

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет *компьютерных наук*

Кафедра *программирования и информационных технологий*

Курсовой проект по дисциплине

*Технологии программирования*

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ «WEATHERHUB»

**Студенты**

\_\_\_\_\_ Н.А. Селиверстов,  
4 курс бакалавриат, д/о

\_\_\_\_\_ Д.И. Демьянов,  
4 курс бакалавриат, д/о

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

**Преподаватели**

\_\_\_\_\_ В.С. Тарасов, старший  
преподаватель

\_\_\_\_\_ К.В. Сиволапов,  
преподаватель

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Воронеж 2021 г.

## Содержание

Содержание .....	2
Введение .....	4
1. Постановка задачи.....	5
1.1. Цели разработки Приложения .....	5
1.2. Функциональность приложения .....	5
2. Анализ предметной области.....	6
2.1. Обзор аналогичных решений.....	6
2.1.1. Яндекс.Погода .....	6
2.1.2. Gismeteo Lite .....	7
2.2. Анализ поставленной задачи .....	9
2.2.1. Варианты использования Приложения.....	9
2.2.2. Действия с Приложением.....	10
2.2.3. Состояния Приложения.....	11
2.2.4. Взаимодействие компонентов Приложения .....	11
2.2.5. Последовательность исполняемых действий Приложения .....	12
2.2.6. Структура базы данных Приложения .....	12
2.2.7. Модель бизнес процесса.....	13
3. Реализация.....	14
3.1. Используемые средства .....	14
3.1.1. Фреймворк Java Spring Boot.....	14
3.1.2. Swagger UI.....	14
3.1.3. Язык программирования Swift .....	15
3.1.4. СУБД PostgreSQL.....	15
3.1.5. Git и GitHub .....	15

3.1.6. Сервис Firebase Analytics .....	15
3.1.7. Сервис SendGrid .....	16
3.2. Развертывание на удаленном сервере .....	17
4. Графический интерфейс .....	19
4.1. Экран просмотра прогноза .....	19
4.2. Экран просмотра списка избранных местоположений .....	20
4.3. Экран поиска местоположения по названию .....	21
4.4. Экран выбора местоположения на карте .....	21
4.5. Экран регистрации .....	22
4.6. Экран авторизации .....	23
4.7. Экраны смены пароля .....	24
5. Перспективы развития Приложения .....	26
Заключение .....	27
Список использованных источников .....	28
Приложение 1 .....	29

## Введение

Во все времена была актуальна задача предсказания погоды, так как от нее зависит планирование многих повседневных активностей. Для решения этой задачи в современном мире разработано большое количество различных сервисов, реализующих алгоритмы предсказания погоды и формирующих ее прогнозы.

Многие сервисы обладают преимуществами над своими конкурентами при составлении прогнозов для определенных местоположений, поэтому люди, желающие получить наиболее точный результат, вынуждены использовать несколько приложений, каждое из которых предоставляет данные прогнозов, составленные одним конкретным сервисом. В связи с этим люди сталкиваются с рядом неудобств при просмотре прогнозов погоды:

- необходимость запуска нескольких приложений и переключения между ними для сравнения данных прогнозов;
- независимость списков избранных местоположений для каждого используемого приложения;
- расход ресурсов устройства для установки и запуска нескольких приложений;
- необходимость запоминания данных аккаунта для каждого используемого приложения.

Для решения перечисленных выше проблем, было принято решение разработать мобильное приложение, выполняющее агрегацию данных прогнозов, полученных из нескольких сервисов, и представление результатов пользователям с учетом их предпочтений.

## **1. Постановка задачи**

### **1.1. Цели разработки Приложения**

Целью курсового проекта является разработка клиент-серверного приложения для платформы iOS, которое удовлетворяет установленным в техническом задании требованиям и направлено на достижение следующих целей:

- упрощение пользовательского сценария просмотра прогнозов погоды, выполненных различными сервисами;
- выявление пользователем сервиса, предоставляющего наиболее точные прогнозы погоды для избранных им местоположений;
- экономия ресурсов мобильного устройства пользователя.

### **1.2. Функциональность приложения**

Для достижения поставленных целей Приложение должно предоставлять пользователям следующие возможности:

- просмотр прогнозов погоды, выполненных различными сервисами, на несколько дней вперед от текущей даты для выбранного пользователем местоположения;
- быстрый доступ к прогнозам погоды для избранных пользователем местоположений;
- регистрация пользователей по адресу электронной почты для доступа к списку избранных местоположений с различных устройств.

## 2. Анализ предметной области

### 2.1. Обзор аналогичных решений

Существует большое количество мобильных приложений для просмотра прогноза погоды. Однако прямых аналогов у приложения WeatherHub (далее - Приложение) нет. Каждое из приложений для просмотра прогноза погоды предоставляет доступ только к одному сервису, в то время как WeatherHub – сразу к нескольким.

Рассмотрим два популярных приложения для просмотра прогноза погоды.

#### 2.1.1. Яндекс.Погода

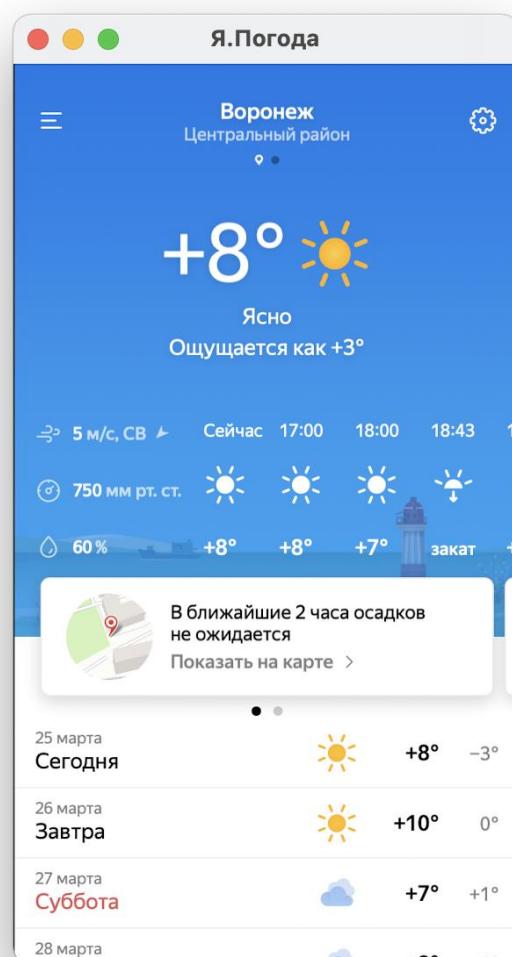


Рисунок 1 - пользовательский интерфейс приложения «Яндекс Погода»

Приложение «Яндекс Погода» обладает следующими особенностями:

- сохранение местоположений в список избранных для быстрого просмотра впоследствии;
- возможность авторизации для синхронизации списка избранных местоположений на разных устройствах;
- просмотр прогноза погоды с шагом в 1 час;
- визуализация осадков на карте;
- отображение прогнозов с часовой и суточной детализацией на одном экране;
- отображение подробной информации о погоде: влажность, давление, и т. д.

