МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Курсовая работа по экзаменационному предмету  
«Технологии программирования»

Поиск работы на HeadHunter

Работу выполнили: студенты 3 курса 3 группы

Рудин П.И., Кушнеренко В.К.

Руководитель работы: Полещук Х. А.

Воронеж 2019

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc515219537)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc515219538)

[1.1. Глоссарий 6](#_Toc515219539)

[1.2. Анализ существующих решений 6](#_Toc515219540)

[1.2.1. Expense Manager 6](#_Toc515219541)

[1.2.2. CoinKeeper 7](#_Toc515219542)

[1.2.3. EasyMoney 8](#_Toc515219543)

[2. ПОСТАНОВКА ЗАДАЧИ 9](#_Toc515219544)

[3. ТРЕБОВАНИЯ К РАЗРАБАТЫВАЕМОЙ СИСТЕМЕ 10](#_Toc515219545)

[3.1. Требования к интерфейсу 10](#_Toc515219546)

[3.2. Входные/выходные данные 10](#_Toc515219547)

[3.3. Требования к использованию программного средства 10](#_Toc515219548)

[3.4. Средства реализации 10](#_Toc515219549)

[4. ПЛАНИРОВАНИЕ РАБОТ 11](#_Toc515219550)

[4.1. Виды работ, которые необходимо выполнить в процессе разработки программного средства 11](#_Toc515219551)

[4.2. Состав команды, распределение задач по участникам и по времени 12](#_Toc515219552)

[5. ПРОЕКТИРОВАНИЕ 14](#_Toc515219553)

[5.1. Архитектура разрабатываемого программного средства 14](#_Toc515219554)

[5.2. Схема базы данных 14](#_Toc515219555)

[5.3. Диаграмма кооперации 15](#_Toc515219556)

[5.4. Диаграмма активностей 16](#_Toc515219557)

[5.5. Диаграмма последовательностей 16](#_Toc515219558)

[5.6. Диаграмма состояний 17](#_Toc515219559)

[5.7. Диаграмма прецедентов 17](#_Toc515219560)

[5.9. Диаграмма классов 18](#_Toc515219561)

[6. ПЛАН ТЕСТИРОВАНИЯ 19](#_Toc515219562)

[7. ИТОГИ РАБОТЫ 20](#_Toc515219563)

[7.1. Проект интерфейсной части программного средства 20](#_Toc515219564)

[СПИСОК ЛИТЕРАТУРЫ 24](#_Toc515219565)

[Приложение 1. Контроллер аутентификации 25](#_Toc515219566)

[Приложение 2. Контроллер аутентификации 28](#_Toc515219567)

**ВВЕДЕНИЕ**

Зачастую люди часто сталкиваются с необходимостью быстро найти работу. Этот процесс поиска работы может осуществляться на основе информации в газетах, информации в специальных изданиях или в простых объявлениях. Но с развитием современных технологий на первое место выходит сеть Интернет. Именно в сети можно быстро и легко найти работодателя и желаемую вакансию. Процесс поиска соискателем работы через Интернет можно сделать удобным и простым.

Для выполнения курсовой работы по экзаменационному предмету Технологии Программирования была выбрана задача “Поиск работы на HeadHunter”. HeadHunter – крупнейшая компания, которая занимается связью работодателей и работников. Этот процесс осуществляется с помощью ресурса hh.ru, на котором происходит публикация открытых вакансий работодателями.

Реализуемая задача направлена на автоматизацию процесса поиска работы и упрощение работы соискателя с сервисом компании HeadHunter.

Решение к поставленной задаче представляет из себя Android – приложение.

Из анализа статистики ключевых слов в запросах поисковых систем можно сделать вывод о том, что ежедневно большое количество людей в РФ используют сеть Интернет для того, чтобы найти работу. На рынке интернет-рекрутинга России представлены множество проектов, направленные на это.

Компания HeadHunter является лидером на рынке существующих решений и предоставляет открытый API, который позволяет получать информацию из вакансий и резюме, а также работать с этими данными.

**1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

**1.1. Глоссарий проекта**

1. *Соискатель* – лицо, заинтересованное найти работу.

2. *Работодатель* – частное лицо или компания, предлагающая работу.

3. *Вакансия* – незамещённое рабочее место.

4. *Резюме* – документ с информацией о навыках, опыте работы и образовании кандидата.

5. *База данных* – организованная определененным образом совокупность логически связанных данных.

6. *Публичный API* – публичный набор классов и функций , который позволяет осуществлять взаимодействие между различным программным обеспечением.

7. *Экран Android-приложения* – сущность приложения с пользовательским интерфейсом.

8. *RecyclerView-список* – компонент пользовательского интерфейса, позволяющий создать прокручиваемый список.

9. *Кэширование* – процесс сохранения в области информации, которая может быть запрошена с максимальной вероятностью.

10. *Сервер* – специализированное оборудование, которое поддерживает работу серверное программного обеспечения.

11. *Сервис* – услуги, предоставляемые с помощью специальных программ.

12. *Рекрутмент* – процесс привлечения и подбора персонала.

13. *Скелет приложения* – набор классов, необходимых для работы приложения.

14. *Activity* – компонент приложения, представленный экраном.

15. *Fragment* – модульная часть Activity.

16. *Xml - файлы* – файлы расширения .xml.

**1.2. Анализ существующих решений**

По информации сервиса Яндекс.Подбор слов ежемесячно ключевое слово “работа” вводится в поисковый запрос более чем 66 миллионов раз за один месяц. Если добавить к этому результаты поиска работы в конкретной области и работы в конкретном городе или области, то количество запросов будет в разы выше. Не стоит забывать и о других поисковых системах, которыми пользуются в нашей стране. Это говорит о том, что поиск работы с помощью средств сети Интернет очень популярен в РФ.

Проведём анализ существующих решений.

