

# **ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»**

## **НАУЧНО-ОБРАЗОВАТЕЛЬНОЕ СОРЕВНОВАНИЕ «ШАГ В БУДУЩЕЕ, МОСКВА»**

3502

*регистрационный номер*

Информатика и системы управления

*название факультета*

ИУ-5 «Системы обработки информации и управления»

*Название кафедры*

### **Разработка автоматизированной системы подбора одежды под погоду**

**Автор:**

Тихонов Владимир Юрьевич

ГБОУ Школа №1251 имени генерала

Шарля де Голля, г. Москва, 11 «А»

класс

**Москва – 2019**

## **Оглавление**

<b>Аннотация:</b> .....	3
<b>Введение:</b> .....	4
<b>1. Аналитическая часть</b> .....	5
<b>2. Конструкторская часть</b> .....	6
<b>2.1 Графические интерфейсы:</b> .....	6
<b>2.1.1 Страница приветствия</b> .....	6
<b>2.1.2 Главная страница</b> .....	6
<b>2.1.3 Страница настроек</b> .....	6
<b>2.1.4 Рорир выбора геолокации</b> .....	7
<b>2.2 Изображения одежды (спрайты)</b> .....	7
<b>3. Технологическая часть</b> <b>3.1 Используемое программное обеспечение</b> .....	7
<b>3.2 Описание функциональных окон</b> .....	8
<b>3.2.1 Страница приветствия</b> .....	8
<b>3.2.2 Главная страница</b> .....	10
<b>3.2.3 Страница настроек</b> .....	11
<b>3.2.4 Быстрый поиск</b> .....	13
<b>3.2.5 “Календарь-слайдер”</b> .....	15
<b>4. Сравнение с аналогами</b> .....	15
<b>4.1 MeteoDress</b> .....	15
<b>4.2 Прогноз погоды от Yandex</b> .....	16
<b>4.3 MeteoLook (Мой проект):</b> .....	16
<b>Заключение</b> .....	17
<b>Ссылки</b> .....	17

**Аннотация:**

Предлагаемая автоматизированная система (АС) представлена в виде приложения (клиентской части) и сервера.

Цель работы – разработка АС, решающей задачу подбора одежды под погоду.

Разработанная АС, сопоставляя погодные данные со стороннего сервиса между собой, подбирает одежду из локального каталога на 7 дней и формирует под каждый из них отдельный комплект одежды.

АС легка в использовании и может быть установлена практически на все мобильные устройства, управляемые операционной системой Android.

Инструменты программной реализации:

1. Клиент

- Apache Cordova (Web View)
- Angular 6 + Redux (NgRx)
- TypeScript
- HTML5 + CSS3

2. Сервер

- Nginx
- Python 3.6
- Django + PIL

## **Введение:**

Во все времена большую роль в жизни играет здоровье. Есть много способов бороться с ОРВИ: витамины в таблетках, закалка, вакцинация. Но самый лучший – это правильно подбирать одежду под разные погодные условия, чтобы не перегреваться в жару и не переохлаждаться в мороз.

Чаще всего люди ориентируются на прогноз погоды, но выходя на улицу понимают, что по ощущениям там намного холоднее, чем ожидалось, или же смотрят за окно, что является совсем неинформативным. В регионах с часто меняющимися погодными условиями такая проблема до сих пор остается насущной. Особенно это актуально для детей и подростков.

Целью моей работы является создание автоматизированной системы (АС) по подбору одежды под погоду.

В процессе разработки АС были поставлены и выполнены следующие задачи:

- Анализ актуальности проблемы на основе пользовательских опросов
- Создание бумажного прототипа
- Пользовательское тестирование и выявление недостатков системы
- Выбор технологий
- Создание Minimum Viable Product (MVP)
- Тестирование MVP на реальных пользователях
- Анализ обратной связи
- Доработка продукта до стадии Production

## 1. Аналитическая часть

Основной целью проекта является создание автоматизированной системы подбора одежды под погоду. Для решения проблемы подбора одежды под погоду в основном используется прогноз погоды, который иногда бывает неточным, а главное, не дает наглядного представления, как одеваться.

Поэтому было принято решение создать наглядную систему с отображением набора одежды на день, а также полезными советами. Для удобства использования системы, ее клиентская часть была обернута в мобильное приложение, а серверная часть расположилась на удаленной машине под управлением Linux (Debian 8).

Мобильное приложение должно: обладать графическим интерфейсом и позволять использовать изображения в формате Base64, обладать функцией поиска по географическим объектам, уметь отправлять запросы на сторонние сервера и обрабатывать ответы, а также хранить данные в локальном хранилище. Так же должна быть реализована функция настройки профиля пользователя, включающая в себя изменение геолокации и гендера.