### 2.1.2. Gismeteo Lite



Рисунок 2 - Пользовательский интерфейс приложения Gismeteo Lite

Приложение «Gismeteo lite» обладает следующими особенностями:

- отображение информации о погоде на текущий момент в нескольких сохраненных населенных пунктах;
- при открытии приложения пользователь вместо прогноза погоды видит список сохраненных местоположений, для просмотра прогноза погоды необходимо выбрать населенный пункт, то есть совершить дополнительное действие;
- большую часть основного экрана приложения занимает единственное число с текущей температурой – пространство экрана используется нерационально;
- на основном экране помимо температуры и вида осадков отображается большое количество дополнительной информации, выделить нужную из таблицы с первого взгляда сложно.



## 2.2. Анализ поставленной задачи

### 2.2.1. Варианты использования Приложения

Варианты использования включают в себя ключевые сценарии использования Приложения клиентами и администратором.

Варианты использования Приложения приведены на рисунке 3.



Рисунок 3 - Варианты использования

### 2.2.2. Действия с Приложением

Действия, выполняемые в ходе работы Приложения приведены на диаграмме активностей на рисунке 4.

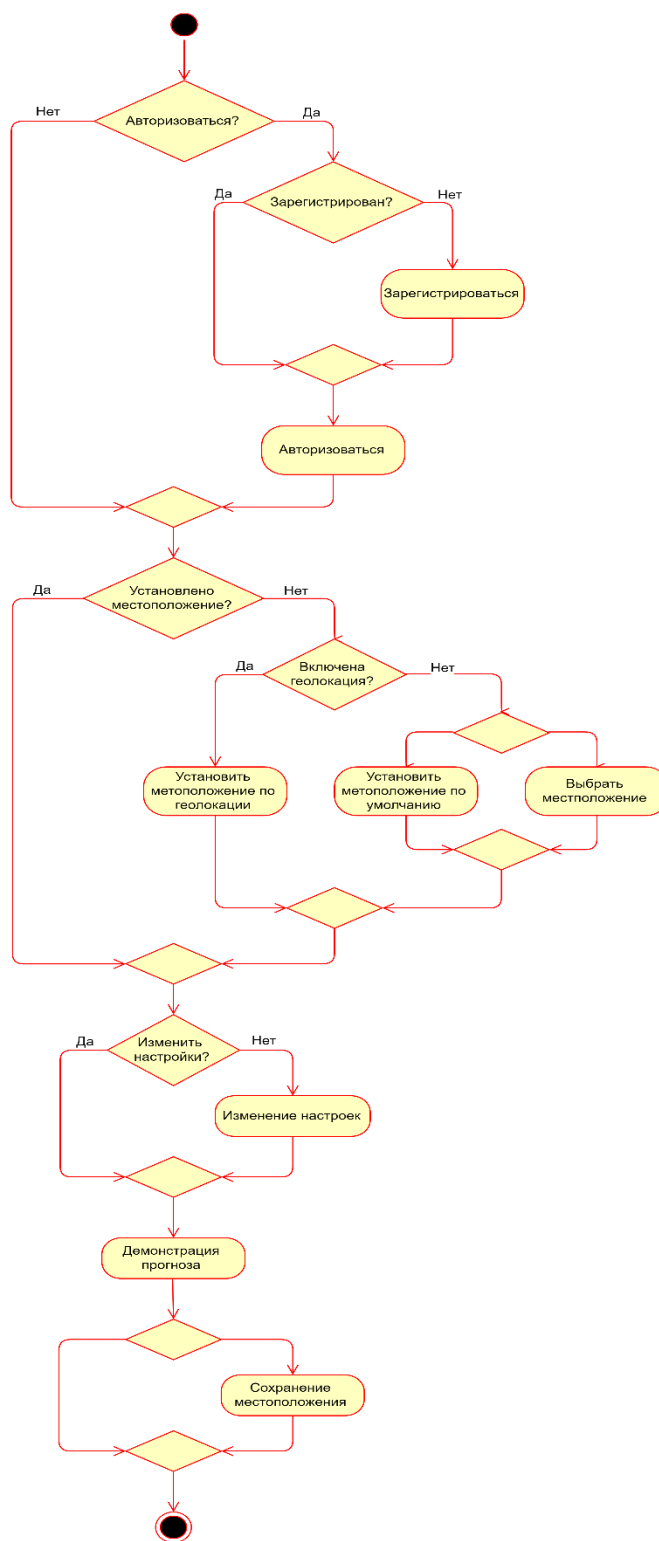


Рисунок 4 - Диаграмма активностей

2.2.3. Состояния Приложения

Состояния, в которых находится Приложение, во время работы приведены на диаграмме состояний на рисунке 5.

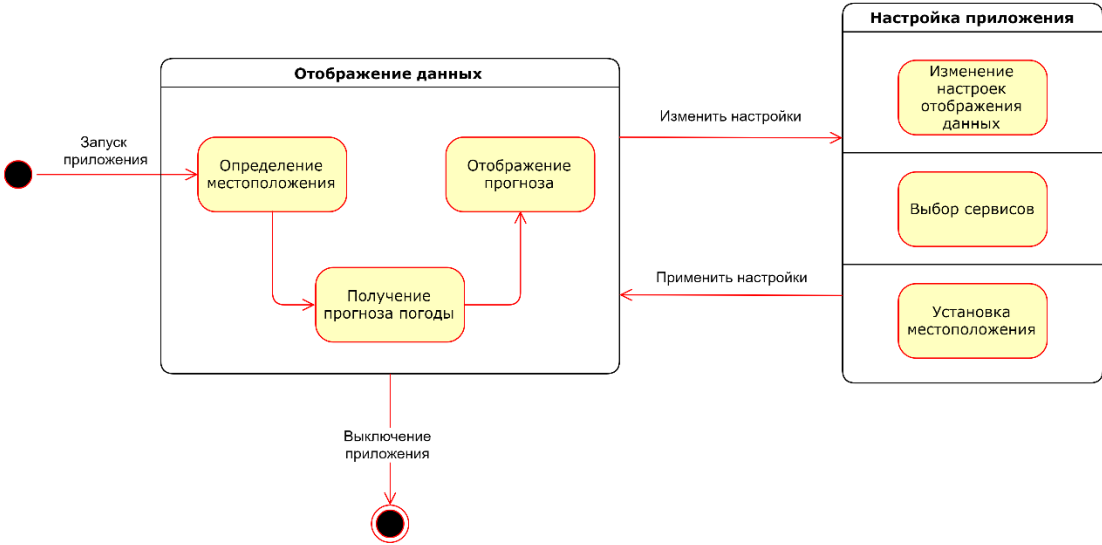


Рисунок 5 - Диаграмма состояний Приложения

2.2.4. Взаимодействие компонентов Приложения

Взаимодействие компонентов Приложения приведено на диаграмме взаимодействия на рисунке 6.



