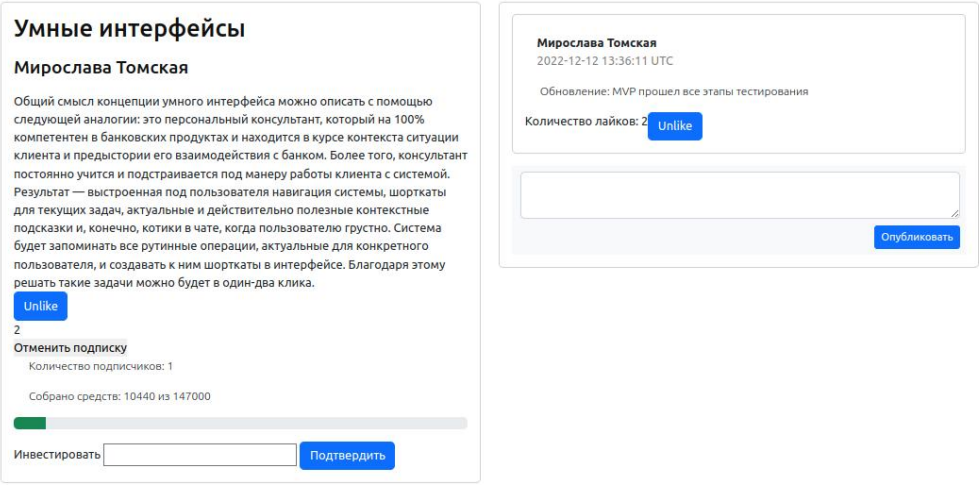


Рисунок 4.23 – Отображение пополнение счета на странице стартапа

Выполним ещё несколько пополнений счёта, для того чтобы проверить корректность построения графической отчётности. Первый график показывает зависимость количества пополняемых денежных средств от времени. Второй график выявляет долю конкурентности в день по созданию стартапов на краудфандинговой платформе (рисунок 4.24).

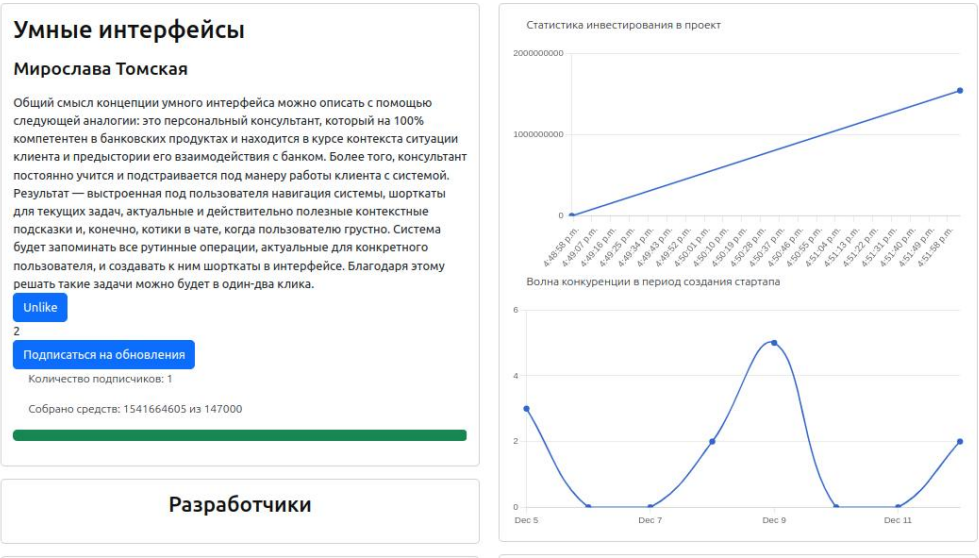


Рисунок 4.24 – Графическое представление прогресса сбора средств и конкуренции

Произведём вход в качестве разработчика и заполним поля для создания резюме (рисунок 4.25).

The screenshot shows a registration form with a dark blue header. The header contains navigation links: 'STARTUPS', 'НОВОСТНЫЕ ПОСТЫ', and 'ФОРУМ' on the left, and 'ПРОФИЛЬ' and 'ВЫХОД' on the right. Below the header is a form with several input fields: 'искусственный интеллект', 'высшее профильное', 'Беларусь', '7', and 'Нейронные сети,'. There is also an 'Add Resume' button.

Рисунок 4.25 – Ввод данных для создания резюме

Результат создания резюме изображён на рисунке 4.26.

The screenshot shows a user profile page. On the left is a profile card for 'Сергей Программист' (Developer) with a pixelated avatar and a 'Редактировать профиль' button. The main content area has two sections: 'Обо мне' (About me) with the text 'Разработка ПО любых доменов' and 'Резюме' (Resume). The resume section lists: 'Специализация — искусственный интеллект', 'Образование — высшее профильное', 'Страна — Беларусь', 'Количество лет опыта — 7', and 'Дополнительно — Нейронные сети, аналитические переборки, бизнес-анализ в программировании'. There is a 'Удалить резюме' button at the bottom of the resume section.

Рисунок 4.26 – Профиль разработчика с созданным резюме

Зайдя на страницу стартапа «Умные интерфейсы», прикрепим резюме (рисунок 4.27).

The screenshot shows a startup page. At the top is a progress bar with the text 'Собрано средств: 1541664605 из 147000'. Below the progress bar are two sections: 'Разработчики' (Developers) and 'Инвесторы' (Investors). The 'Инвесторы' section lists 'Илья Фролов' and 'Виктор Вкладчиков'. At the bottom is a button labeled 'Прикрепить резюме' (Attach resume).

Рисунок 4.28 – Прикрепление резюме

Перейдя на страницу стартапа с аккаунта соискателя инвестиций, видим прикреплённое резюме Сергея. Стартапер может его принять на работу либо отклонить (рисунок 4.28).