Серверная часть должна: иметь хранилище картинок (спрайтов), уметь принимать, отправлять, обрабатывать запросы и отдавать статику, иметь алгоритм подбора одежды основываясь на погодных факторах (температура, ветер, влажность).

Опираясь на данные рассуждения можно сделать вывод, что для решения поставленной задачи требуется:

- Проработать и написать алгоритм для подбора одежды по погодным данным
- Наладить связь с со сторонними API сервисами (DarkSky Api + Open StreetMap)
- Выбор технологий под написание сервера

- Нарисовать картинки одежды (спрайты)
- Становить сервер на удаленную машину
- Выбрать технологии под написание клиента
- Связать клиента с сервером по методу json
- Создать алгоритма отображения
- Организовать хранения данных на клиенте
- Создать алгоритм обновления данных на клиенте
- Создать алгоритм подбора одежды по погодным данным, которые относятся к введенному пользователем географическому объекту

## **2. Конструкторская часть**

### **2.1 Графические интерфейсы:**

Для отображения контента требуется использовать язык программирования, имеющий возможность создания графического интерфейса, поддерживающего изображения в формате Base64.

Для навигации по приложению в графическом интерфейсе должны быть разработаны следующие разделы:

#### **2.1.1 Страница приветствия**

Страница приветствия показывается только при первом запуске приложения, в ней пользователь вводит свое постоянное местоположение и выбирает гендер.

#### **2.1.2 Главная страница**

Главная страница приложения. Отвечает за отображение контента, на ней так же находится поле поиска географических объектов.

#### **2.1.3 Страница настроек**

На данной странице можно изменить свое постоянное местоположение и гендер.

#### **2.1.4 Рорир выбора геолокации**

Всплывающее окно для выбора геолокации. Открывается, когда пользователь ищет что-либо и предлагает выбрать нужный географический объект из списка совпадений.

#### **2.2 Изображения одежды (спрайты)**

Изображения, из которых по алгоритму собираются комплекты одежды, хранятся на сервере.

### **3. Технологическая часть**

#### **3.1 Используемое программное обеспечение**

Back-end приложения был развернут на удаленной машине под управлением операционной системы Linux (Debian 8). За основу веб-сервера был взят Nginx + Gunicorn. Это было обусловлено тем, что данный веб-сервер в разы производительнее своих конкурентов, таких как Apache. В будущем это позволит масштабировать проект без дополнительных затрат. Для разработки серверной части системы был выбран язык Python 3. Это было связано с тем, что под него была написана очень удобная библиотека для работы с изображениями PIL (Python Image Library).

Front-end приложения был написан с использованием технологии WebView на языке TypeScript, на фреймворке Angular 6 с использованием Redux flow (NgRx), что в дальнейшем позволит быстро масштабировать функционал проекта. Технология WebView была выбрана неспроста. Это позволило писать код один раз, а при “билде” проекта выбирать, под какую операционную систему телефона собирать приложение. Тем самым не придется писать приложение с нуля под различные платформы.

Графическая составляющая приложения писалась на HTML5 + CSS3.

В целом технологии для front-end`а были выбраны не случайно, дело в том, что уже имелся опыт написания подобных приложений, только в web.

## 3.2 Описание функциональных окон

Приложение было написано с использованием подхода SPA (Single Page Application), это означает то, что приложение имеет всего одну страницу, а нужный контент подгружается и изменяется динамически.

### 3.2.1 Страница приветствия

Первая страница, которую видит пользователь при установке приложения на свое устройство – это страница приветствия. На данной странице находится небольшая форма для получения параметров пользователя: его местоположение и гендер (данные поля обязательны для заполнения).