Рисунок 6 - Диаграмма взаимодействия компонентов Приложения

2.2.5. Последовательность исполняемых действий Приложения

Последовательность действий, исполняемых Приложением, во время работы приведена на диаграмме последовательностей на рисунке 7.

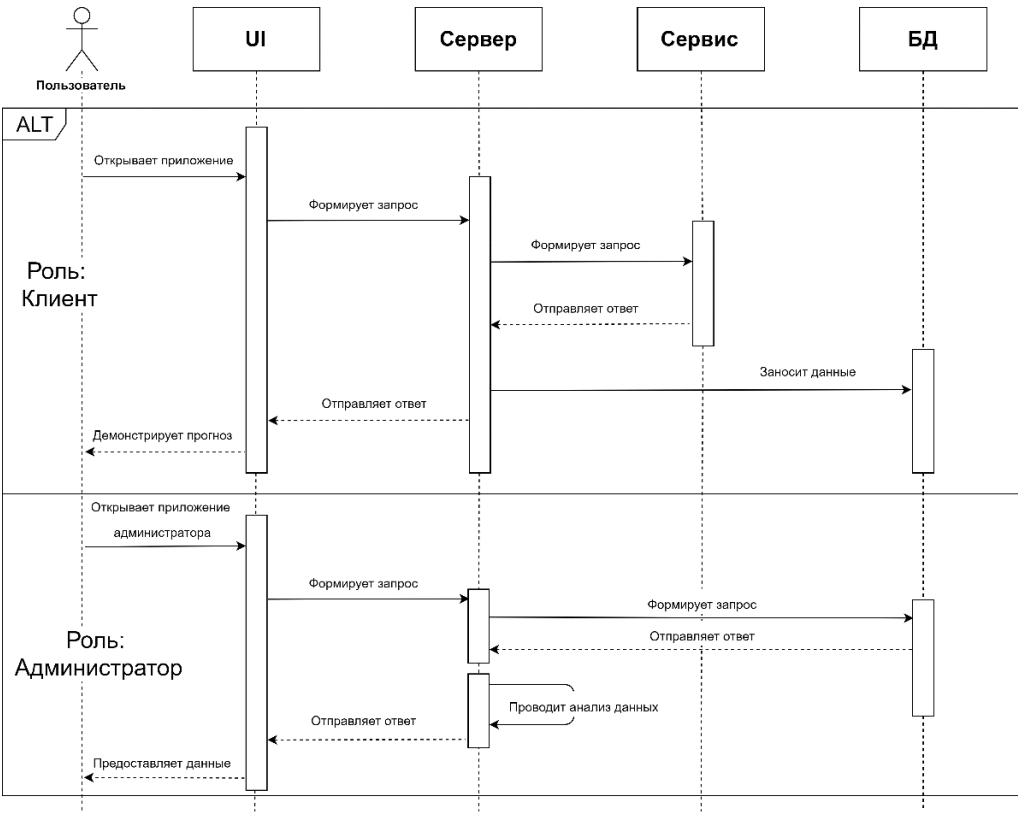


Рисунок 7 - Диаграмма последовательностей Приложения

2.2.6. Структура базы данных Приложения

Структура базы данных Приложения представлена на ER-диаграмме на рисунке 8.

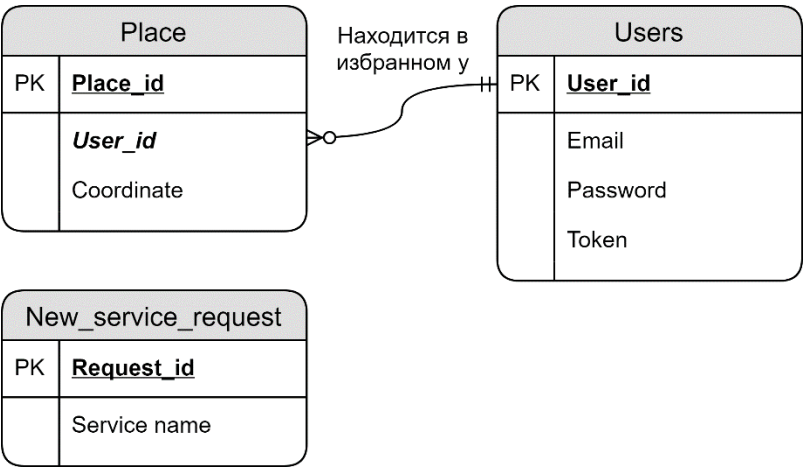


Рисунок 8 - ER-диаграмма базы данных

### 2.2.7. Модель бизнес процесса

Диаграмма IDEF0, иллюстрирующая модель бизнес процесса, реализуемого Приложением, приведена на рисунках 9 и 10.