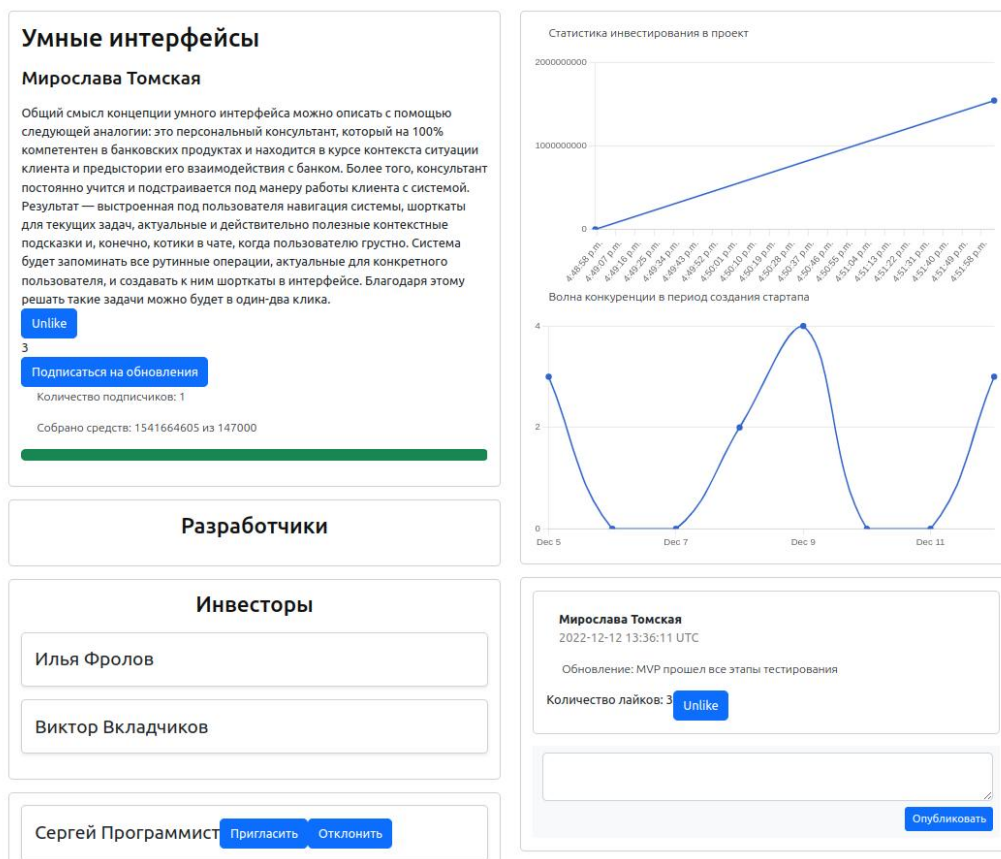


Рисунок 4.28 – Пригласить/отклонить разработчика на проект

Таким образом, была разработана специальная инструкция для описания всех возможностей созданной краудфандинговой платформы.

## ЗАКЛЮЧЕНИЕ

В ходе курсовой работы были изучены особенности инвестиционных программ для сбора средств на развитие бизнес-проектов на начальных этапах. Программа сделана удобно даже для неопытных пользователей, в будущем возможно усовершенствование программы и её интерфейса.

Итогом данного курсового проекта является функционирующее программное приложение, пригодное для использования с целью поиска инвестиционных проектов.

Для достижения поставленных целей были разработаны и реализованы все пользовательские функции. Также была создана диаграмма IDEF0 и её декомпозиция, а также соответствующие UML диаграммы.

Приложение успешно построено по описанной актуальной архитектуре.

Стоит отметить, что пользователи могут с лёгкостью разобраться с программой, так как в ней реализованы понятные меню. А само приложение успешно запускается без использования интегрированных средств разработки. Интерфейс программы является вполне приятным для восприятия и работы с ней.

Сама система предназначена для повышения эффективного старта потока инвестирования растущих бизнес-планов.

Основным преимуществом системы является полная автоматизация процесса сбора бизнес-идей и денежных средств.

Данные, используемые программным средством, хранятся в базе данных и доступ к ней предоставляется в соответствии с установленными полномочиями.

При этом не исключается факт, что разработанное ПО может улучшаться и модернизироваться в будущем.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- [1] Зейналов, А.А. Краудинвестинг: современная система организации и финансирования проектной деятельности: монография / колл. авторов. – М.: РУСАЙНС, 2017. – 168 с.
- [2] Клещева, С.А. Краудфинансирование как инструмент инвестирования инновационного предпринимательства / С.А. Клещева // Экономика и банки. – 2017. – № 2. – С. 40–46.
- [3] Горовая, В.В. Практическое пособие по краудфандингу / В.В. Горовая; под ред. канд. экон. наук Ф.В. Мурачковского. – М., 2016. – 125 с
- [4] Пособие по краудфандингу [Электронный ресурс]. – Электронные данные. – Режим доступа: [http://www.kdobru.ru/netcat\\_files/171/143](http://www.kdobru.ru/netcat_files/171/143)
- [5] Архитектура Ruby [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://adrianmejia.com/ruby-on-rails-architectural-design>
- [6] Диаграмма состояний [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://it.wikireading.ru/5134>
- [7] Диаграмма состояний [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://portal.tpu.ru/SHARED/1/LIKHUD/ucheb/modelling/modelling/pdf>
- [8] MVC простыми словами [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://ru.hexlet.io/blog/posts/chto-takoe-mvc-rasskazyvaem-prostymi-slovami>
- [9] HTML [Электронный ресурс]. – Электронные данные. – Режим доступа: [https://developer.mozilla.org/ru/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/HTML_basics)
- [10] Руководство по RSpec [Электронный ресурс]. – Электронные данные. – Режим доступа: RSpec - Краткое руководство - CoderLessons.com

