

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

Курсовой проект по дисциплине

Технологии программирования

Мобильное приложение «FastWeather»

Студент

_____ Э.Э. Яйлаев, 4 курс
бакалавриат, д/о

«___»_____ 2021 г.

Преподаватель

_____ В.С. Тарасов, старший
преподаватель

«___»_____ 2021 г.

Воронеж 2021

Содержание

Содержание	2
Введение	4
1. Постановка задачи	5
1.1. Цель разработки Приложения	5
2. Анализ предметной области	7
2.1. Обзор аналогичных решений	7
2.1.1. Gismeteo Lite	7
2.1.2. Погода для iPad	9
3. Анализ задачи	10
3.1. Варианты использования приложения	10
3.2. Действия с Приложением	11
3.3. Состояния Приложения	12
3.4. Взаимодействие компонентов Приложения	13
3.5. Последовательность исполняемых действий Приложения	14
3.6. Структура базы данных Приложения	15
3.7. Модель бизнес-процесса	16
4. Анализ средств реализации	17
4.1. Используемые средства	17
4.1.1. Язык программирования Swift	17
4.1.2. СУБД MySQL	17
4.1.3. Git и GitHub	17
4.1.4. Сервис Firebase Analytics	18
5. Графический интерфейс	19

5.1. Экран регистрации.....	19
5.2. Экран входа	20
5.3. Экран поиска местоположения по названию.....	21
5.4. Экран просмотра списка избранных местоположений.....	22
6. Тестирование	23
6.1. UI тесты.....	23
6.2. Юзабилити тестирование	24
7. Перспективы развития Приложения	25
Заключение	26

Введение

В настоящее время человеку необходимо повседневно узнавать погоду, так как от нее зависит планирование многих повседневных задач. Для решения этой проблемы в современном мире разработано большое количество различных сервисов, реализующих алгоритмы предсказания погоды и формирующих ее прогнозы.

Есть много сервисов, определяющих погоду для выбранной местности или по местоположению, но для людей, которые часто путешествуют есть актуальная проблема, которую на данный момент не решают многие сервисы. В связи с этим люди сталкиваются с неудобством: необходимость каждый раз менять местоположение для просмотра погоды разных локаций.

Для решения перечисленных выше проблем, было принято решение разработать мобильное приложение, выполняющее объединение погоды разных мест и предоставление результатов в одном приложении.

1. Постановка задачи

1.1. Цель 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 разных локаций одновременно
- Простая работа с локациями- добавление и удаление из избранных

Функциональность приложения

Для достижения поставленных целей Приложение должно предоставлять пользователям следующие возможности:

- Доступ к погоде избранных местоположений с главной страницы приложения
- Регистрация пользователей по адресу электронной почты для доступа к списку избранных местоположений.
- Добавление и удаление местоположений
- Разработка Front-end части сервиса;
- Разработка Back-end части сервиса;
- Создание связи между Front-end и Back-end частями приложения;
- Разработка базы данных.