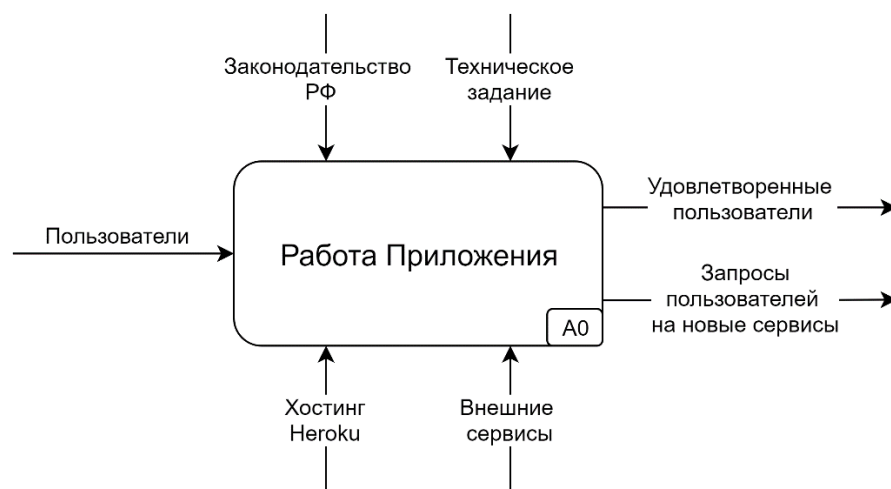


Рисунок 9 - Диаграмма IDEF0 уровень A-0

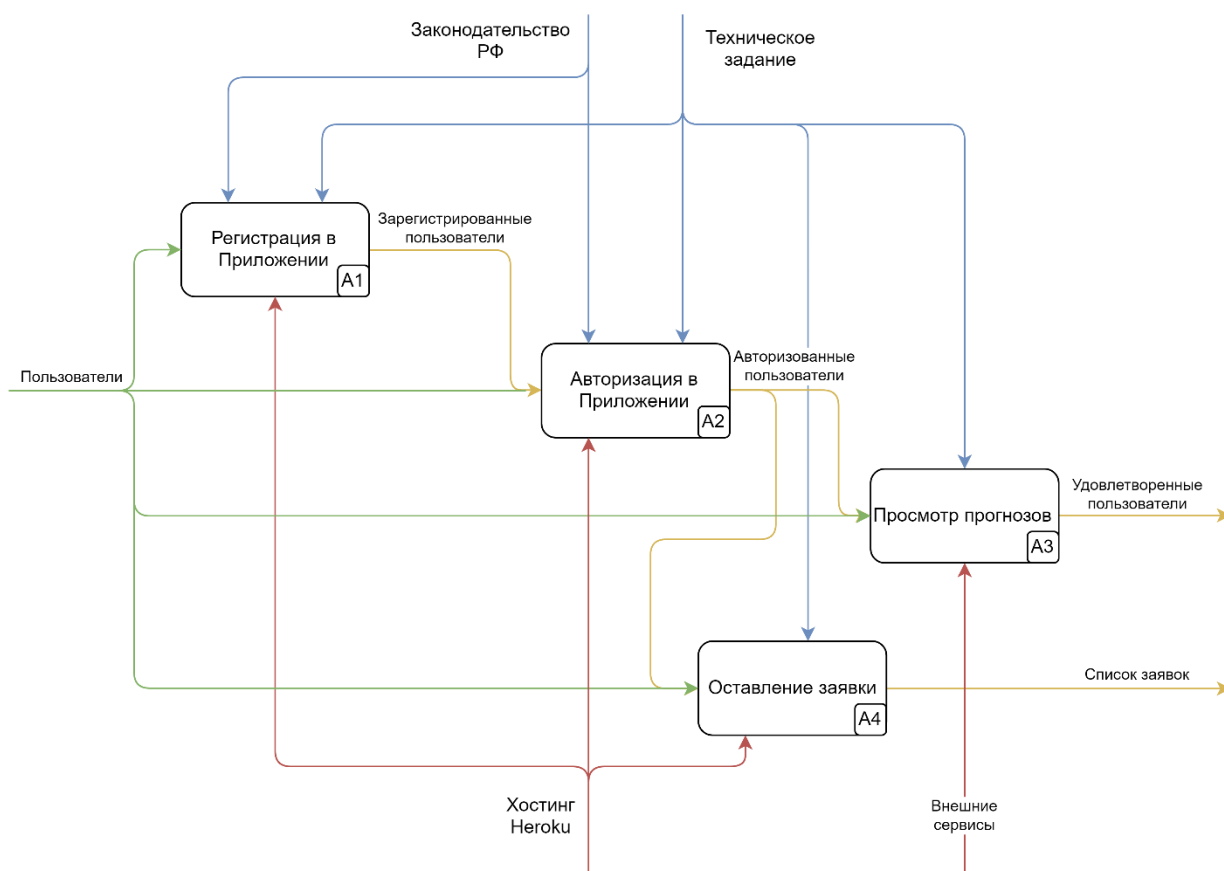


Рисунок 10 - Диаграмма IDEF0 уровень A0

## 3. Реализация

### 3.1. Используемые средства

Для реализации Приложения было использовано множество программных средств. Ниже приведены основные рабочие ресурсы и их краткие описания.

#### 3.1.1. Фреймворк Java Spring Boot

Java Spring Boot – это одно из самых популярных средств для разработки серверных приложений. Своей популярностью данный фреймворк обязан следующим ключевым особенностям:

- управление зависимостями;
- автоматическая конфигурация.

#### 3.1.2. Swagger UI

Swagger – это фреймворк, который позволяет документировать спецификацию RESTful API. В Приложении данный фреймворк используется для автоматической генерации документации методов контроллеров. Результат работы Swagger приведен на рисунке 11.

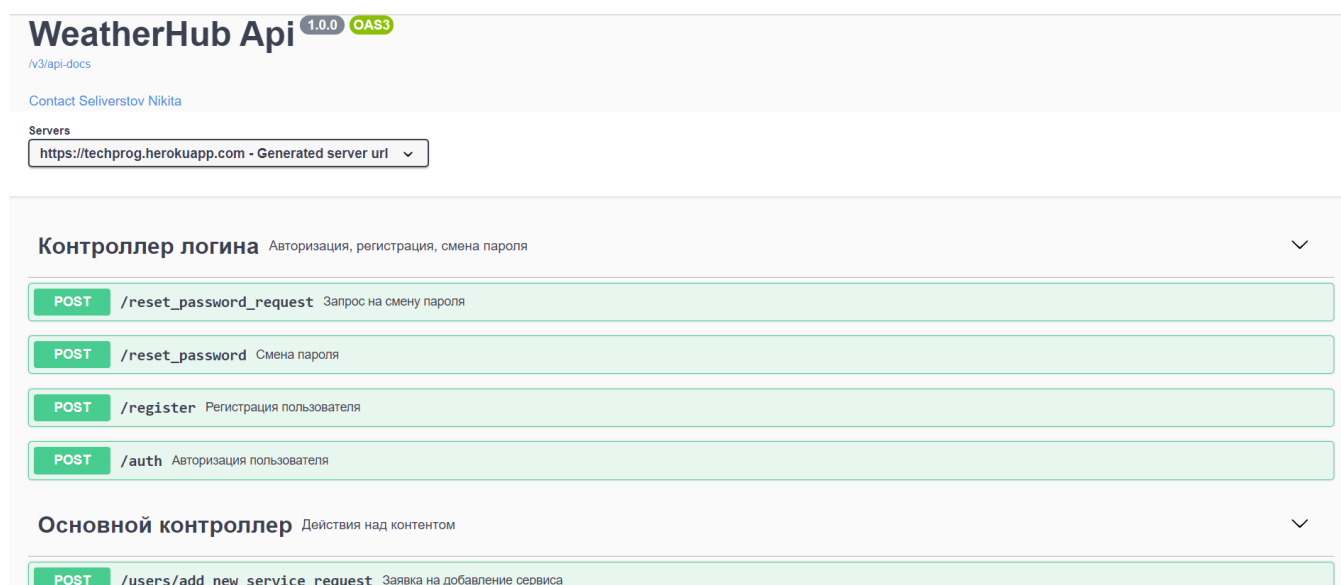


Рисунок 11 - Swagger UI

### **3.1.3. Язык программирования Swift**

Swift – это один из самых популярных языков программирования, который используется для написания приложений на платформу iOS. Разработанный компанией Apple и вышедший в релиз в 2014 году язык Swift позиционируется как легкий для чтения, простой для освоения и устойчивый к ошибкам программиста. В Приложении данный язык использован для написания клиентской части Приложения.