Аналогами разрабатываемого приложения являются мобильные приложения “Яндекс.Работа – вакансии”, “Worki – работа рядом с домом”, “Работа Superjob: поиск вакансий и создание резюме”, а также собственное приложение компании HeadHunter – “Поиск работы на hh.Вакансии рядом с домом”.

Представленные аналоги являются самыми популярными в российском сегменте поиска работы с помощью мобильных приложений и занимают первые позиции в PlayMarket при использовании ключевого слова “работа”.

Они обладают большим доверием со стороны пользователей, так как существуют на рынке много лет и имеют обширную базу зарегистрированных соискателей и работодателей. Несмотря на это, они имеют некоторые существенным недостатки. Рассмотрим подробнее.

Преимущества систем-аналогов, которые могут быть конкурентами разрабатываемой системы:

* возможность регистрации из приложения
* возможность настройки приложения
* заслуженное доверие пользователей
* известность, по сравнению с разрабатываемой системой

Недостатки систем-аналогов, которые могут быть конкурентами разрабатываемой системы:

* навязчивая реклама
* наличие платного контента
* постоянные оповещения на почту
* постоянные push-уведомления

Разрабатываемое в ходе работы решение позволит избежать навязчивой рекламы в приложениях – аналогах.

**2. ПОСТАНОВКА ЗАДАЧ**

Цель выполнения курсовой работы: создать Android приложение, которое:

* позволяет загрузить трудовое резюме
* позволяет проверить трудовое резюме на ошибки
* позволяет получить подборку интересующих вакансий по выбранным критериям
* позволяет исключать неподходящие вакансии из подборки
* позволяет откликаться на подходящую вакансию, то есть осуществлять дальнейшее взаимодействие с работодателем
* позволяет получить помощь по программе
* позволяет получить контакты разработчиков программы

**3. ТРЕБОВАНИЯ К ПРИЛОЖЕНИЮ**

Теперь перейдём к требованиям, которые предъявляются к приложению.

**3.1. Средства, используемые для реализации приложения**

1. Мобильное устройство с операционной системой Android 7.0 и выше

2. Язык программирования Java

3. IDE Android Studio

4. СУБД SQLite

5. API компании HeadHunter

**3.2. Средства, используемые для проведения командной работы над приложением**

1. Система контроля версий Git

2. Программное обеспечение Skype

3. Корпоративный мессенджер Slack

4. Система контроля требований Trello

5. Платформа для визуализации Miro

**4. ПЛАН ВЫПОЛНЕНИЯ РАБОТЫ**

**4.1. Основные этапы выполнения работы**

Процесс выполнения работы включает в себя следующие основные этапы:

* Разработка документации к приложению
* Разработка приложения
* Тестирование приложения

Каждый из этапов содержит в себе множество подзадач. Рассмотрим этапы по порядку.

Этап разработки документации к приложению включает в себя:

1. Постановка задач
2. Установка сроков реализации проекта и документации
3. Проведение анализа предметной области
4. Проведение анализа средств, используемых для создания приложений
5. Формирование глоссария
6. Создание диаграммы Ганта
7. Создание модель проекта
8. Создание технических диаграмм приложения
9. Создание прототипа технического задания
10. Создание отчётного документа по ролям
11. Создание итогового документа по работе
12. Создание проекта в системе Git
13. Загрузку результатов разработки документации в Git
14. Оформление страницы проекта в сервисе GitHub

Этап разработки приложения включает в себя:

1. Подготовку используемых для разработки средств к работе
2. Проектирование Базы Данных
3. Подключение Базы Данных
4. Создание скелета приложения
5. Настройка подключения к HeadHunter API
6. Реализация модулей, которые отражены в модели проекта
7. Реализация вывода результатов для пользователя
8. Загрузку результатов разработки в Git

Этап тестирования приложения включает в себя:

1. Создание прототипа документа, по которому будет происходить тестирование приложения
2. Проведение тестирования каждого модуля
3. Выявление ошибок
4. Проверка выполнения пользовательских сценариев
5. Создание документа по результатам тестирования
6. Принятие решения о соответствии приложения предъявленным требованиям

**4.2. Команда разработки и распределение задач**

Курсовую работу выполнила команда:

* Рудин Павел, 3.1 группа
* Кушнеренко Виктор, 3.1 группа

Диаграмма Ганта, представленная на рисунке 1, содержит в себе информацию о сроках исполнения и задачах, которые были выполнены исполнителями в рамках подготовки курсовой работы командой разработки.