2. Анализ предметной области

2.1. Обзор аналогичных решений

2.1.1. Gismeteo Lite

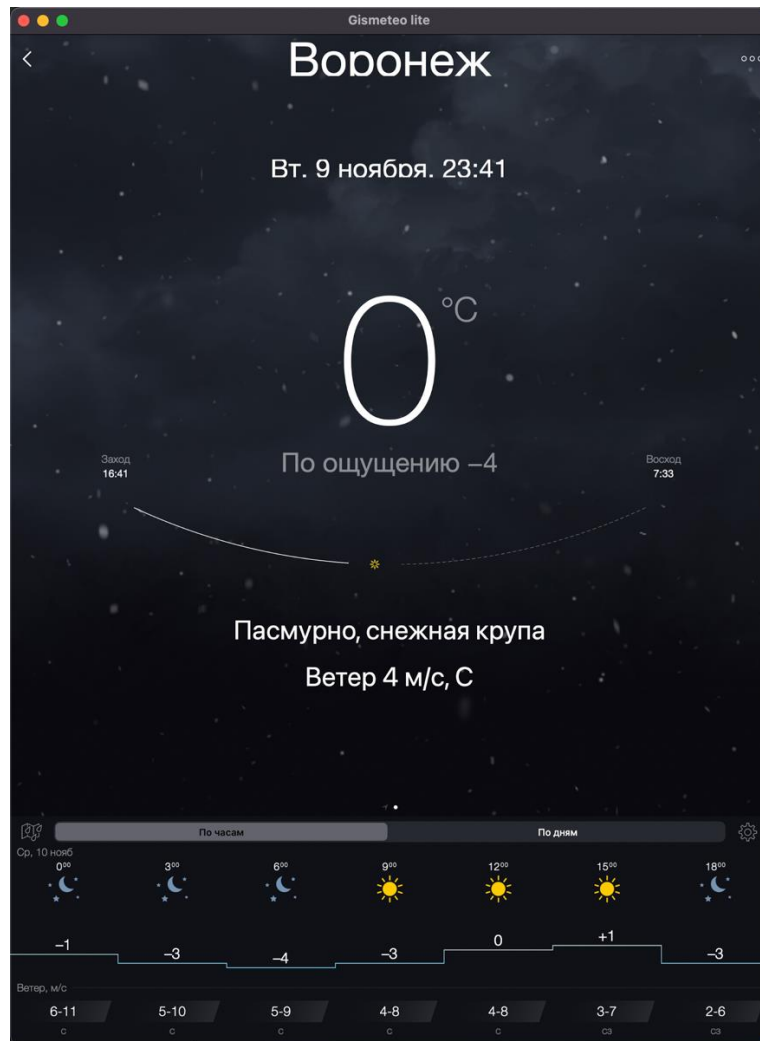


Рисунок 1 - Пользовательский интерфейс приложения Gismeteo Lite

Приложение «Gismeteo lite» обладает следующими особенностями:

- При открытии приложения пользователь вместо прогноза погоды видит список сохраненных местоположений, для просмотра прогноза погоды необходимо выбрать населённый 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.

2.1.2. Погода для iPad

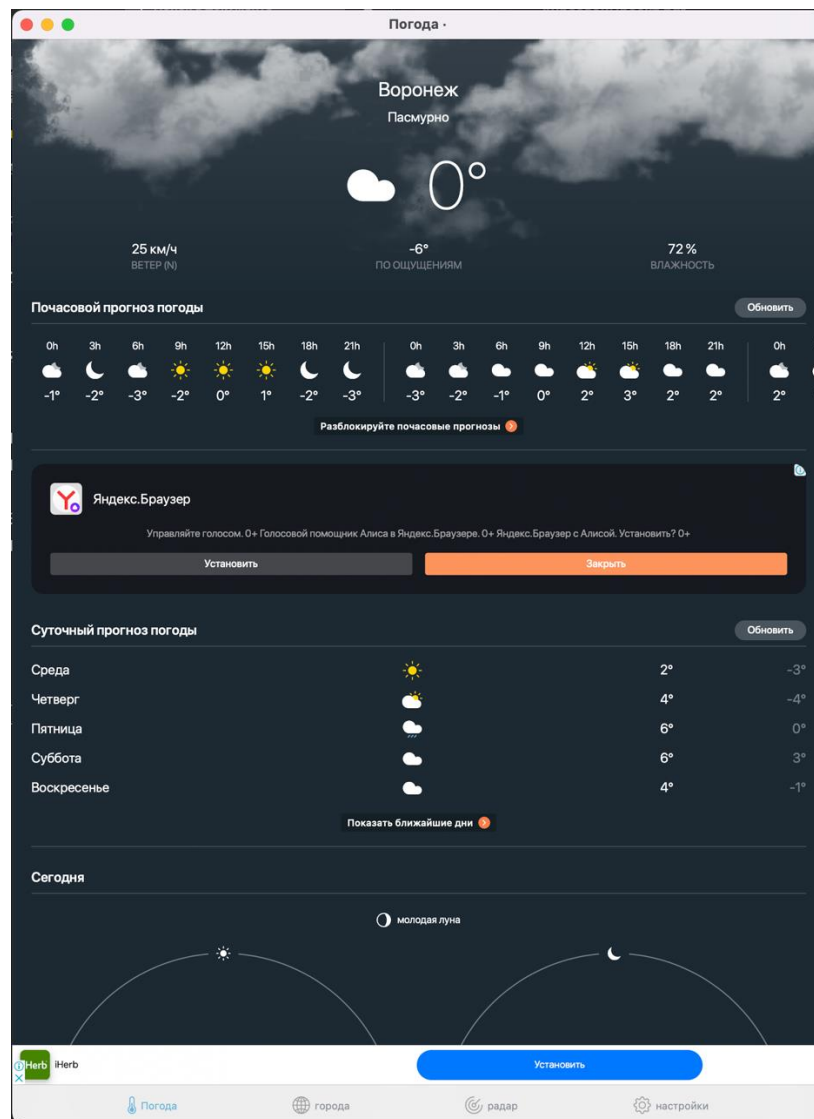


Рисунок 2 - пользовательский интерфейс приложения Погода для iPad

Приложение «Яндекс Погода» обладает следующими особенностями:

- 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.