# ПРИЛОЖЕНИЕ А (обязательное)

## Антиплагиат

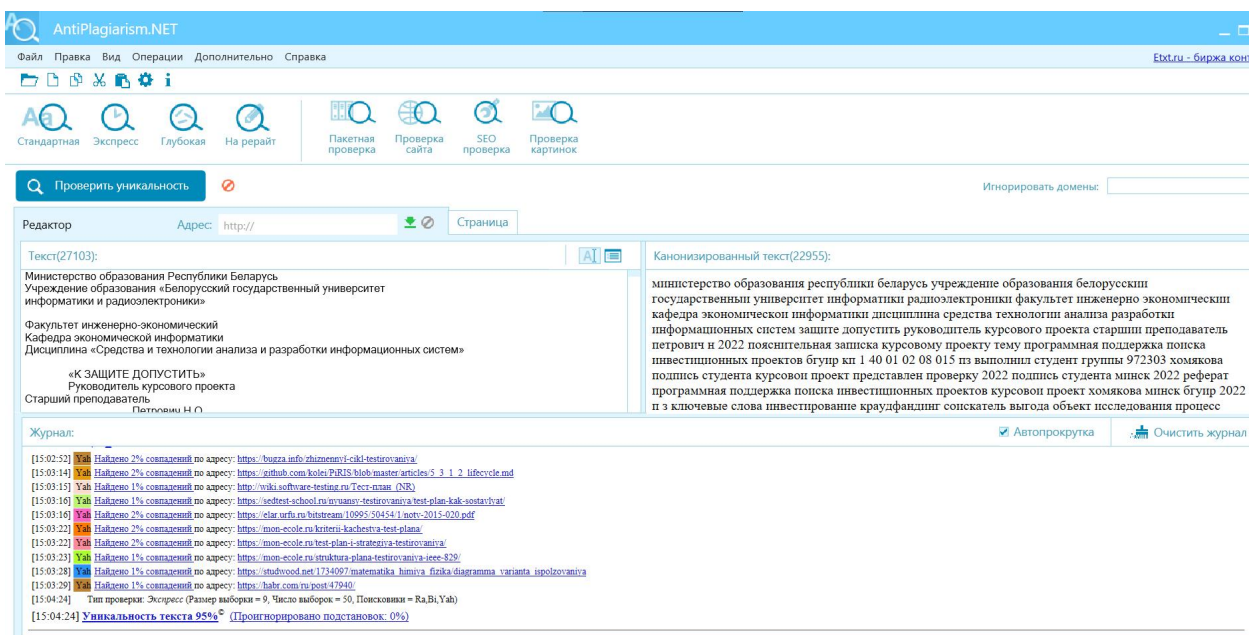


Рисунок А.1— Результат проверки на плагиат в системе  
«AntiPlagiarism.NET»



## ПРИЛОЖЕНИЕ Б (обязательное)

### Модели представления системы

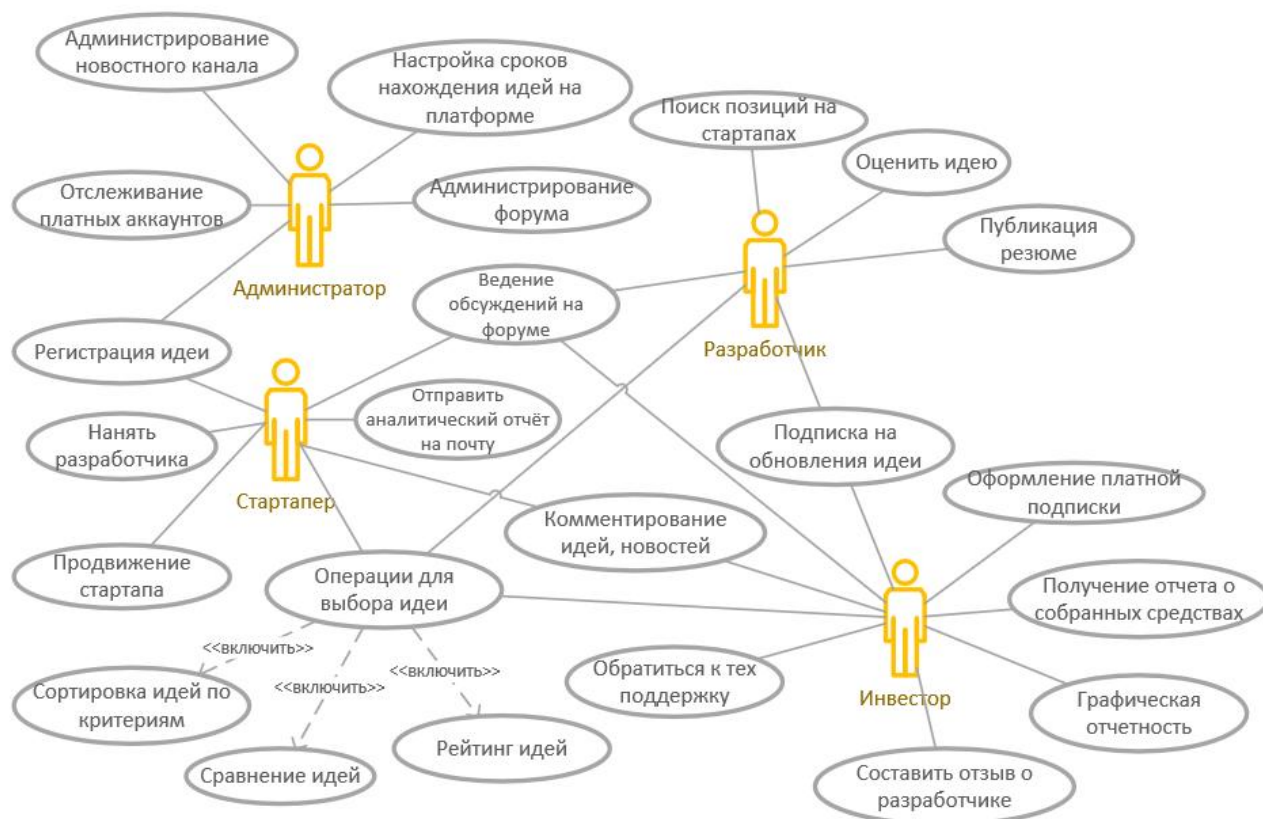


Рисунок Б.1 – Диаграмма вариантов использования (к подпункту 2.2.1)

## Продолжение приложения Б



Рисунок Б.2 – Диаграмма последовательности (к пункту 2.2.2)