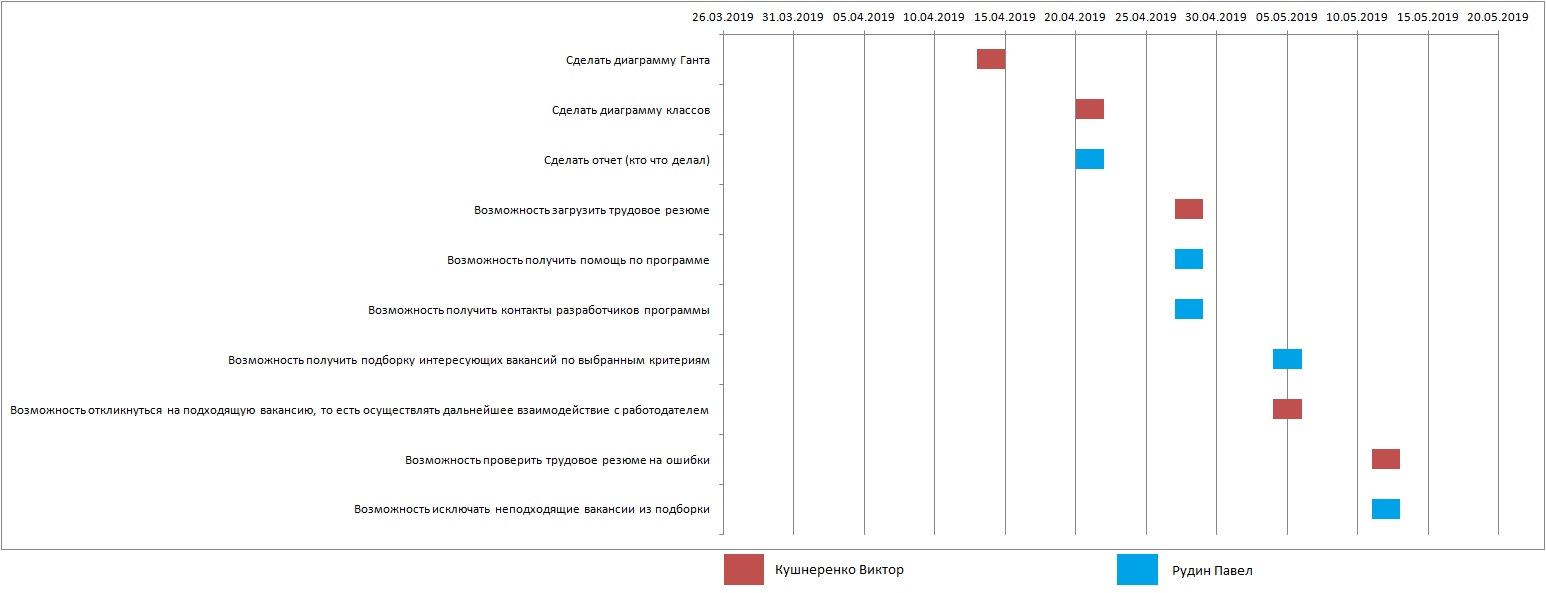


Рис. 1 Диаграмма Ганта с этапами подготовки курсовой работы

**5. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ**

**5.1. Общая модель**

Разрабатываемое в ходе работы приложение состоит из:

* Frontend-части
* Backend-части
* Базы данных

Frontend-часть состоит из нескольких Activity, Fragment и специальных xml-файлов.

Backend-часть состоит из следующих модулей:

1. Работы с авторизацией
2. Взаимодействия с резюме
3. Тестирования
4. Работы с базой данных
5. Работы с API
6. Состоящего из Activity

Используется база данных SQLite, подключенная к решению.

Общая модель разрабатываемого приложения представлена на рисунке 2.

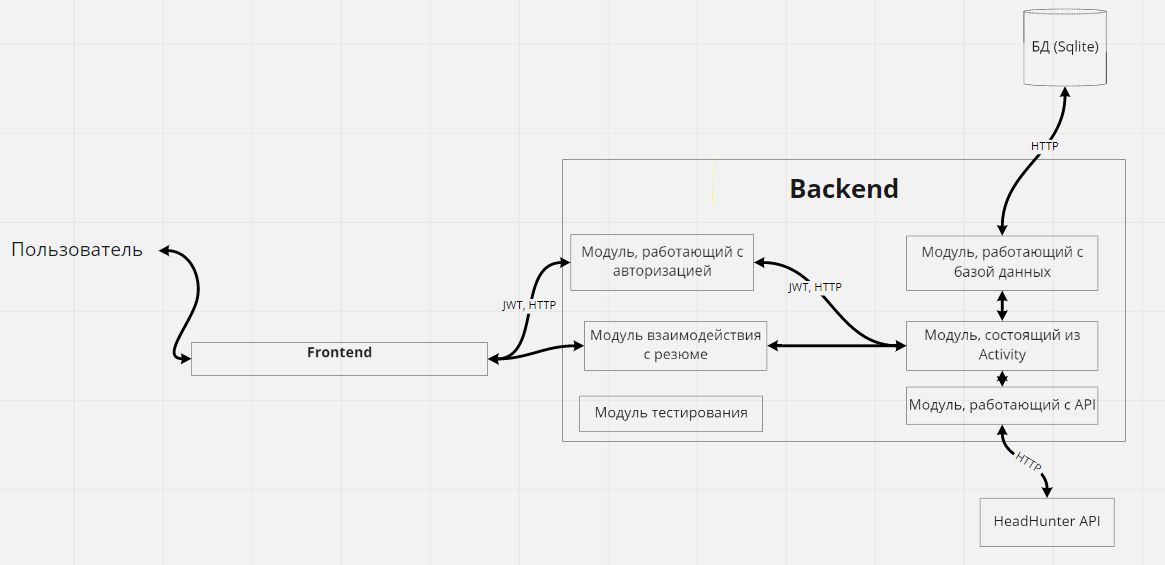
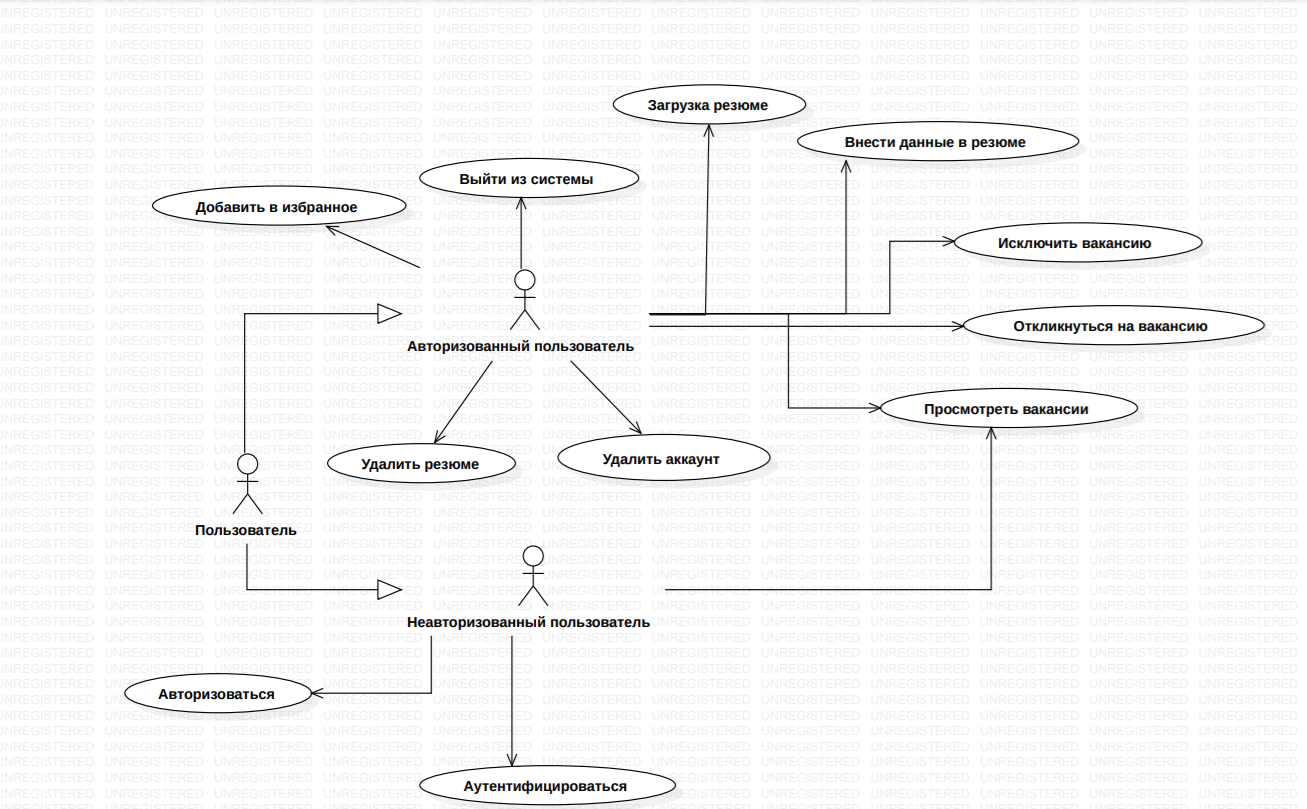