3. Анализ задачи

3.1. 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

3.2. Действия с Приложением

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

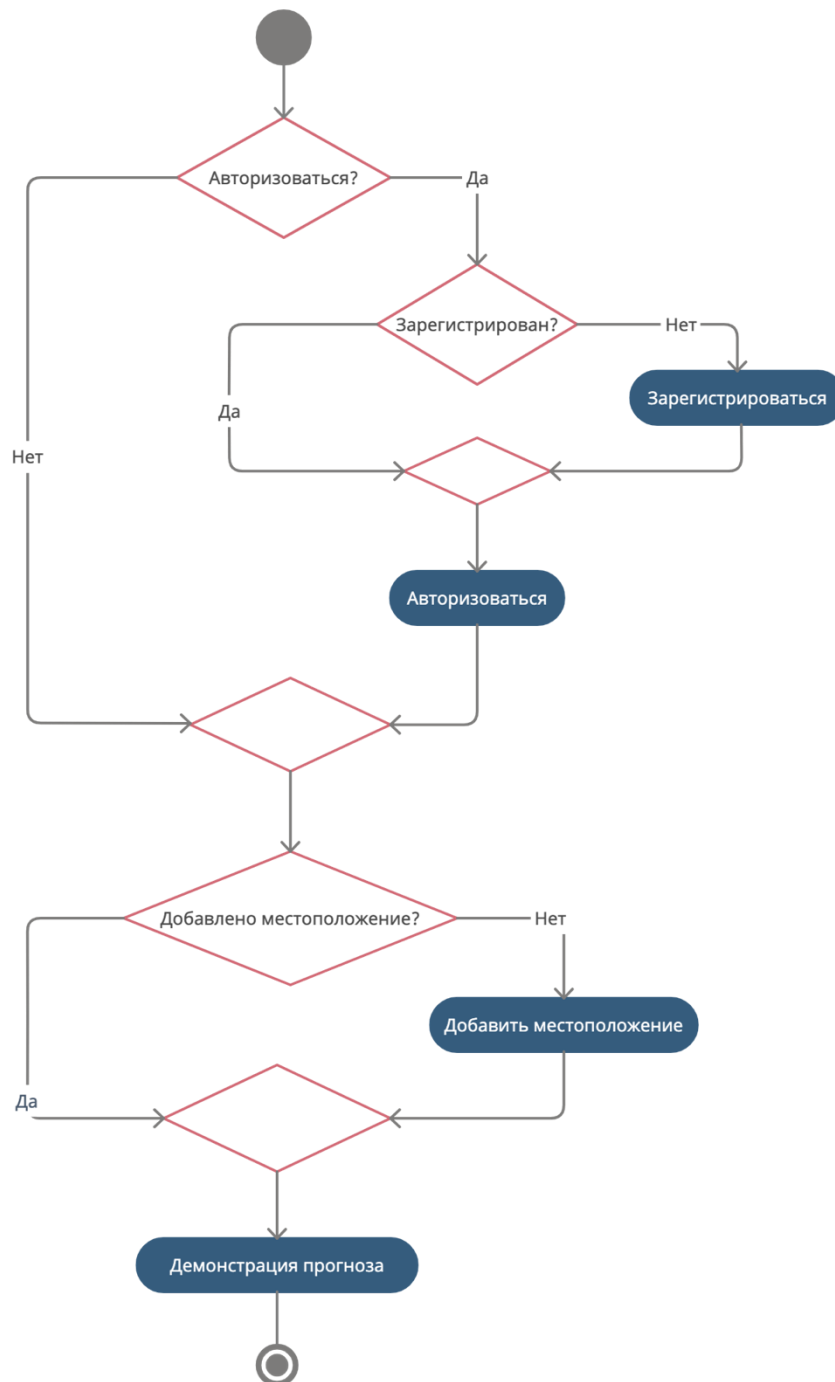


Рисунок 4 - Диаграмма активностей

3.3. Состояния Приложения

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

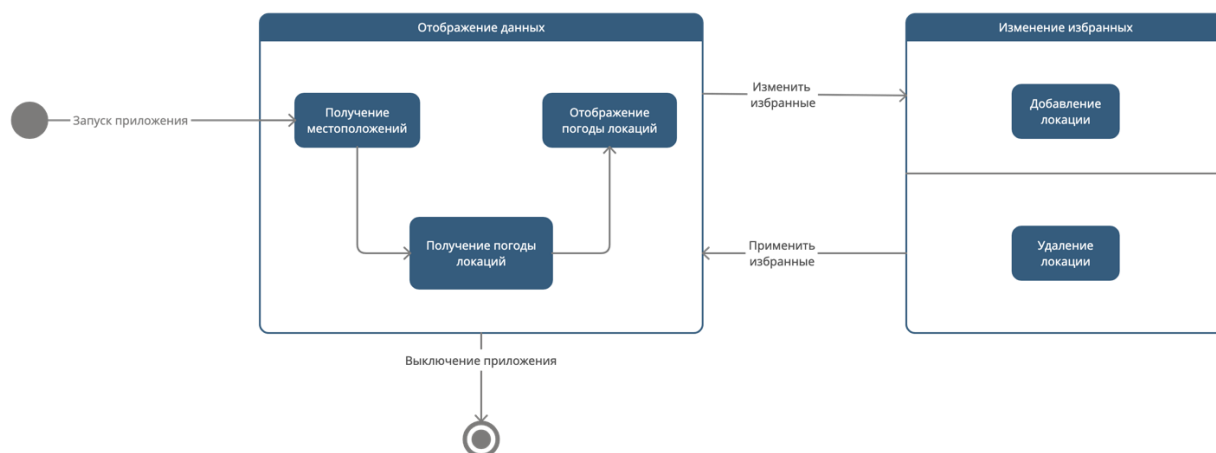


Рисунок 5 - Диаграмма состояний Приложения

3.4. Взаимодействие компонентов Приложения

Взаимодействие компонентов Приложения приведено на диаграмме взаимодействия на рисунке 6.