### **3.1.4. СУБД PostgreSQL**

PostgreSQL – это реляционная СУБД. Данная СУБД обладает рядом преимуществ перед конкурентами: полная SQL-совместимость, расширяемость, надежность. В Приложении PostgreSQL используется для хранения данных пользователей: данных авторизации, избранных местоположений и заявок на добавление новых сервисов, предоставляющих прогнозы погоды.

### **3.1.5. Git и GitHub**

Git – это распределенная система управления версиями, которая позволяет фиксировать изменения, внесенные в файлы проекта, в локальном репозитории. В Приложении Git также служит для развертывания сервера на удаленном хостинге.

GitHub – веб-сервис, позволяющий сохранять Git-репозитории в удаленном хранилище.

### **3.1.6. Сервис Firebase Analytics**

Firebase Analytics – это сервис для сбора метрик использования приложения. Он предоставляет SDK для удобной отправки событий в приложении, а также личный кабинет для просмотра статистики и анализа собранных данных.

На рисунке 12 приведены метрики, собираемые в реальном времени при помощи сервиса Firebase Analytics. Эти метрики описаны в Приложении 1 настоящей работы.

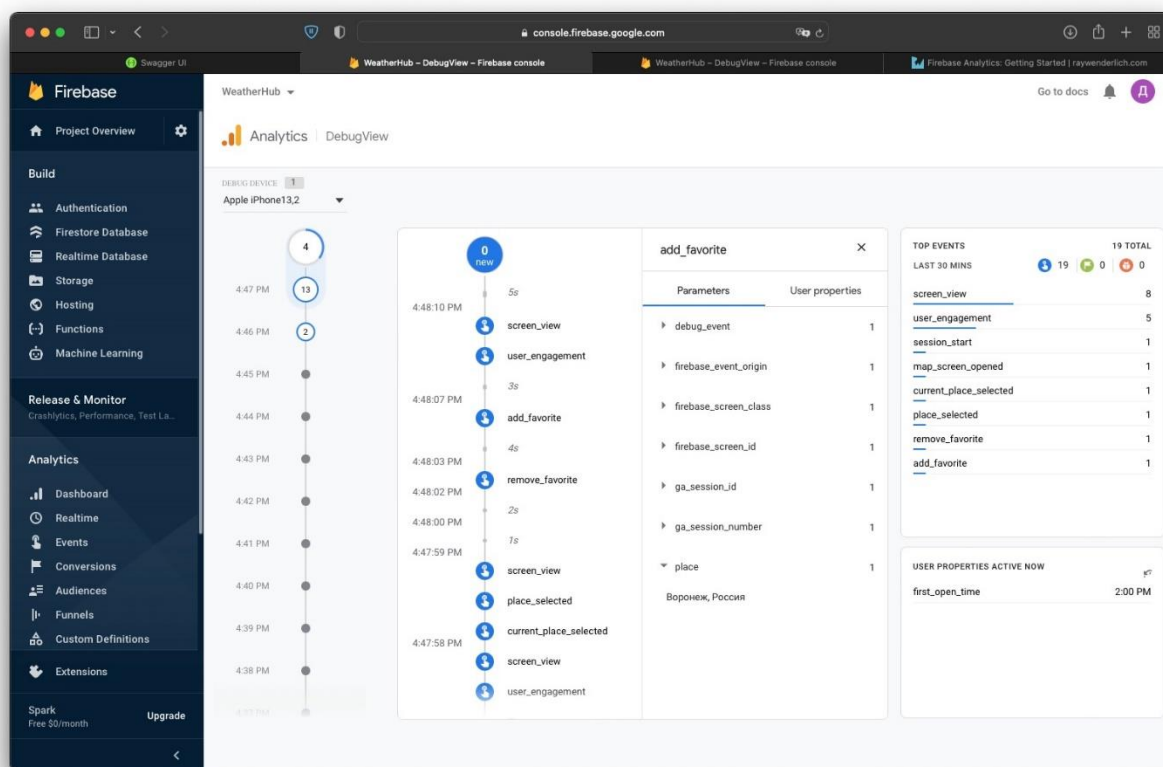


Рисунок 12 - Метрики, собираемые Firebase Analytics

### 3.1.7. Сервис SendGrid

SendGrid – это удобный сервис для электронного маркетинга. Он обладает обширным функционалом: формирование маркетинговой стратегии рассылок, адаптивные рекламные письма, SMS-маркетинг и другие. В Приложении используется SendGrid API, который позволяет отправлять электронные письма через облачную службу SMTP, а также собирать статистику по отправленным и полученным письмам. При помощи данного сервиса реализована отправка электронных писем с кодом подтверждения для смены пароля.

На рисунке 13 приведена статистика SendGrid по электронным письмам, посланных Приложением.