## Продолжение приложения Б

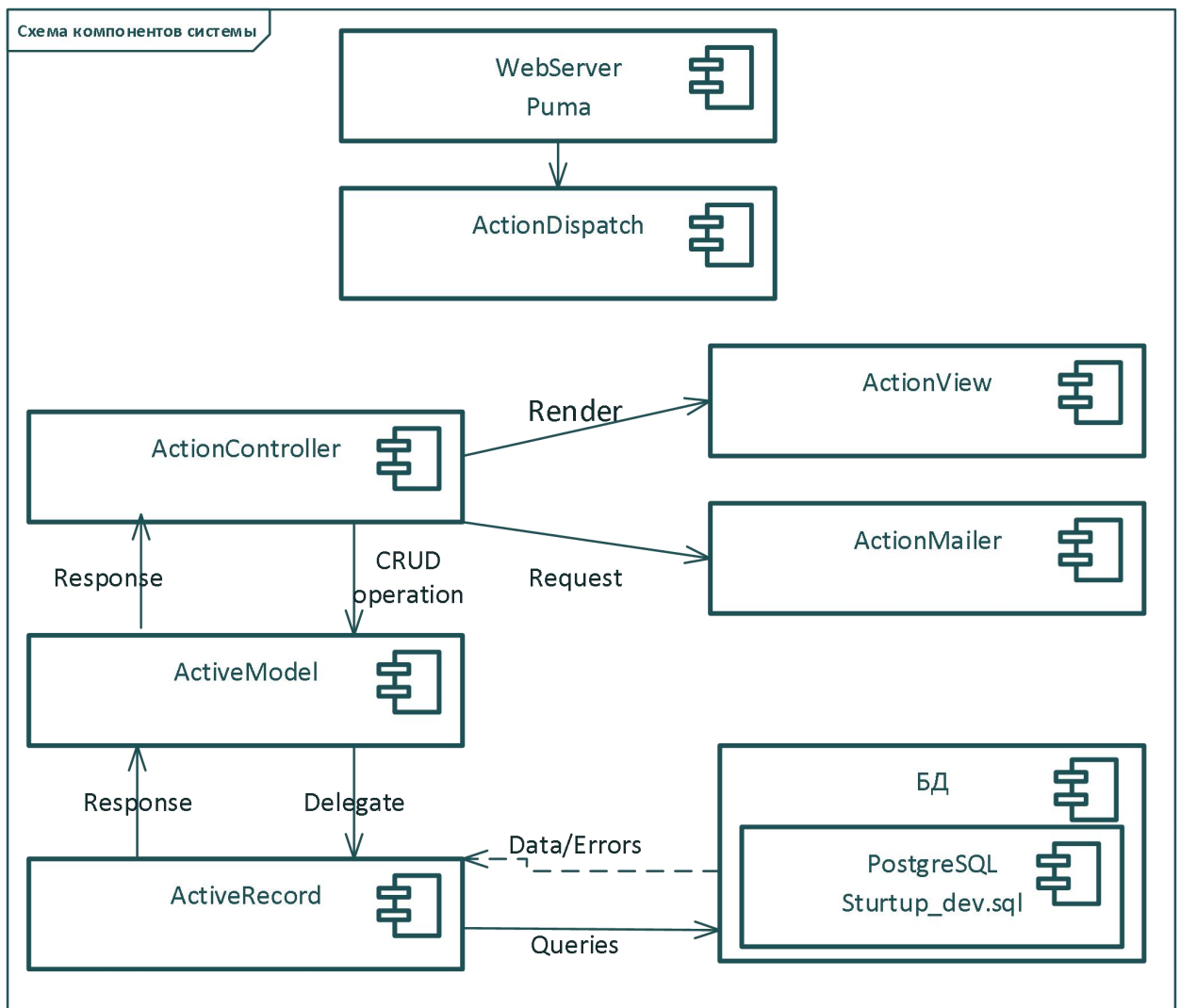


Рисунок Б.3 – Диаграмма компонентов системы (к пункту 2.2.3)

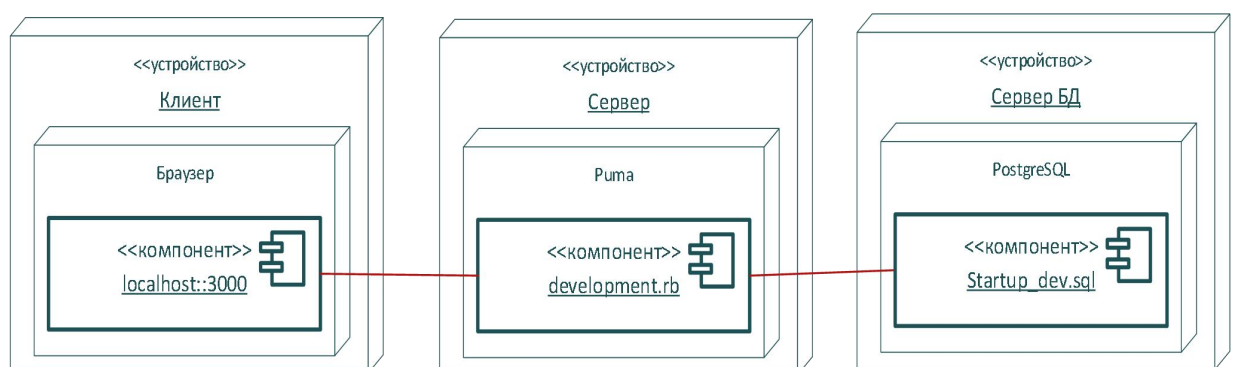


Рисунок Б.4 – Диаграмма развёртывания (к пункту 2.2.4)