Рисунок 6 - Диаграмма взаимодействия компонентов Приложения

3.5. Последовательность исполняемых действий Приложения

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

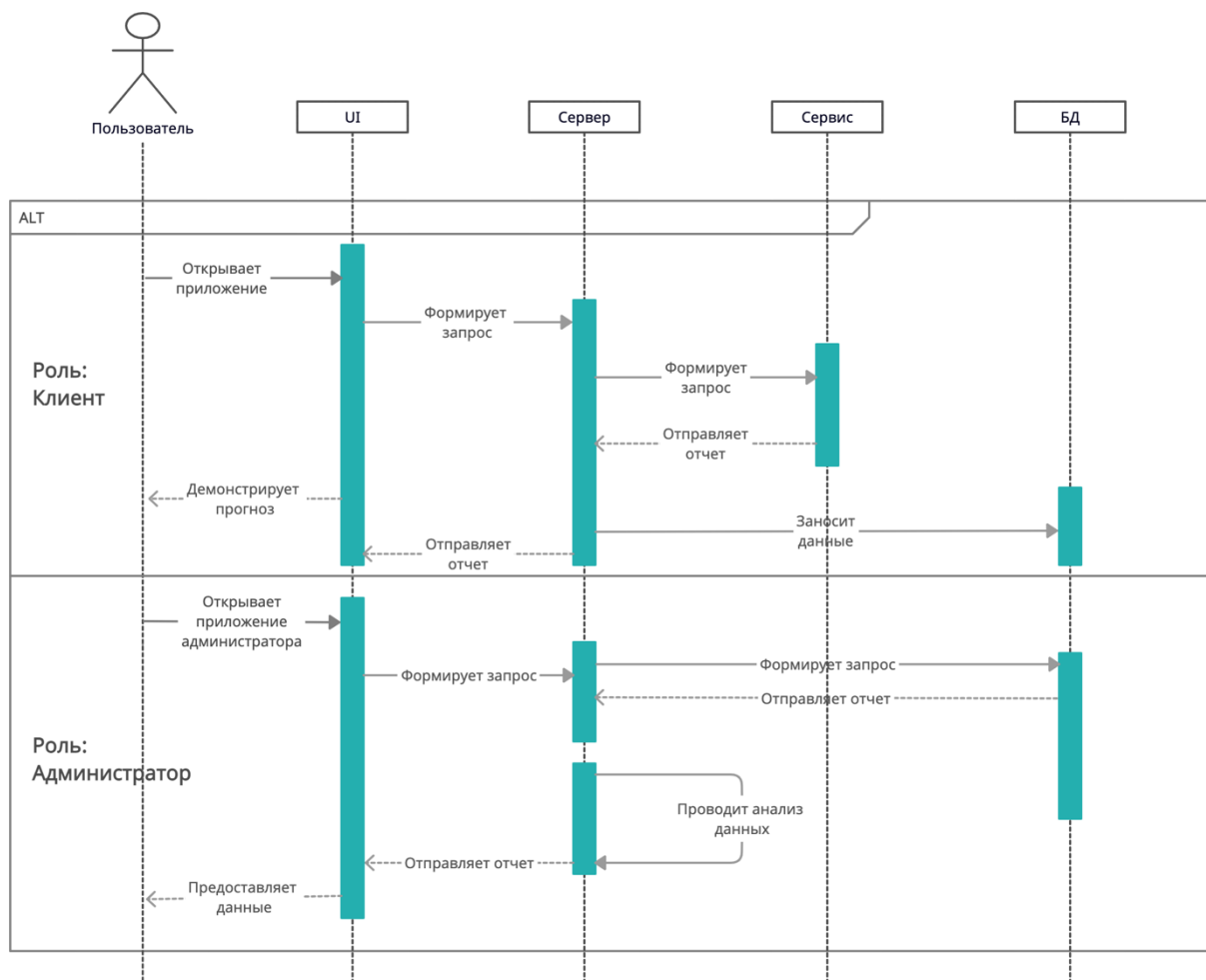


Рисунок 7 - Диаграмма последовательностей Приложения

3.6. Структура базы данных Приложения

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

Users	
id (PK)	int
login	varchar
password	varchar
name	varchar
surname	varchar
time	int

3.7. Модель бизнес-процесса

Диаграмма IDEF0, иллюстрирующая модель бизнес-процесса, реализуемого Приложением, приведена на рисунке 9