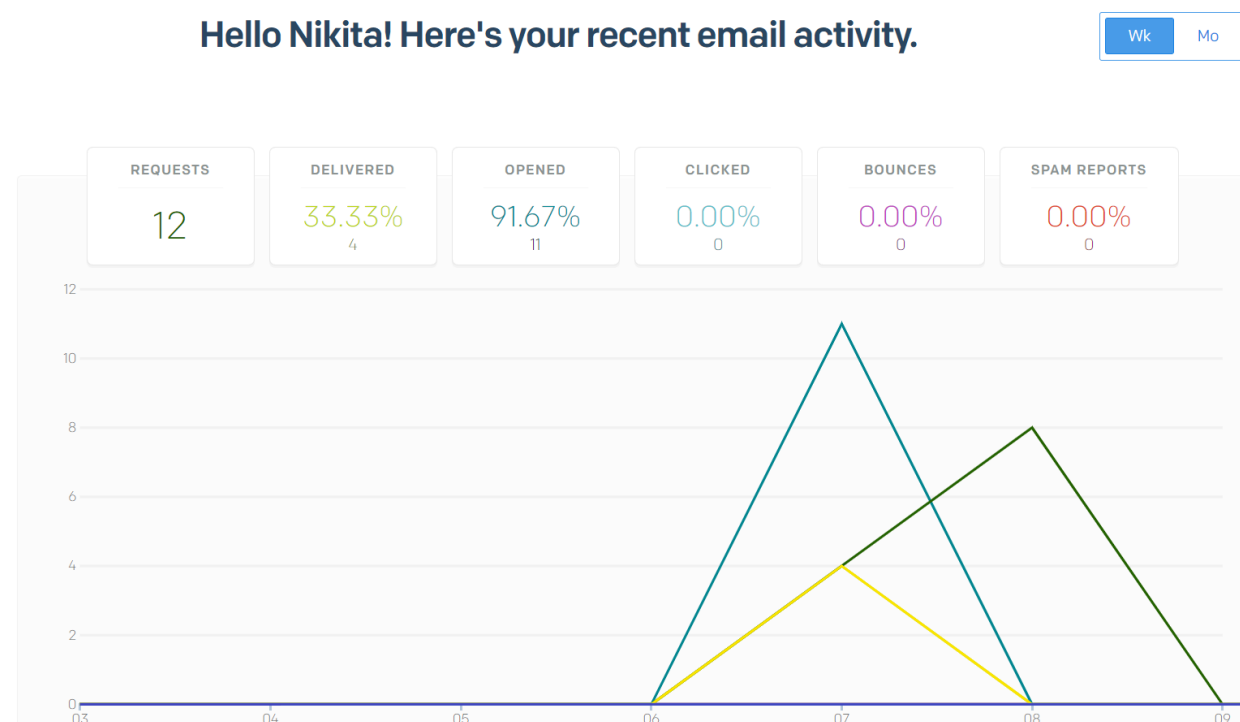


Рисунок 13 - Статистика SendGrid

### 3.2. Развертывание на удаленном сервере

Для развертывания Приложения используется удаленный хостинг Heroku, так как он обладает рядом преимуществ:

- простота развертывания java-приложений при помощи Git;
- большое количество расширений, например, метрики приложения, приведенные на рисунке 14 и описанные в Приложении 1 настоящей работы;
- возможность создания базы данных PostgreSQL на хостинге.



Рисунок 14 - Метрики, собираемые Heroku

## 4. Графический интерфейс

Графический интерфейс имеет два варианта темы: светлый и темный; нужный выбирается автоматически в зависимости от настроек устройства. На рисунке 15 приведены два варианта темы.

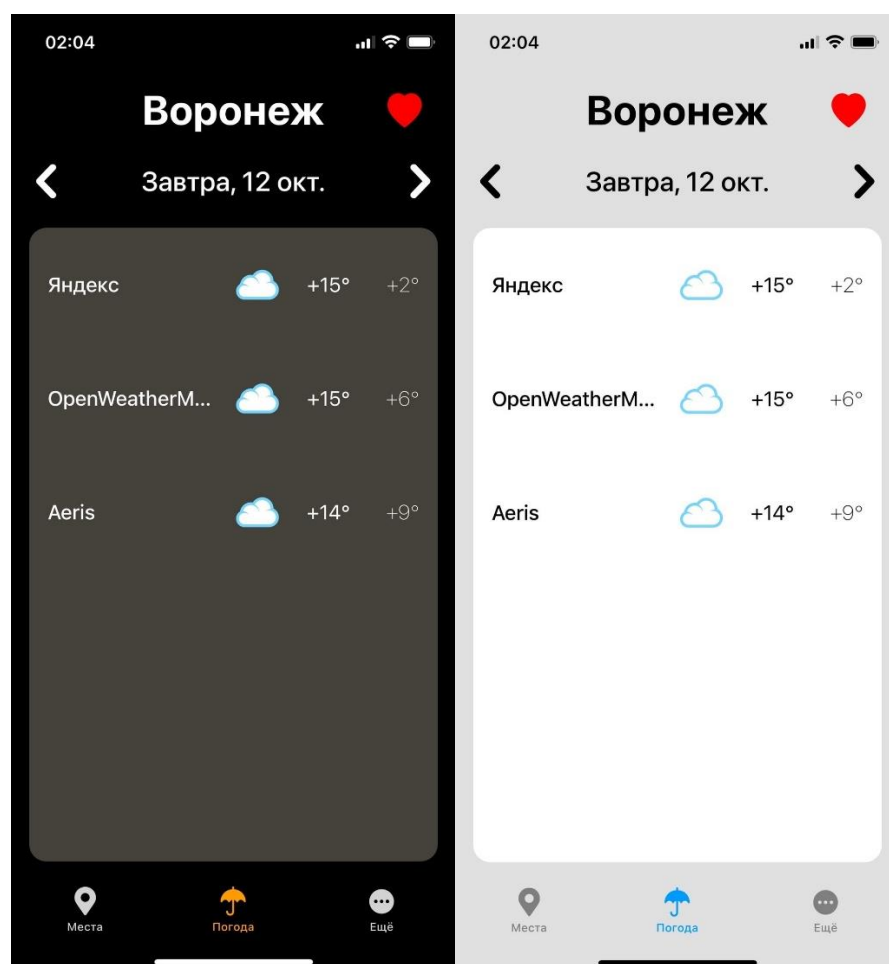


Рисунок 15 - Сравнение тем интерфейса

Графический интерфейс Приложения состоит из нескольких окон. Ниже приведены экраны Приложения с кратким описанием каждого.

### 4.1. Экран просмотра прогноза

Экран просмотра прогноза приведен на рисунке 15. На данном экране отображен выбранный пользователем город, цвет знака сердца показывает добавлен ли город в список избранных местоположений. Ниже расположено меню для выбора даты прогноза. Основную часть экрана занимают прогнозы,

выполненные выбранными сервисами с дневной и ночной температурами, а также иконкой, иллюстрирующей прогнозируемую погоду.

#### 4.2. Экран просмотра списка избранных местоположений

Экран просмотра списка избранных местоположений представлен на рисунке 16. Основную часть экрана занимают названия местоположений, добавленных пользователем в список избранных. По нажатии на местоположение открывается экран просмотра прогноза для него. В нижней части экрана находятся кнопки для поиска местоположения:

- кнопка с иконкой лупы, позволяющая искать местоположения по названию;
- кнопка с иконкой карты, позволяющая искать местоположения на карте;
- кнопка с иконкой указателя, позволяющая определить местоположение при включенной геолокации на устройстве пользователя.

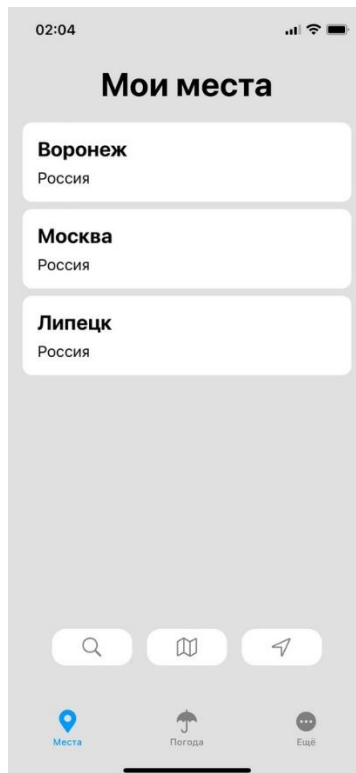


Рисунок 16 - Экран просмотра списка избранных местоположений

#### 4.3. Экран поиска местоположения по названию

Экран поиска местоположения по названию представлен на рисунке 17. Он содержит поле для ввода названия местоположения. При вводе символов открывается список местоположений, начинающихся на данные символы. По нажатию на местоположение открывается экран просмотра прогноза для него.