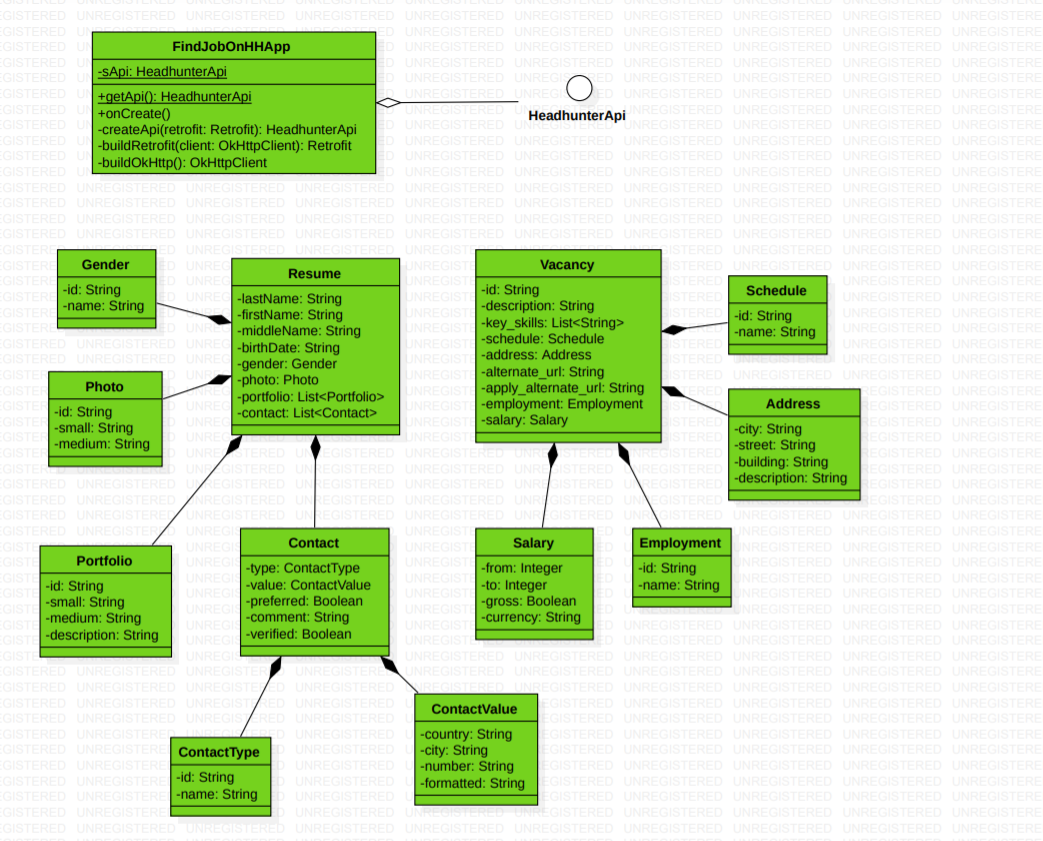
Рис. 2 Общая модель разрабатываемого приложения