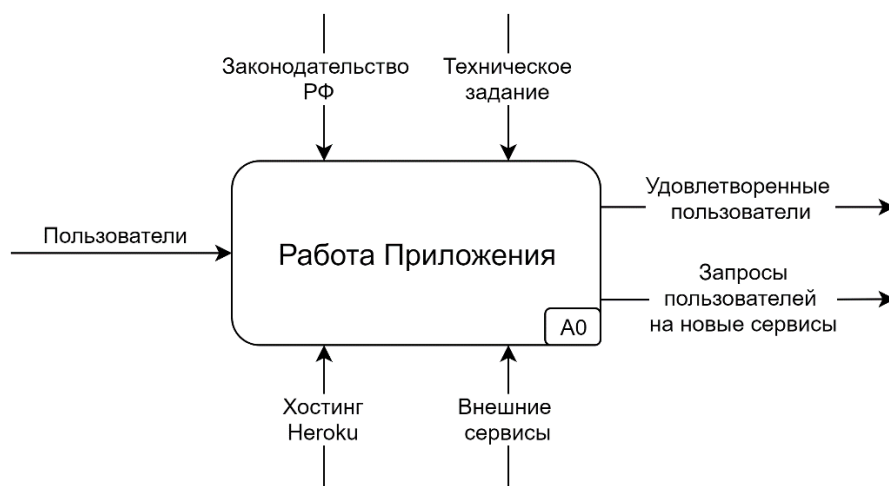


Рисунок 9 - Диаграмма IDEF0 уровень A-0

4. Анализ средств реализации

4.1. Используемые средства

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

4.1.1. 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.

4.1.2. СУБД MySQL

MySQL- свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

4.1.3. 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.

4.1.4. 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 10 приведены метрики, собираемые в реальном времени при помощи сервиса Firebase Analytics. Эти метрики описаны в Приложении 1 настоящей работы.