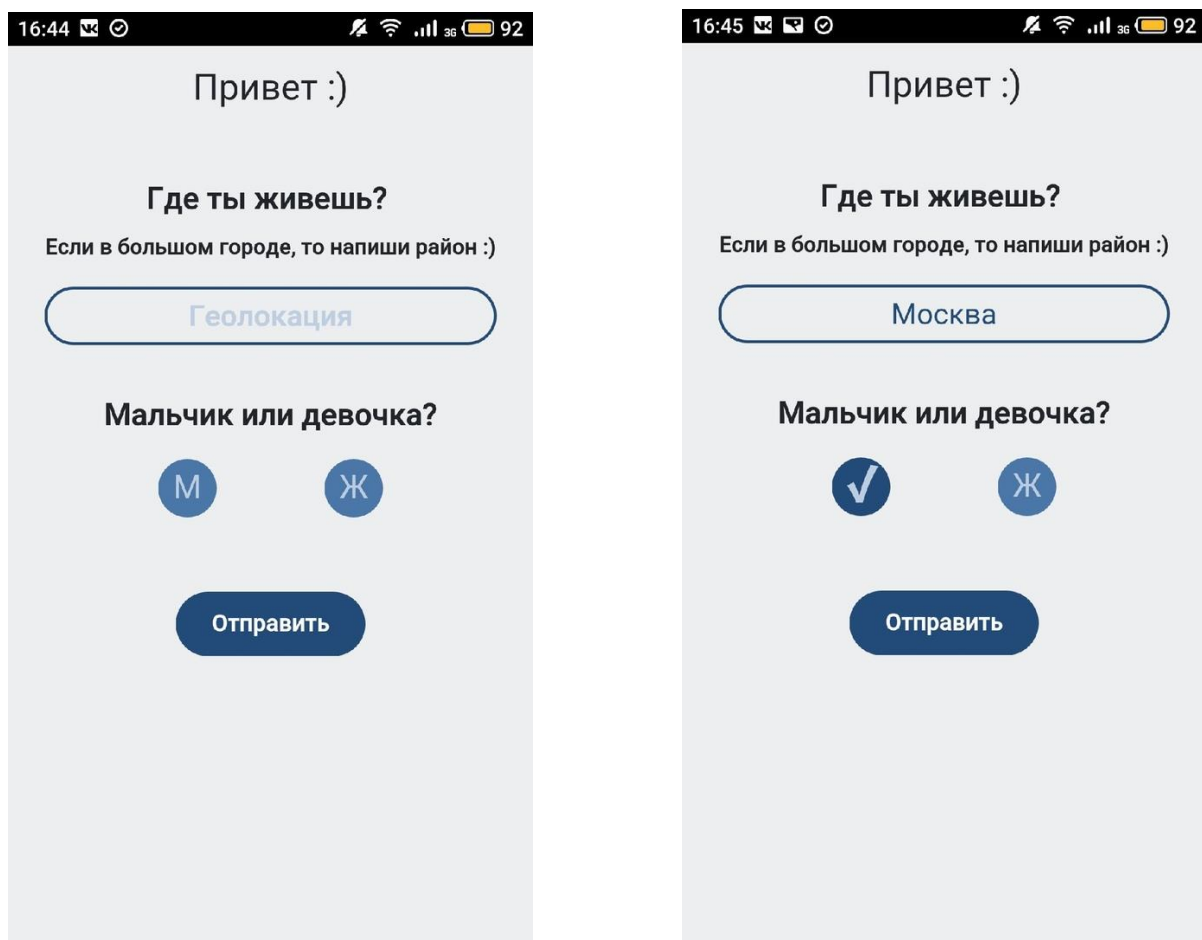


Рисунок 1. Пустая и заполненная страница приветствия.

После заполнения формы пользователь нажимает на кнопку “Отправить”, после чего отправляется запрос на сторонний API сервер (OpenStreetMap) с одним параметром (местоположение). Ответом на наш



запрос приходит JSON строка со списком совпадений (максимум может прийти 10 совпадений) строки из нашего запроса с существующими географическими объектами. Далее автоматически открывается Rorur, где можно выбрать интересующий нас географический объект. Если совпадений нет, то список будет пуст.

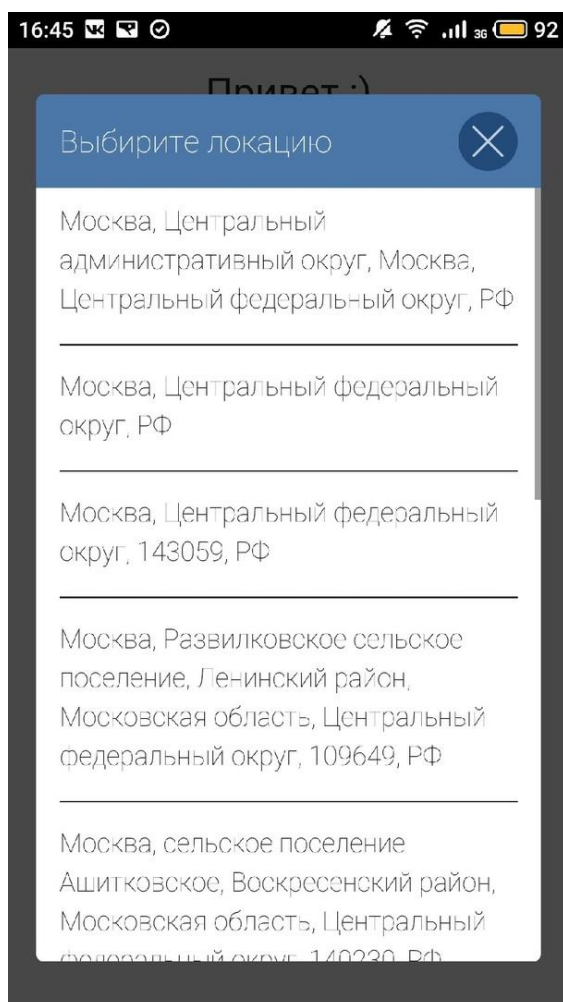


Рисунок 2. Список совпадений.