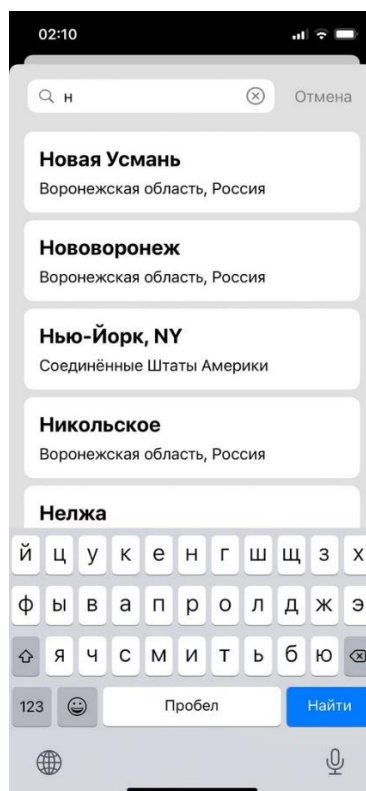


Рисунок 17 - Экран поиска местоположения по названию

#### 4.4. Экран выбора местоположения на карте

Экран выбора местоположения на карте приведен на рисунке 18. Он представляет собой интерактивную карту. По нажатию на точку на карте, а затем – на кнопку «Выбрать» открывается экран просмотра прогноза для данного местоположения.

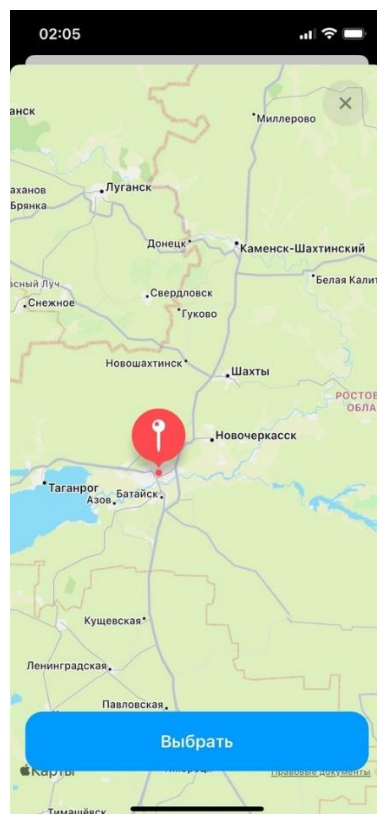


Рисунок 18 - Экран выбора местоположения на карте

#### 4.5. Экран регистрации

Экран регистрации приведен на рисунке 19. Он содержит поля для ввода адреса электронной почты пользователя, его пароля и подтверждения пароля. По нажатии на кнопку «Зарегистрироваться» происходит регистрация нового пользователя при условиях того, что пароли в обоих полях совпадают, а также не существует пользователя, зарегистрированного с введенным адресом электронной почты.

По нажатии на ссылку с текстом «Войти» открывается экран авторизации.

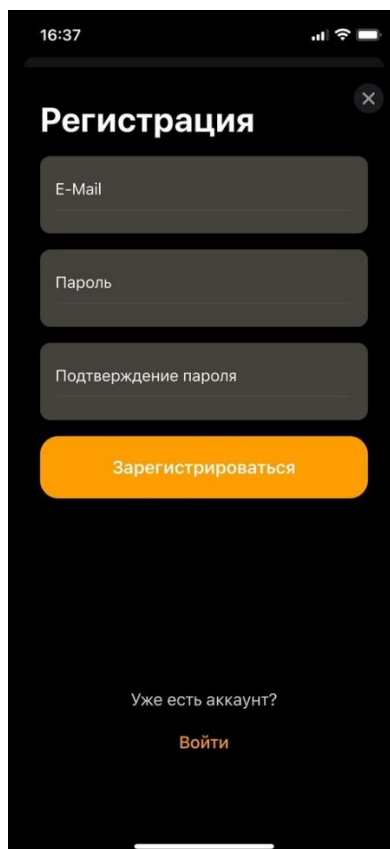


Рисунок 19 - Экран регистрации

#### 4.6. Экран авторизации

Экран авторизации содержит поля для ввода адреса электронной почты пользователя и его пароля. По нажатии на кнопку «Войти» происходит авторизация пользователя при условии того, что пароль совпадает с установленным при регистрации или после его замены. После авторизации открывается экран, показывающий данные аккаунта и позволяющий выйти из него. Экраны авторизации и отображения аккаунта приведены на рисунке 20.

По нажатии на ссылку с текстом «Я не помню пароль» открываются экраны смены пароля.

По нажатии на ссылку с текстом «Зарегистрироваться» открывается экран регистрации.