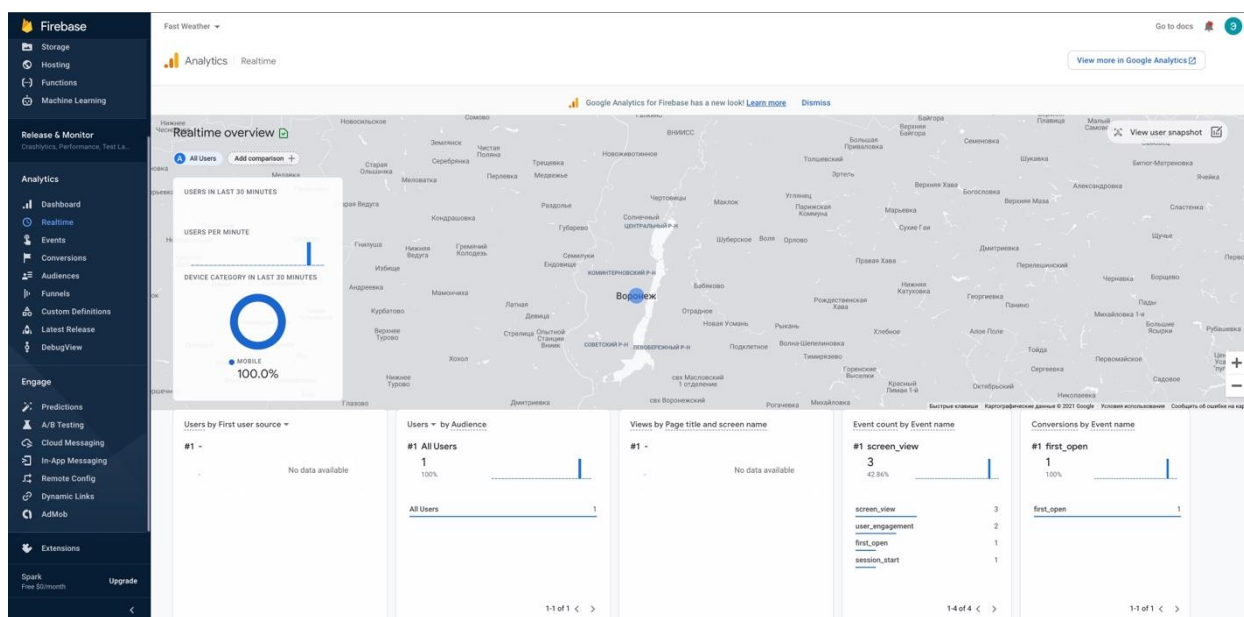


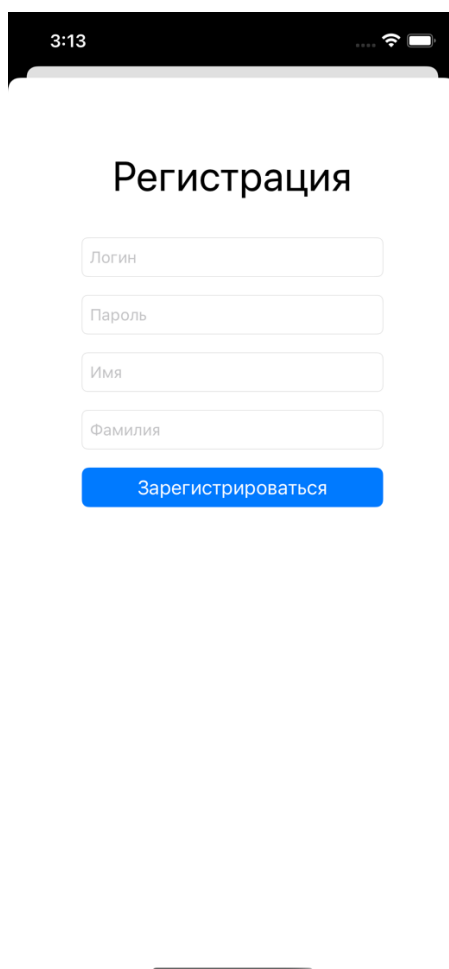
Рисунок 10 – Метрики, собираемые Firebase Analytic

5. Графический интерфейс

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

5.1. Экран регистрации

Экран регистрации приведен на рисунке 11. Он содержит поля для ввода логин, пароль, имя и фамилию пользователя. По нажатии на кнопку «Зарегистрироваться» происходит регистрация нового пользователя при условии того, что не существует пользователя, зарегистрированного с введенным логином.



The image shows a mobile application registration screen. At the top, there is a status bar with the time 3:13 and icons for signal, Wi-Fi, and battery. Below the status bar, the title "Регистрация" is centered. Under the title, there are four input fields stacked vertically, labeled "Логин", "Пароль", "Имя", and "Фамилия". Below these fields is a blue button with the text "Зарегистрироваться". At the bottom of the screen, there is a horizontal line representing the home indicator bar.

Рисунок 11 – Окно регистрации

5.2. Экран входа

Экран авторизации содержит поля для ввода логин пользователя и его пароля. По нажатии на кнопку «Войти» происходит авторизация пользователя. После авторизации открывается экран, показывающий данные аккаунта и позволяющий выйти из него. Экраны авторизации и отображения аккаунта приведены на рисунке 12.