**5.2. Схема базы данных**

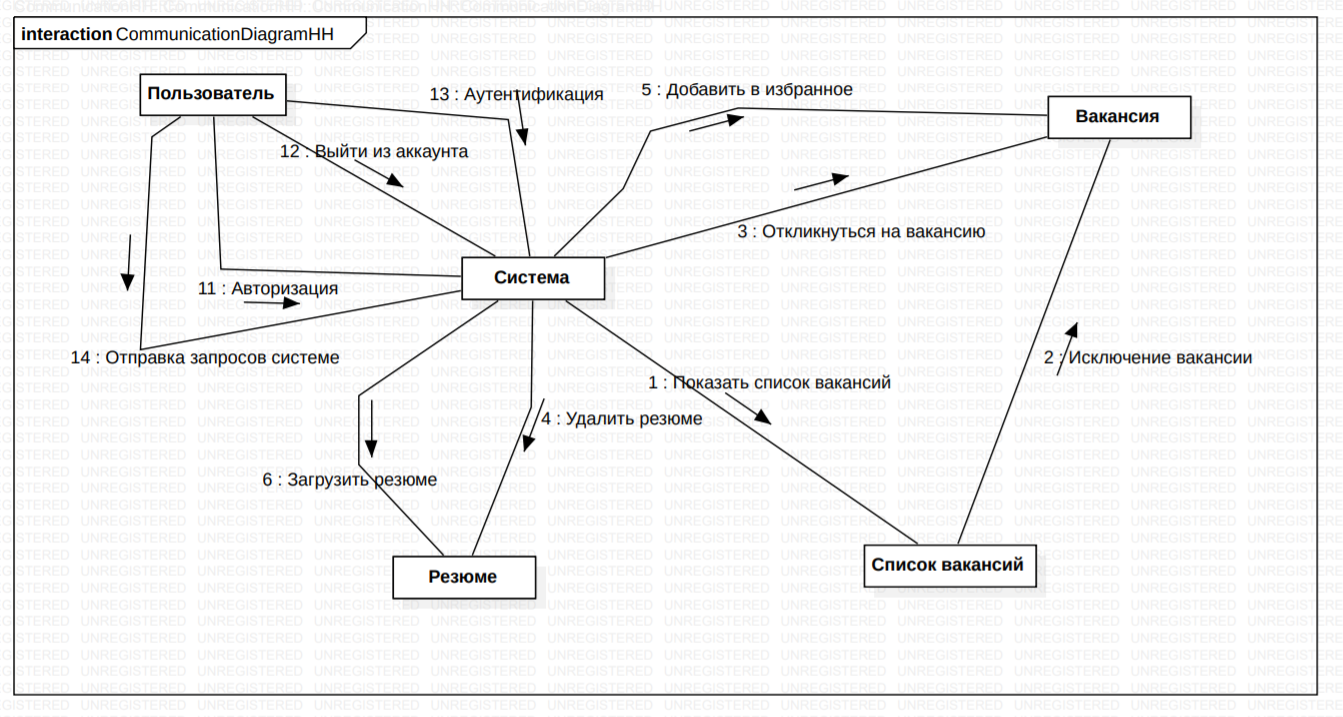
**5.3. Диаграмма прецедентов**

****

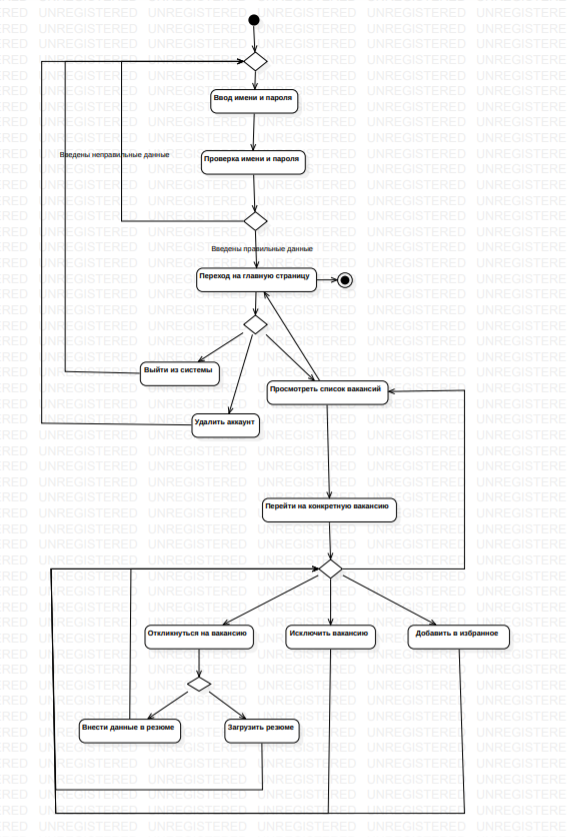
**5.4. Диаграмма классов**

****

**5.5. Диаграмма коммуникаций**

****

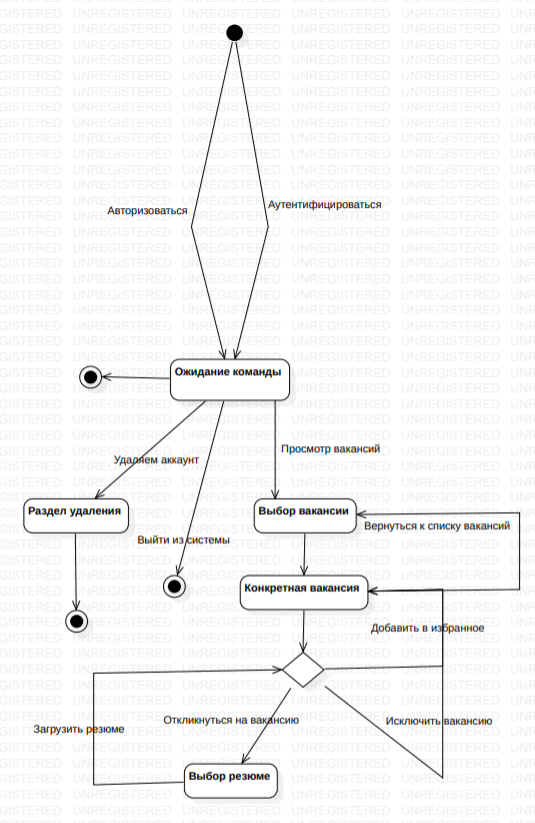
**5.6. Диаграмма активности**

****

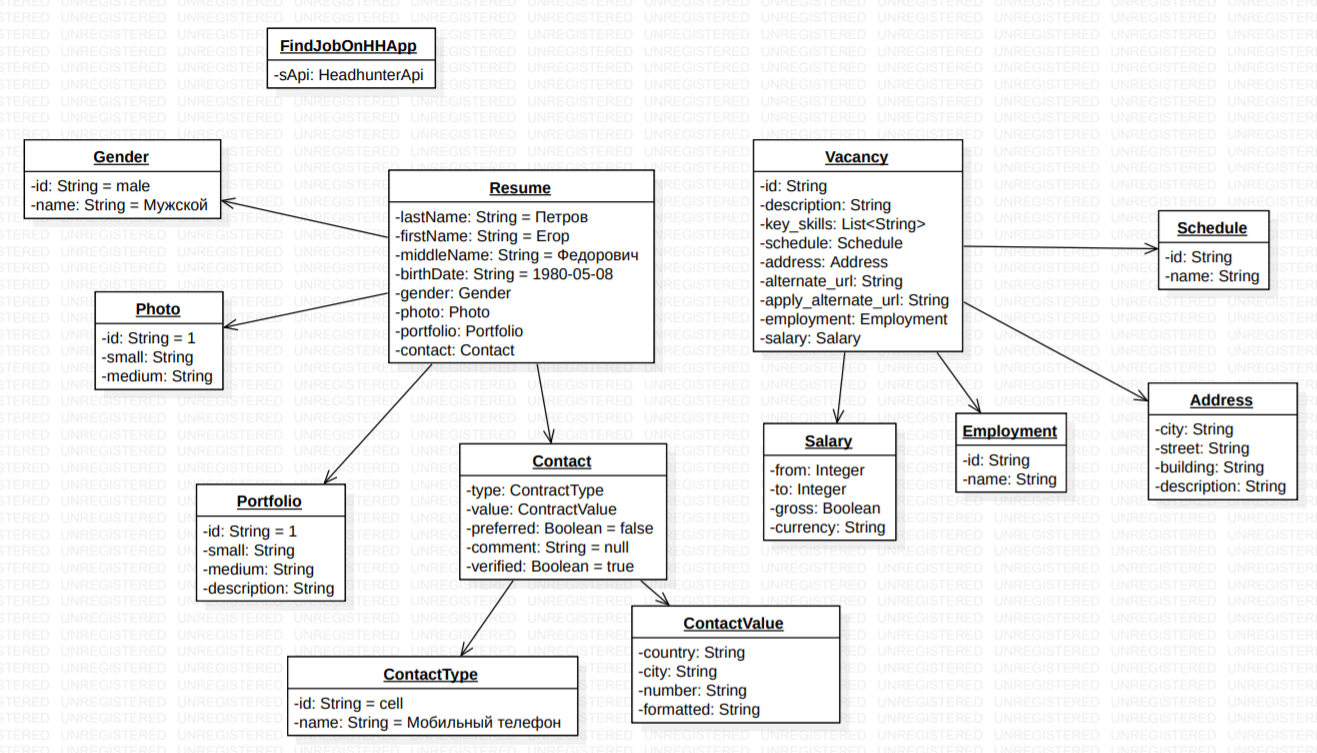