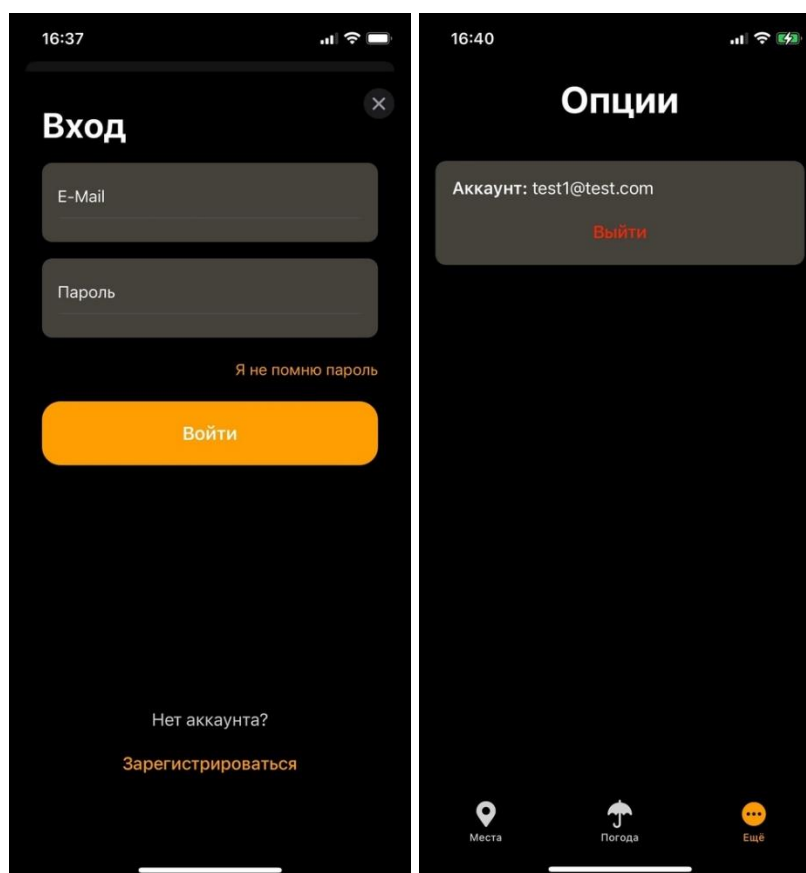


Рисунок 20 - Экраны авторизации и отображения аккаунта

#### 4.7. Экраны смены пароля

Экраны смены пароля представлены на рисунке 21. Первый экран содержит поле для ввода адреса электронной почты. По нажатии на кнопку «Отправить код» на этот адрес отправляется цифровой код, если пользователь с введенным адресом зарегистрирован в Приложении. Далее происходит перенаправление на второй экран, который содержит поля для ввода кода подтверждения, нового пароля и подтверждения пароля. По нажатии на кнопку «Установить» происходит замена пароля, при условии совпадения введенного кода с отправленным на адрес электронной почты, а также авторизация в Приложении.



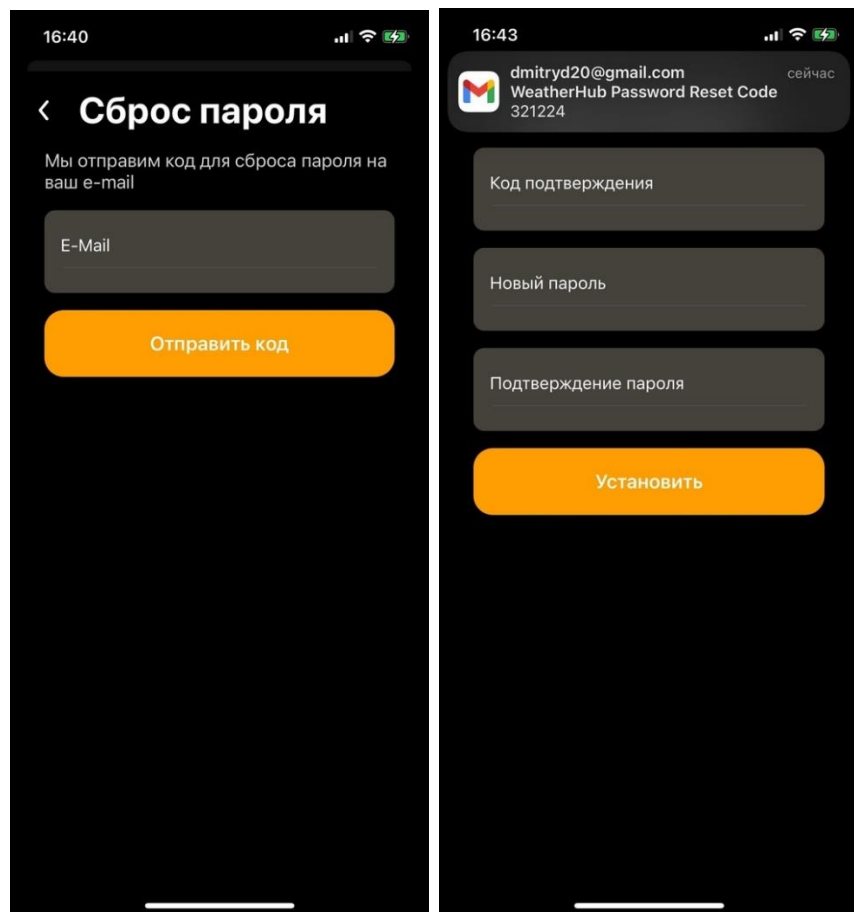


Рисунок 21 - Экраны смены пароля

## 5. Перспективы развития Приложения

Существует несколько вариантов развития функциональности и доступности Приложения:

- увеличение числа поддерживаемых погодных сервисов;
- возможность просмотра более подробных прогнозов;
- поддержка различных языков, например, упрощенного китайского;
- разработка клиентского приложения на платформу Android;
- разработка web-клиента.

## Заключение

Результатом выполнения курсового проекта является клиент-серверное приложение, которое выполняет поставленные задачи согласно установленным требованиям.

В ходе выполнения курсового проекта были выполнены следующие задачи:

- разработано клиентское приложение;
- разработано серверное приложение, развернутое на удаленном хостинге;
- составлена документация REST API;
- составлены диаграммы, описывающие Приложение;
- написан настоящий документ.

## Список использованных источников

1. Swagger UI [Электронный ресурс] .— режим доступа <https://habr.com/ru/post/434798/> — Дата обращения 10.10.2021.
2. SendGrid [Электронный ресурс] .— режим доступа <https://ecommerce-platforms.com/ru/email-marketing-services-reviews/sendgrid-review-e-mail-marketing-service/> — Дата обращения 10.10.2021.
3. Heroku [Электронный ресурс] .— режим доступа <https://ru.education-wiki.com/3703940-heroku-vs-aws/> — Дата обращения 10.10.2021.
4. PostgreSQL [Электронный ресурс] .— режим доступа <https://tproger.ru/translations/sqlite-mysql-postgresql-comparison/> — Дата обращения 10.10.2021.
5. Spring Boot [Электронный ресурс] .— режим доступа <https://habr.com/ru/post/435144/> — Дата обращения 10.10.2021.

В ходе работы Приложения автоматически собираются различные аналитические данные, которые позволяют получить представление о состоянии Приложения в процессе работы и поведении пользователей во время его использования.

Метрики серверного приложения, собираемые хостингом Heroku:

- события приложения: ошибки, активность, предупреждения;
- использование оперативной памяти;
- время ответа на запросы;
- пропускная способность;
- использование ресурсов хостинга.

Для анализа поведения пользователя в клиентском приложении созданы цепочки из последовательных событий, отправляющихся в Firebase Analytics:

- цепочка выбора местоположения состоит из следующих событий:
  - открытие экрана просмотра списка избранных местоположений;
  - опциональные события открытия экранов поиска местоположения по карте или по названию;
  - выбор местоположения;
- цепочка синхронизации списка избранных местоположений состоит из следующих событий:
  - переход на экран авторизации;
  - регистрация и авторизация;

- переход на экран просмотра списка избранных местоположений;

- события добавления местоположения в список избранных и удаления оттуда не образуют цепочку, но также учитываются в сборе аналитических данных.

Сервис SendGrid собирает следующие аналитические данные об отправке электронной почты Приложением:

- число запросов на отправку письма;
- процент успешно доставленных писем;
- процент открытых писем;
- процент писем, помеченных, как спам.