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

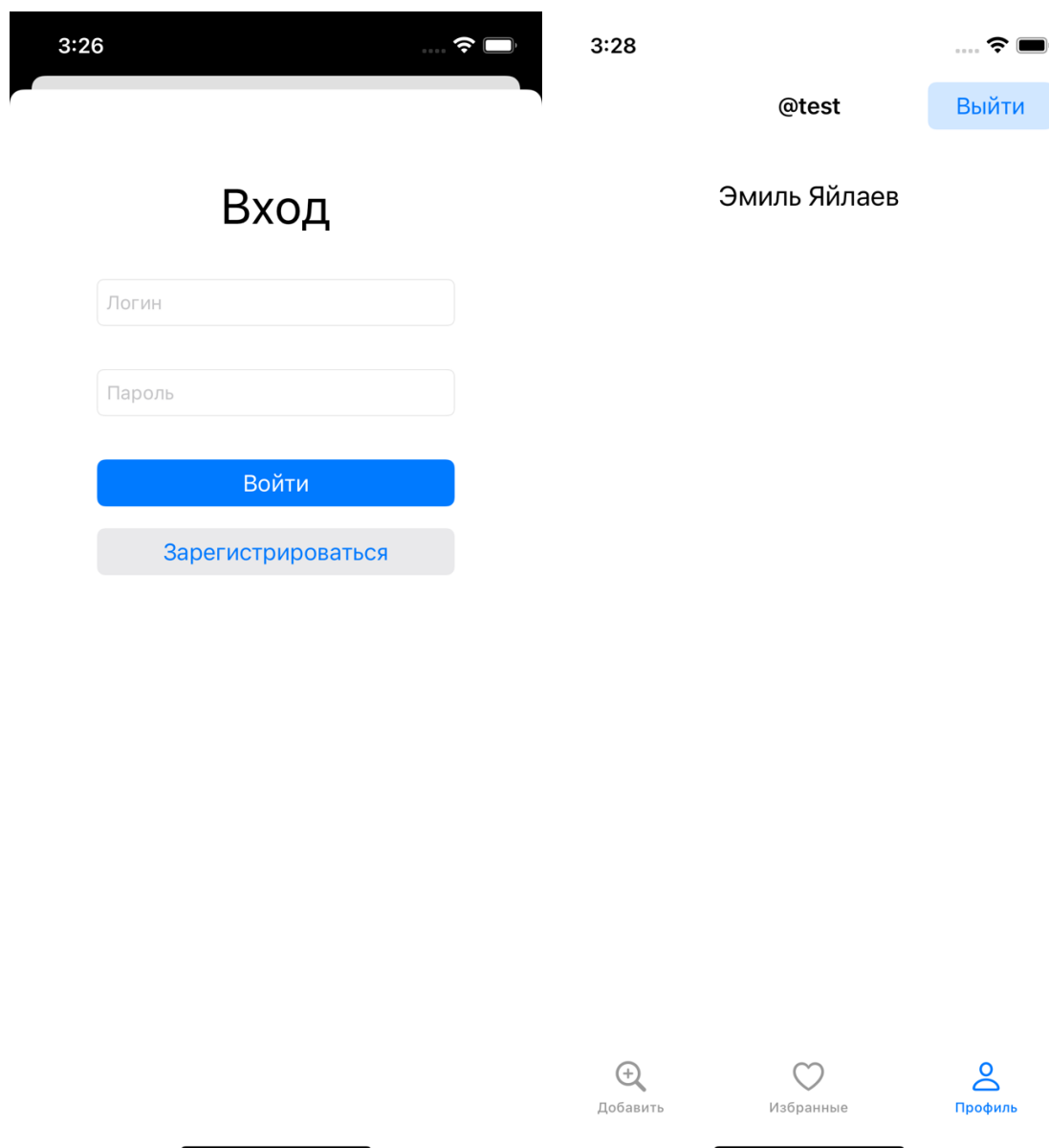


Рисунок 12 - Экраны входа и отображения аккаунта

5.3. Экран поиска местоположения по названию

Экран поиска местоположения по названию представлен на рисунке 13. Он содержит поле для ввода названия местоположения. При вводе символов открывается список местоположений, начинающихся на данные символы. По нажатию на местоположение отображается кнопка для добавления или удаления из Избранных.

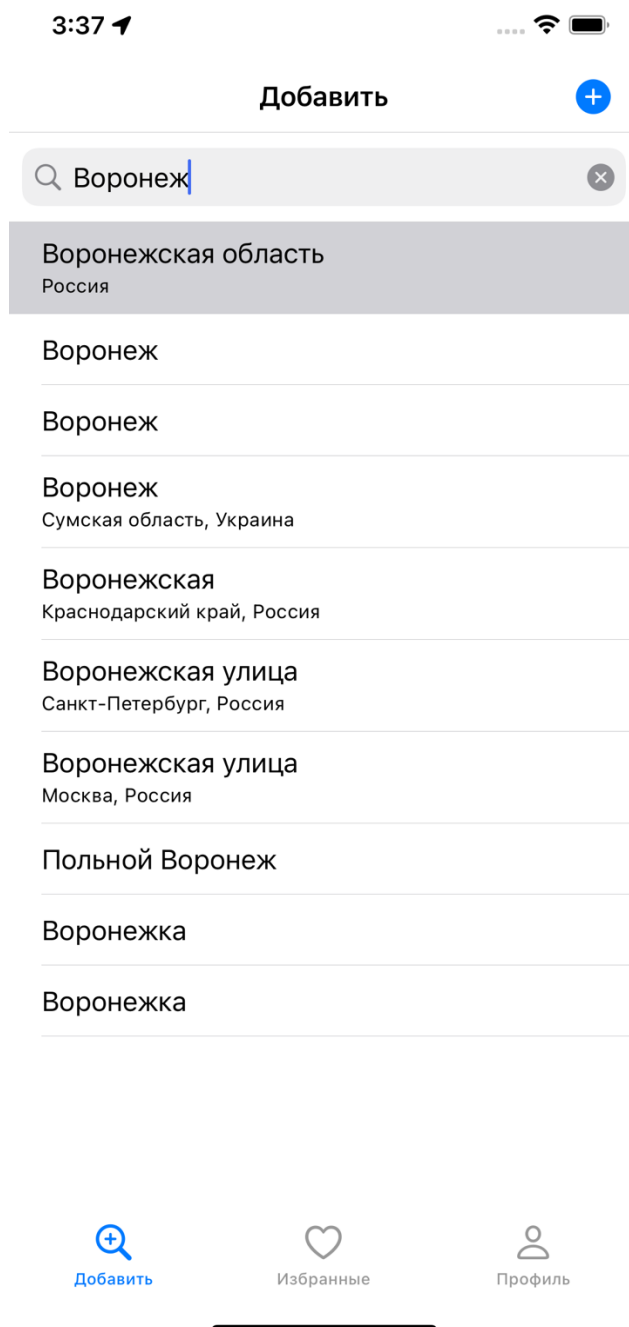


Рисунок 13 - Экран поиска местоположения по названию

5.4. Экран просмотра списка избранных местоположений

Экран просмотра списка избранных местоположений представлен на рисунке 14. Основную часть экрана занимают местоположения с описанием погоды, добавленных пользователем в список избранных.

Основные функции:

- Кнопка с иконкой карандаша, в правом верхнем углу, позволяет открыть список действий над списком.
- Swiре местоположения влево, дает возможность быстрого удаления местоположения из избранных.