**5.7. Диаграмма последовательностей**

****

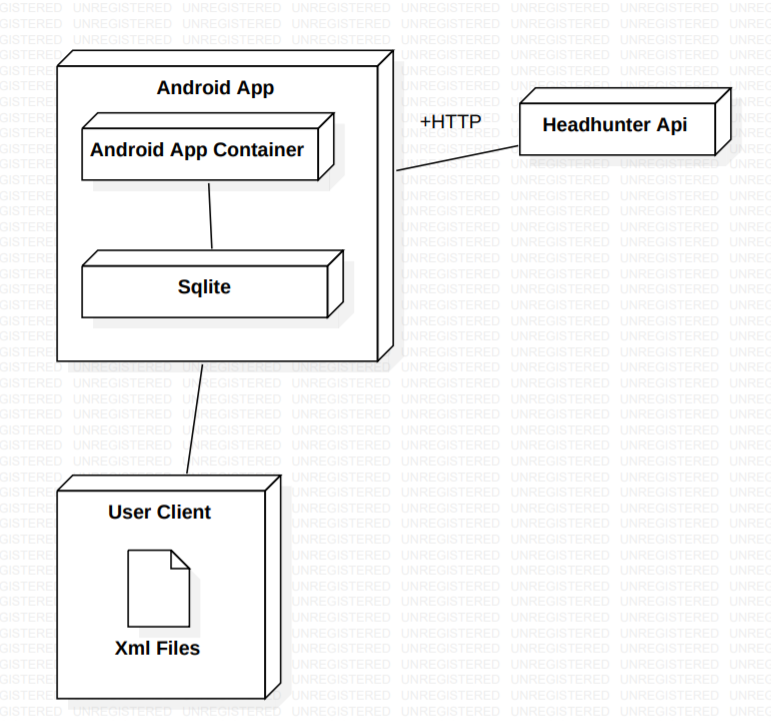
**5.8. Диаграмма состояний**

****

**5.9. Диаграмма объектов**

****

**5.10. Диаграмма развёртывания**

****

**7. ИТОГИ ВЫПОЛНЕНИЯ РАБОТЫ**

В результате работы над проектом была реализована клиентская и серверная часть. Описание работы приведено ниже.