## Продолжение приложения Б

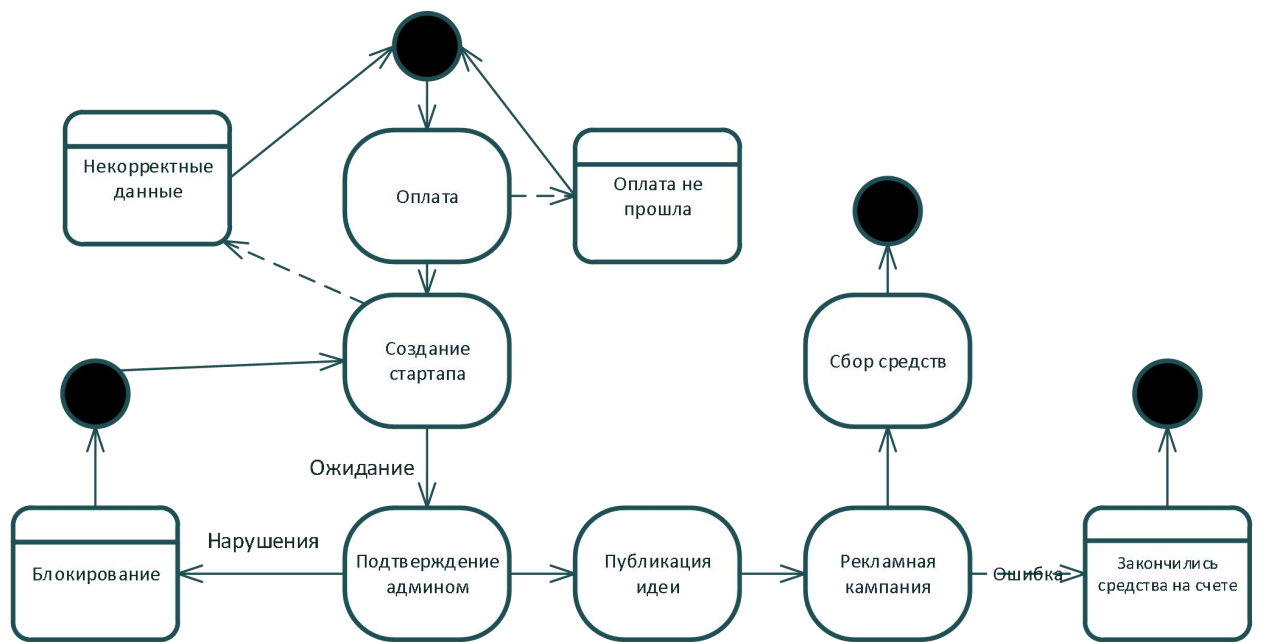


Рисунок Б.5 – Диаграмма состояний (к пункту 2.2.5)

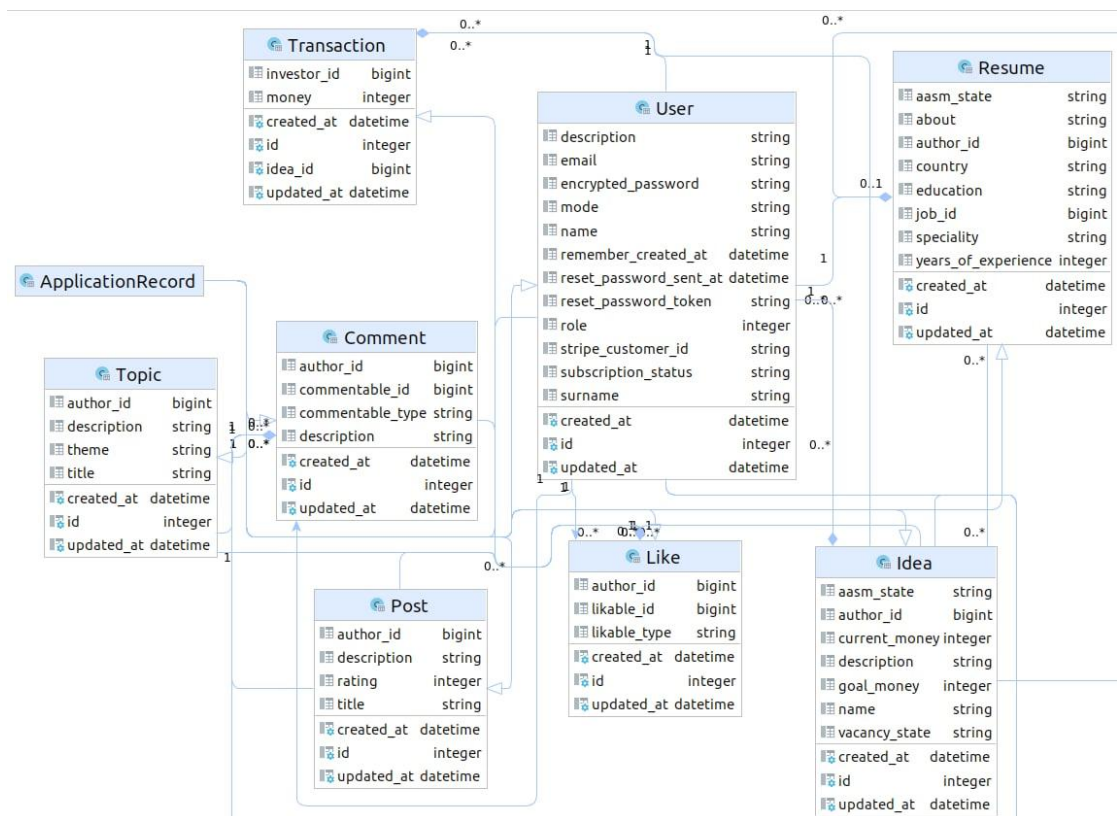


Рисунок Б.6 – Диаграмма классов(к пункту 2.2.6)