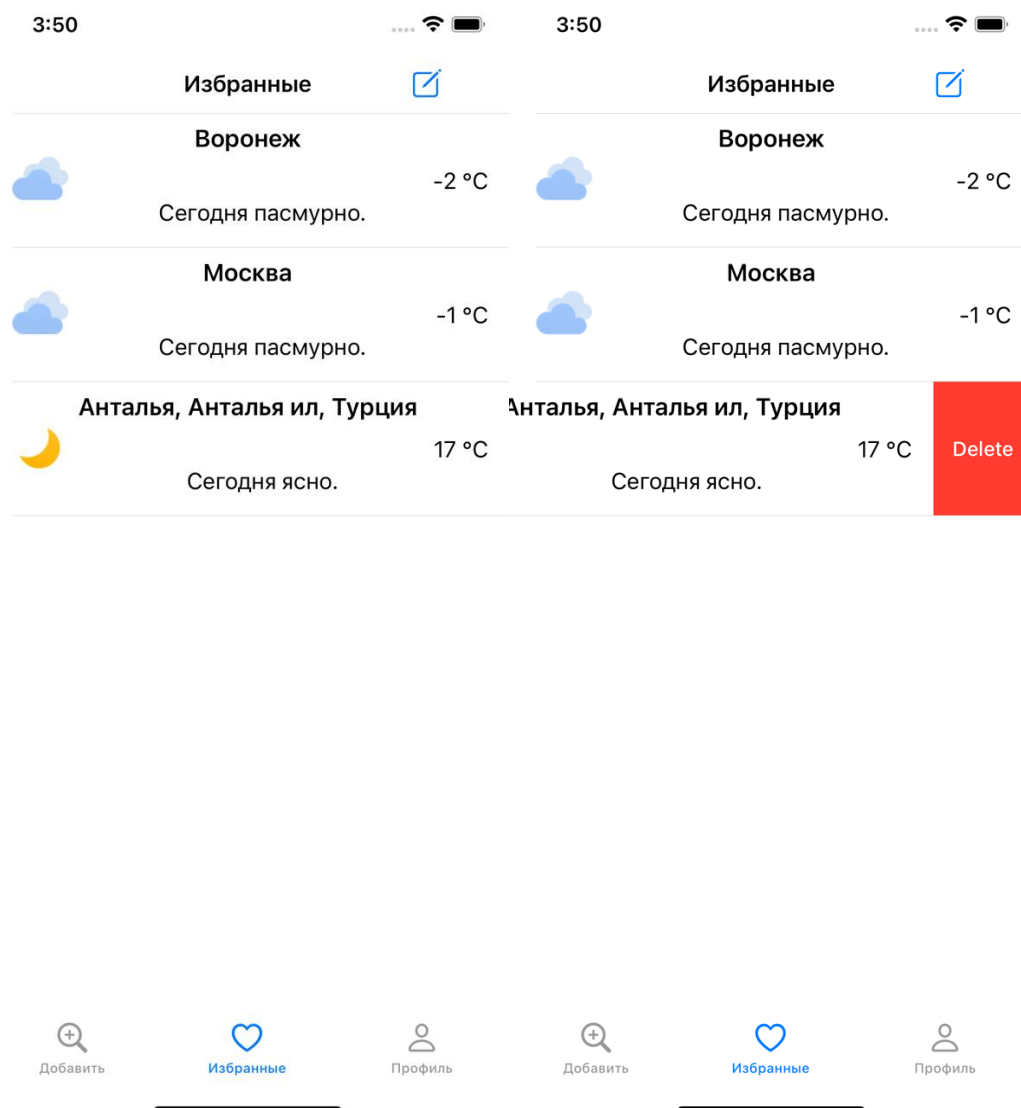


Рисунок 14 - Экран просмотра списка избранных местоположений

6. Тестирование

Для проверки работоспособности приложения, после его создания было проведено 2 вида тестирования:

- UI тесты;
- Юзабилити тесты.

6.1. UI тесты

В результате UI тестирования было выполнено 5 тестов, представленных в таблице 1, охватывающие основные возможности сайта.

Таблица 1 – Результаты UI тестирования

Шаги теста	Ожидаемый результат	Статус
1.Вход пользователя	Переход в окно пользователя	Пройден
2.Регистрация	Переход в окно пользователя	Пройден
3.Добавление местоположения в Избранные	Появление местоположения в списке	Пройден
4.Удаление местоположения из Избранных	Местоположение пропало из списка	Пройден
5.Выход пользователя	Переход в окно авторизации	Пройден

6.2. Юзабилити тестирование

Для данного вида тестирования были приглашены 2 человека, не пользовавшихся приложением ранее, для проверки работоспособности данного проекта. Цель была удостовериться в работе его основных сценариев взаимодействия с клиентами.

- регистрация;
- вход;
- добавление местоположения в избранные;
- удаление местоположения из избранных;
- поиск местоположений по названию;
- выход из аккаунта.

Основные виды тестирования были пройдены успешно. Отклонение и ошибок от ожидаемого результат в отчетах тестировавших, не обнаружено.

7. Перспективы развития Приложения

Существует несколько вариантов развития функциональности и доступности Приложения:

- Возможность просмотра более подробных прогнозов;
- Поддержка различных языков, например, английского;
- Сделать приложение кроссплатформенным;

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;
- составлена документация REST API;
- составлены диаграммы, описывающие Приложение;
- написан настоящий документ.