**7.1. Проект интерфейсной части программного средства**

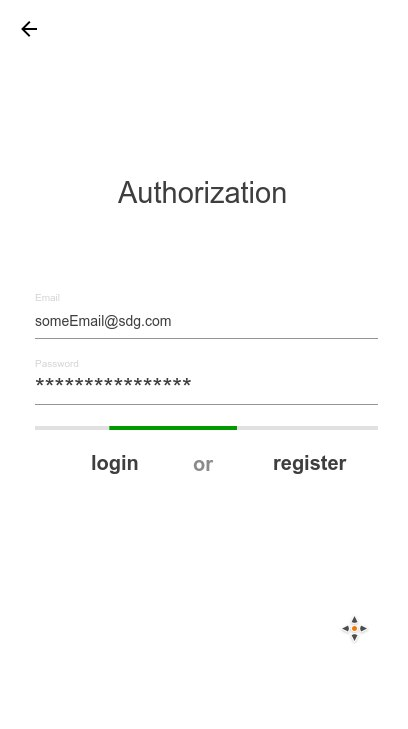
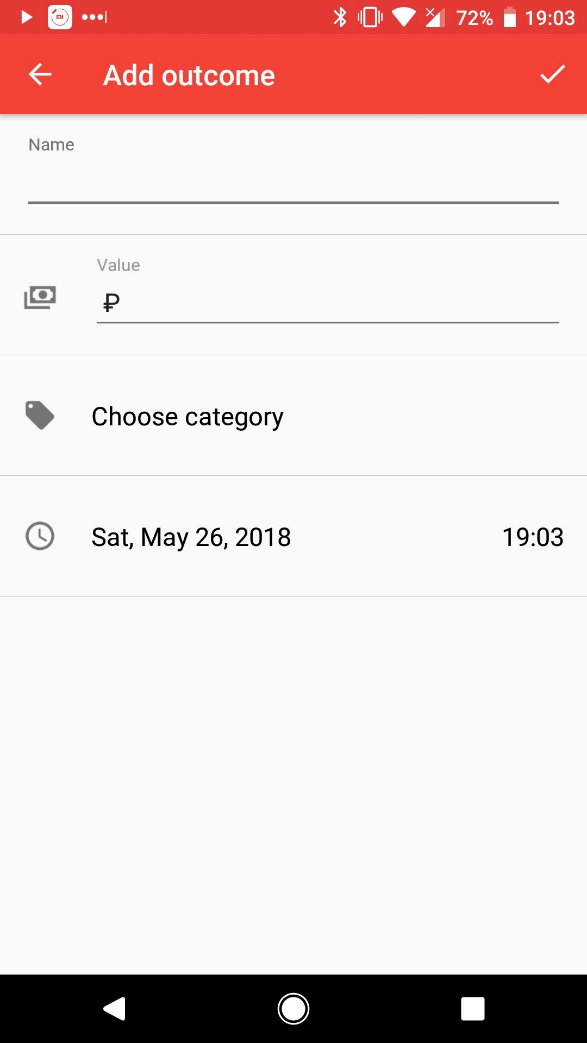
На рис 13 представлен стартовый экран приложения для

Рис. 14 Экран добавления расхода

Рис. 13 Экран авторизации

неавторизованного пользователя. Он имеет возможность войти в систему или зарегистрироваться.

На рис 14 представлен экран добавления расхода. Реализована возможность добавления названия расхода, категории, даты, суммы расхода. Экран добавления дохода аналогичен.

На рис 15 представлен главный экран авторизованного пользователя. На нем отображается список доходов и расходов, а также меню.

На рис 16 представлен экран, на котором отображается список категорий данного пользователя. Реализована возможность редактирования, удаления и добавления категории.

На рис 17 представлен экран добавления цели. Реализована возможность добавления названия цели, даты, суммы, периодичности напоминания.

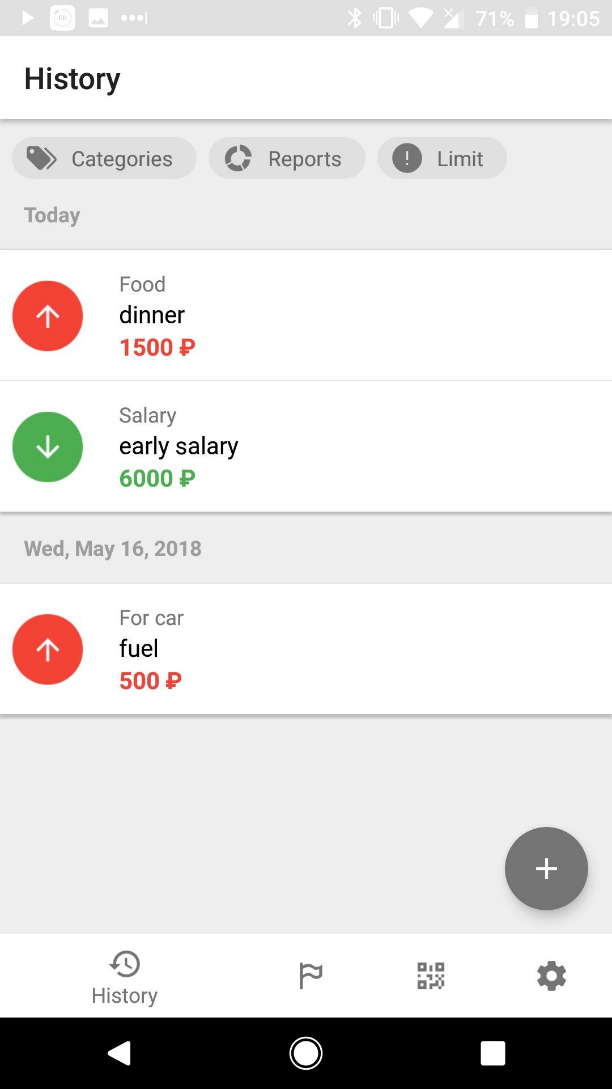
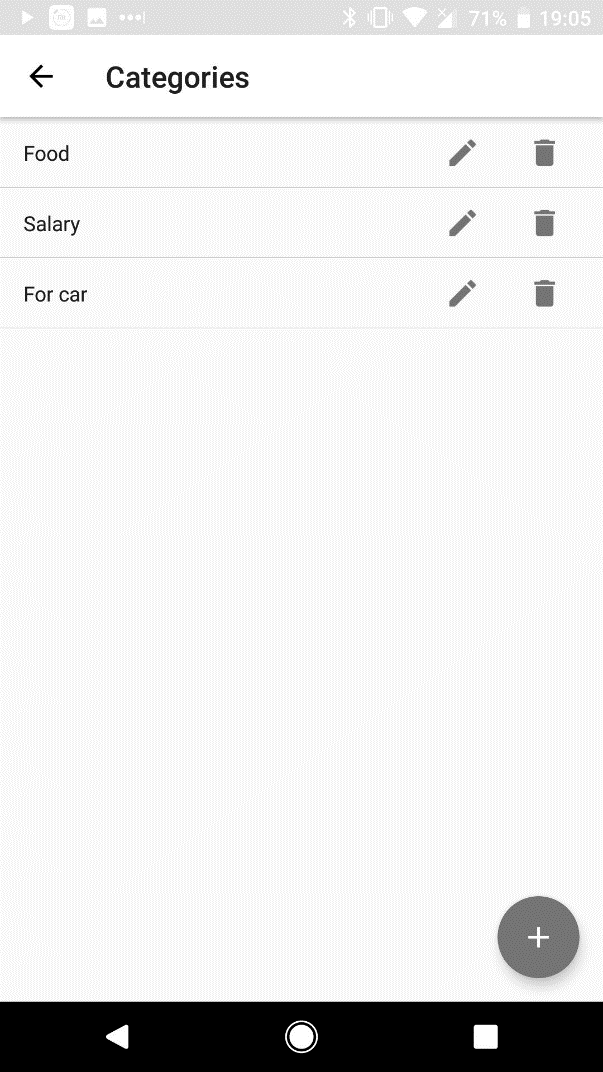
На рис 18 представлен экран, отображающий фотографии чеков.