## ПРИЛОЖЕНИЕ В (обязательное)

### Схемы бизнес-процессов

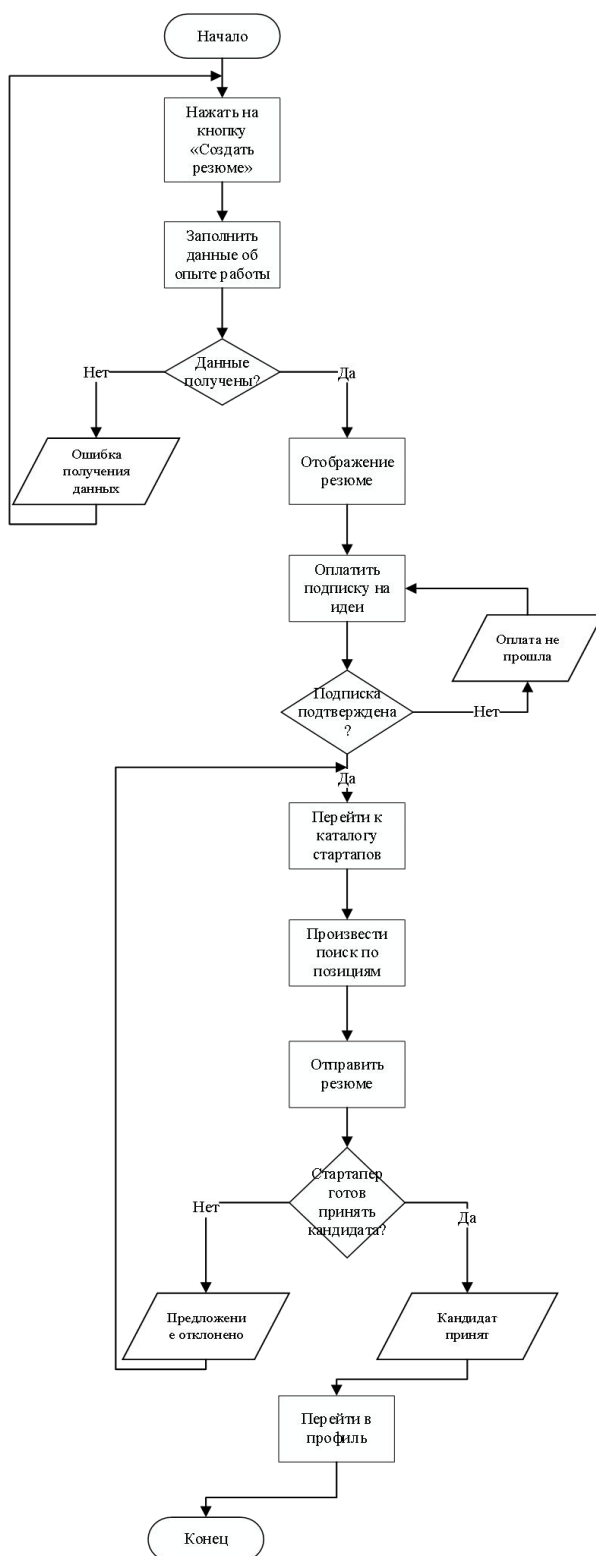
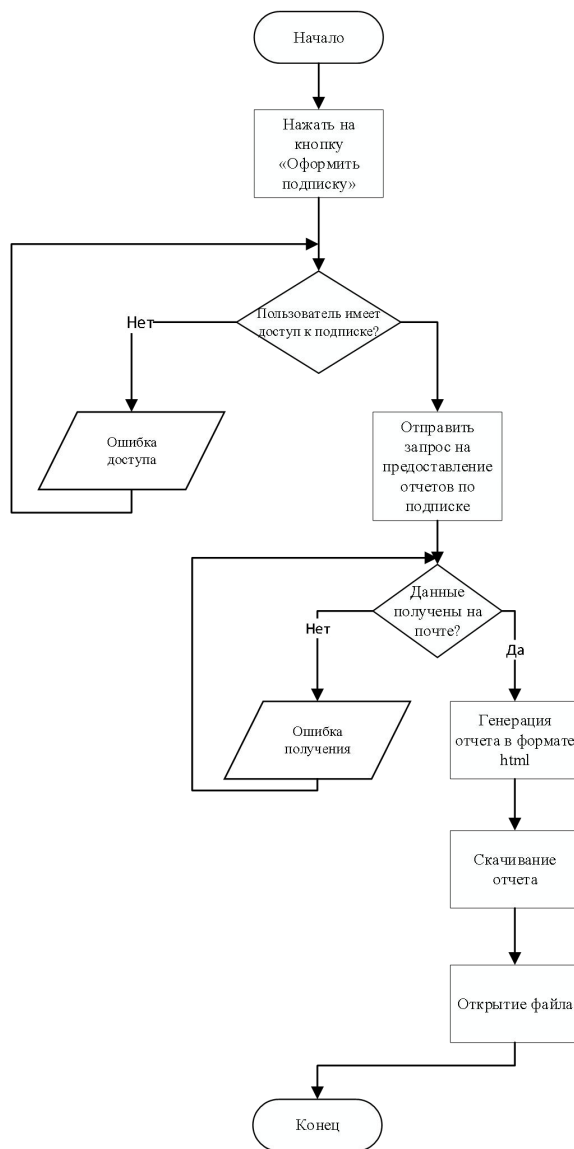


Рисунок В.1 – Схема алгоритма поиска позиции на стартапе



**Рисунок В.2 – Схема алгоритма подписки на получение отчётности**

**ПРИЛОЖЕНИЕ Г**  
**(обязательное)**  
**Листинг скрипта генерации базы данных**

```
ActiveRecord::Schema[7.0].define(version: 2022_10_02_100729) do
  enable_extension "plpgsql"
  create_table "comments", force: :cascade do |t|
    t.string "description"
    t.string "commentable_type"
    t.bigint "commentable_id"
    t.datetime "created_at", null: false
    t.datetime "updated_at", null: false
    t.bigint "author_id", null: false
    t.index ["author_id"], name: "index_comments_on_author_id"
    t.index ["commentable_type", "commentable_id"], name:
      "index_comments_on_commentable"
  end
  create_table "ideas", force: :cascade do |t|
    t.string "name"
    t.string "description"
    t.integer "goal_money"
    t.bigint "author_id"
    t.datetime "created_at", null: false
    t.datetime "updated_at", null: false
    t.string "aasm_state"
    t.string "vacancy_state"
    t.integer "current_money", default: 0
    t.index ["author_id"], name: "index_ideas_on_author_id"
  end
  create_table "ideas_resumes", id: false, force: :cascade do |t|
    t.bigint "idea_id", null: false
    t.bigint "resume_id", null: false
    t.index ["idea_id"], name: "index_ideas_resumes_on_idea_id"
    t.index ["resume_id"], name: "index_ideas_resumes_on_resume_id"
  end
  create_table "likes", force: :cascade do |t|
    t.bigint "author_id"
    t.string "likable_type"
    t.bigint "likable_id"
    t.datetime "created_at", null: false
    t.datetime "updated_at", null: false
    t.index ["author_id"], name: "index_likes_on_author_id"
    t.index ["likable_type", "likable_id"], name: "index_likes_on_likable"
  end
  create_table "posts", force: :cascade do |t|
    t.string "title"
    t.string "description"
    t.integer "rating"
    t.bigint "author_id"
    t.datetime "created_at", null: false
    t.datetime "updated_at", null: false
```