Рис. 16 Экран категорий

Рис. 15 Главный экран

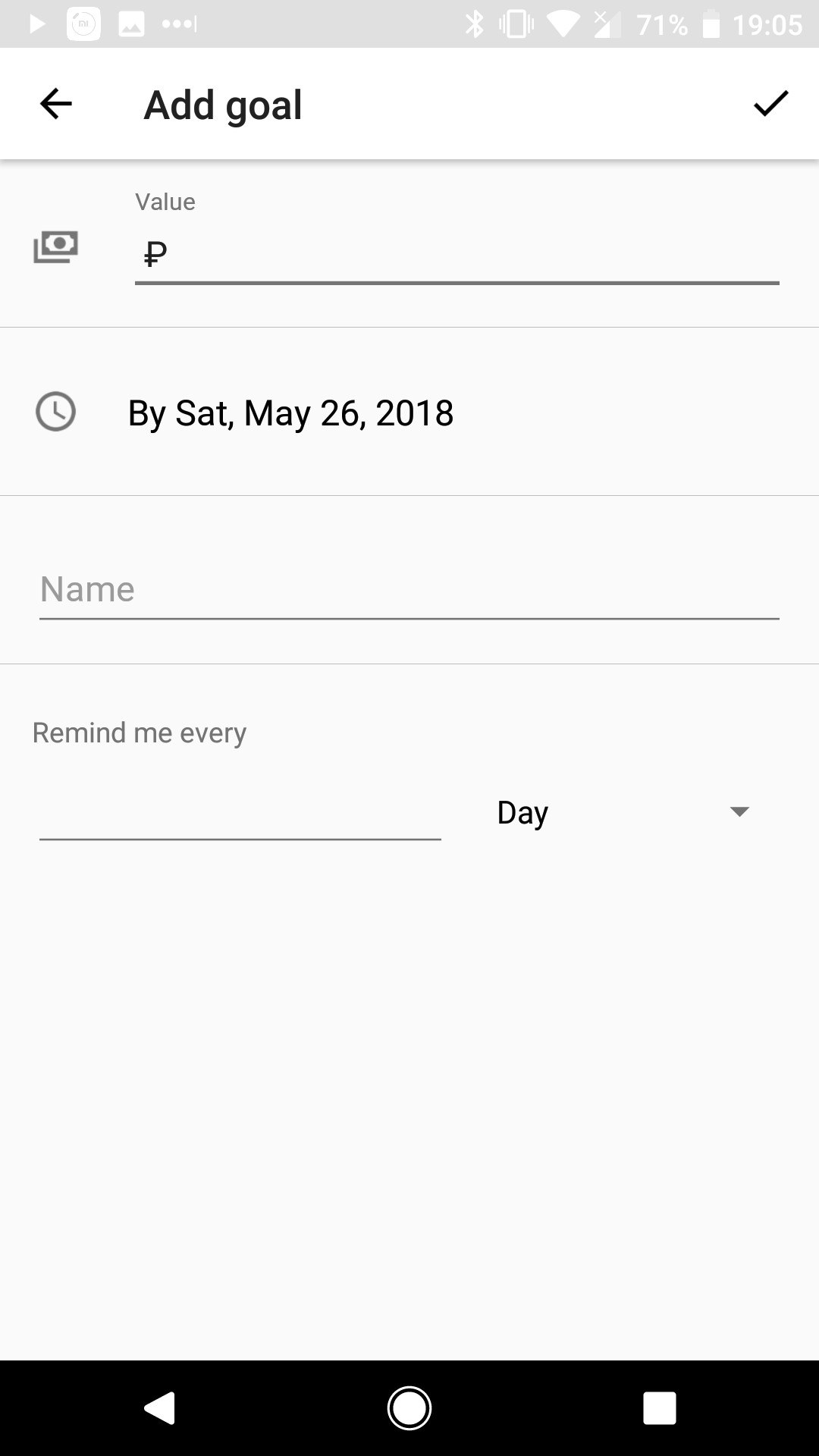


Рис. 18 Экран целей

Рис. 17 Экран добавления цели

**ЗАКЛЮЧЕНИЕ**

В результате написания курсовой работы было разработано Android приложение «Личная бухгалтерия». В ходе реализации были решены следующие задачи:

1. Проанализированы существующие аналоги, их характеристики, возможности.
2. Обеспечено уведомление пользователя о превышении расходов.
3. Обеспечено отображение доходов и расходов на графике в различных вариантах.
4. Реализовано напоминание пользователю о целях;
5. Обеспечено xранение данных в СУБД;
6. Обеспечена синхронизация данных с сервером;
7. Создана пользовательская документация.

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. Документация Java 8 (<http://docs.oracle.com/javase/8>)
2. Герберт Шилдт. Java 8. Полное руководство 9-е издание. Пер. с англ.: – М: Издательский дом «Вильямс», 2015

**Приложение 1. Контроллер аутентификации сервера**

**Приложение 2. Контроллер аутентификации клиента**