## Продолжение приложения Г

```
t.index ["author_id"], name: "index_posts_on_author_id"
end
create_table "resumes", force: :cascade do |t|
  t.string "speciality"
  t.string "education"
  t.string "country"
  t.integer "years_of_experience"
  t.string "about"
  t.bigint "author_id"
  t.string "aasm_state"
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
  t.bigint "job_id"
  t.index ["author_id"], name: "index_resumes_on_author_id"
  t.index ["job_id"], name: "index_resumes_on_job_id"
end
create_table "subscribers_subscriptions", force: :cascade do |t|
  t.bigint "user_id"
  t.bigint "idea_id"
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
  t.index ["idea_id"], name: "index_subscribers_subscriptions_on_idea_id"
  t.index ["user_id"], name: "index_subscribers_subscriptions_on_user_id"
end
create_table "topics", force: :cascade do |t|
  t.string "theme"
  t.string "title"
  t.string "description"
  t.bigint "author_id"
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
  t.index ["author_id"], name: "index_topics_on_author_id"
end
create_table "transactions", force: :cascade do |t|
  t.bigint "idea_id"
  t.bigint "investor_id"
  t.integer "money"
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
  t.index ["idea_id"], name: "index_transactions_on_idea_id"
  t.index ["investor_id"], name: "index_transactions_on_investor_id"
end
create_table "users", force: :cascade do |t|
  t.string "name", default: "", null: false
  t.string "surname", default: "", null: false
  t.string "description", default: ""
  t.string "email", default: "", null: false
  t.string "encrypted_password", default: "", null: false
  t.string "reset_password_token"
  t.datetime "reset_password_sent_at"
  t.datetime "remember_created_at"
  t.datetime "created_at", null: false
```



## Продолжение приложения Г

```
t.datetime "updated_at", null: false
t.integer "role"
t.string "mode"
t.string "subscription_status", default: "incomplete"
t.string "stripe_customer_id"
t.index ["email"], name: "index_users_on_email", unique: true
t.index ["reset_password_token"], name:
"index_users_on_reset_password_token", unique: true
end

end
```

## ПРИЛОЖЕНИЕ Д (обязательное)

### Листинг алгоритмов реализующих бизнес-логику

```
class UsersController < ApplicationController
  skip_before_action :authenticate_user!
  helper_method :user

  def edit
    authorize user
  end

  def show
    authorize user
    @comment = user.user_comments.build
    @comments = user.user_comments
    @resume = user.resume
    @ideas = Idea.where(author_id: user.id)
    @transactions = user.transactions
  end

  def update
    authorize user
    user.update!(user_params)
    redirect_to user_path(user)
  end

  def user
    @user ||= User.find(params[:id])
  end

  def user_params
    params.require(:user).permit(:name, :surname, :description)
  end
end

class UserMailer < ApplicationMailer
  def subscription_mailer
    @user = User.find(params[:user])
    mail(to: @user.email, subject: 'Weekly news')
  end

  def invitation_mailer
    @user = params[:user]
    @idea = params[:idea]
    mail(to: @user.email, subject: 'You have been invited to startup')
  end

  def cancel_resume_mailer
    @user = params[:user]
    @idea = params[:idea]
    mail(to: @user.email, subject: 'Your resume was canceled')
  end
end
```

## Продолжение приложения Д

```
class User < ApplicationRecord
  devise :database_authenticatable, :registerable,
         :recoverable, :rememberable, :validatable

  enum role: { developer: 0, startuper: 1, investor: 2, admin: 3 }

  has_one :resume, foreign_key: 'author_id', class_name: 'Resume',
  inverse_of: :author, dependent: :destroy
  has_many :user_comments, class_name: 'Comment', as: :commentable
  has_many :likes, foreign_key: 'author_id', class_name: 'Like',
  inverse_of: :author, dependent: :destroy
  has_many :transactions, foreign_key: 'investor_id'
  has_many :comments, foreign_key: 'author_id', class_name: 'Comment',
  inverse_of: :author, dependent: :destroy
  has_and_belongs_to_many :subscriptions, class_name: 'Idea', join_table:
  'subscribers_subscriptions'

  after_create do
    customer = Stripe::Customer.create(email: self.email)
    update(stripe_customer_id: customer.id)
  end

  def author?(obj)
    obj.author_id == id
  end

  def owner?(profile)
    profile.id == id
  end
end

class TransactionsController < ApplicationController
  def create
    @transaction = current_user.transactions.new(transaction_params)
    idea = @transaction.idea
    authorize @transaction

    @transaction.save!
    Idea.update(idea.id, current_money: idea.current_money +
@transaction.money)
    redirect_back fallback_location: root_path
  end

  def transaction_params
    params.require(:transaction).permit(:investor_id, :idea_id, :money)
  end
end

class UserPolicy < ApplicationPolicy
  def index?
    if @record.startuper?
      permission = @user.investor? || @user.admin?
    elsif @record.investor?
```

## Продолжение приложения Д

```
      permission = @user.admin?
    elsif @record.admin?
      permission = false
    end
    permission
  end

  def show?
    if @record.startuper? || @record.investor? || @record.developer?
      permission = true
    elsif @record.admin?
      permission = false
    end
    permission
  end

  def update?
    return @user.owner?(@record) if @record.admin?

    @user.admin? || @user.owner?(@record)
  end

  def destroy?
    update?
  end
end

  source 'https://rubygems.org'
git_source(:github) { |repo| "https://github.com/#{repo}.git" }

ruby '3.1.2'

gem 'aasm'
gem 'bootsnap', require: false
gem 'devise'
gem 'chartkick'
gem 'importmap-rails'
gem 'jbuilder'
gem 'groupdate'
gem 'pg', '~> 1.1'
gem 'puma', '~> 5.0'
gem 'pundit'
gem 'rails', '~> 7.0.4'
gem 'ransack'
gem 'sidekiq'
gem 'sidekiq-cron'
gem 'sprockets-rails'
gem 'stimulus-rails'
gem 'stripe'
gem 'turbo-rails'
gem 'tzinfo-data', platforms: %i[mingw mswin x64_mingw jruby]

group :development, :test do
```

## Продолжение приложения Д

```
gem 'debug', platforms: %i[mri mingw x64_mingw]
gem 'rspec-rails'
gem 'rubocop'
gem 'rubocop-rails'
gem 'rubocop-rspec'
end

group :development do
  gem 'web-console'
end

group :test do
  gem 'capybara'
  gem 'selenium-webdriver'
  gem 'webdrivers'
end

require 'sidekiq/web'
require 'sidekiq/cron/web'

Rails.application.routes.draw do
  devise_for :users

  devise_scope :user do
    get '/users/sign_out' => 'devise/sessions#destroy'
  end

  get '/ideas/:id/attach', to: 'ideas#attach_resume', as: 'attach'
  get '/ideas/:id/accept', to: 'ideas#accept_resume', as: 'accept'
  get '/ideas/:id/cancel_invitation', to: 'ideas#cancel_invitation', as:
'reject'
  get '/ideas/:id/subscribe', to: 'ideas#subscribe', as: 'subscribe'
  get '/ideas/:id/unsubscribe', to: 'ideas#unsubscribe', as: 'unsubscribe'
  get '/compare', to: 'ideas#compare_ideas', as: 'compare'
  post "checkout/create", to: "checkout#create", as: "checkout_create"

  resources :transactions
  resources :webhooks, only: [:create]
  resources :topics
  resources :resumes
  resources :posts
  resources :likes
  resources :comments
  resources :ideas
  resources :users
  root 'pages#index'

  mount Sidekiq::Web => '/sidekiq'
end

class SubscriptionJob < ApplicationJob
  queue_as :default
  def perform(*_args)
```

## Продолжение приложения Д

```
User.find_each do |user|
  UserMailer.with(user: user.id).subscription_mailer.deliver_now if
user.subscriptions.count.positive? && !user.nil?
end
end
end

class IdeaExpirationJob < ApplicationJob
  queue_as :default
  def perform(*_args)
    Idea.where(aasm_state: 'running').find_each do |idea|
      idea.expire! if idea.updated_at + 2.weeks < DateTime.now
    end
  end
end

# frozen_string_literal: true

class PostsController < ApplicationController
  skip_before_action :authenticate_user!
  after_action :verify_authorized

  def index
    @posts = Post.all
    authorize @posts
  end

  def show
    authorize post
  end

  def new
    @post = Post.new
    authorize @post
  end

  def create
    @post = Post.new(post_params)
    authorize @post

    @post.save!
    redirect_to posts_path
  end

  def post
    @post ||= Post.find(params[:id])
  end

  def post_params
    params.require(:post).permit(:title, :description, :author_id)
  end
end

class IdeasController < ApplicationController
  skip_before_action :authenticate_user!
```

## Продолжение приложения Д

```
after_action :verify_authorized, only: %i[index new show create destroy]

def index
  @advertisement = Idea.joins(:author).where(author:
{ subscription_status: :complete }).sample
  @q = Idea.where(aasm_state: 'running').ransack(params[:q])
  @ideas = @q.result(distinct: true).includes(:likes)
  authorize @ideas
end

def new
  @idea = Idea.new
  authorize @idea
end

def show
  @idea = Idea.find(params[:id])
  authorize @idea

  @resumes = @idea.resumes
  @to_compare = []
  @comment = @idea.comments.build
  @transaction = @idea.transactions.build
  @comments = @idea.comments
  @developers = @idea.developers
  @investors = @idea.transactions.to_a.pluck(:investor_id).uniq
end

def create
  @idea = Idea.new(idea_params)
  authorize @idea

  @idea.save!
  redirect_to ideas_path
end

def destroy
  @idea = Idea.find(params[:id])
  authorize @idea

  @idea.destroy!
  redirect_to ideas_path
end

def attach_resume
  @idea = Idea.find(params[:id])
  @idea.resumes << current_user.resume
  redirect_back fallback_location: root_path
end

def compare_ideas
  @first_idea = Idea.find_by(id: params[:f_idea])
```

## Продолжение приложения Д

```
@second_idea = Idea.find_by(id: params[:s_idea])
@first_investors =
@first_idea.transactions.to_a.pluck(:investor_id).uniq.first unless
@first_idea.nil?
  @first_investors = 0 if @first_investors.nil?
  @second_investors =
@second_idea.transactions.to_a.pluck(:investor_id).uniq.first unless
@second_idea.nil?
  @second_investors = 0 if @second_investors.nil?
end

def subscribe
  @idea = Idea.find(params[:id])
  current_user.subscriptions << @idea
end

def unsubscribe
  @idea = Idea.find(params[:id])
  current_user.subscriptions.delete(@idea)
end

def accept_resume
  @idea = Idea.find(params[:id])
  resume = Resume.find(params[:resume])
  @idea.resumes.delete(resume)
  @idea.developers << resume
  resume.job = @idea
  resume.disable!
  redirect_back fallback_location: root_path
end

def cancel_invitation
  @idea = Idea.find(params[:id])
  resume = Resume.find(params[:resume])
  @idea.resumes.delete(resume)
  UserMailer.with(user: resume.author).cancel_resume_mailer.deliver_later
  redirect_back fallback_location: root_path
end

def idea_params
  params.require(:idea).permit(:name, :description, :goal_money, :author_id)
end
```