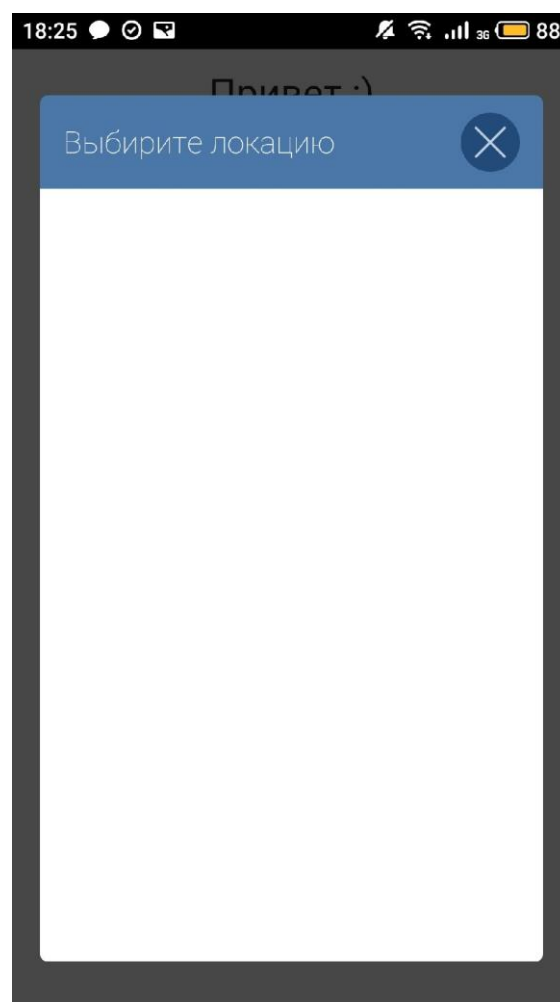


Рисунок 3. Пустой список

После выбора интересующего нас географического объекта приложение обрабатывает форму, сохраняет данные пользователя в локальном хранилище и отправляет запрос на наш сервер для того, чтобы получить комплект одежды, температуру и прочую информацию по погоде.

Далее после успешного получения всей нужно информации нас переносит на главную страницу приложения.

### 3.2.2 Главная страница

Главную страницу приложения пользователь видит при всех последующих заходах. На главной странице можно увидеть следующую информацию:

- Поле поиска (По умолчанию в нем указана настоящая геолокация)
- Кнопка для открытия страницы настроек
- Картинка с комплектом одежды
- Слайдер с выбором интересующего дня

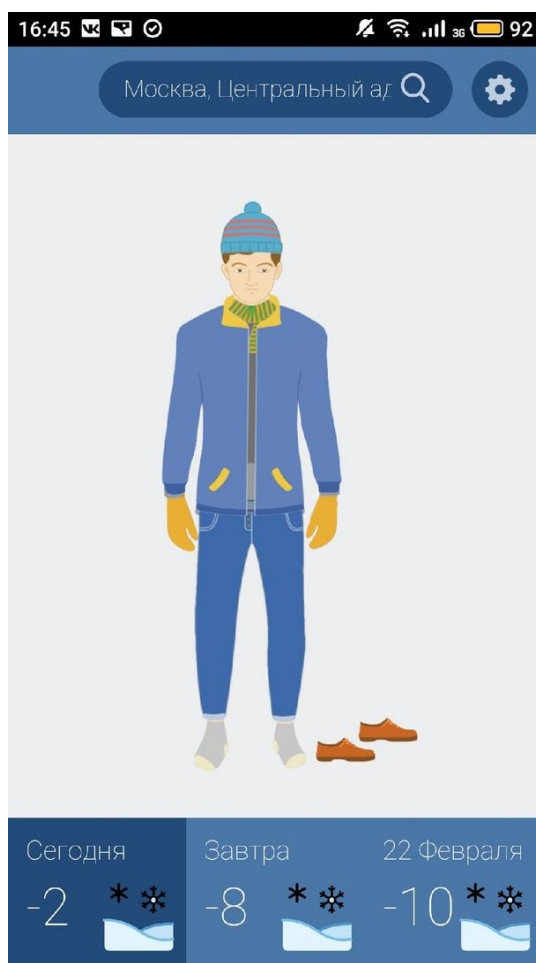


Рисунок 4. Главная страница

### 3.2.3 Страница настроек

Для изменения настроек пользователя есть отдельный рорир. Для его открытия нужно нажать на значок шестеренки находящийся в правом верхнем углу.



Рисунок 5. Кнопка открытия настроек

При нажатии на данную кнопку открывается рорир, где можно изменить параметры пользователя:

- Местоположение
- Гендер.

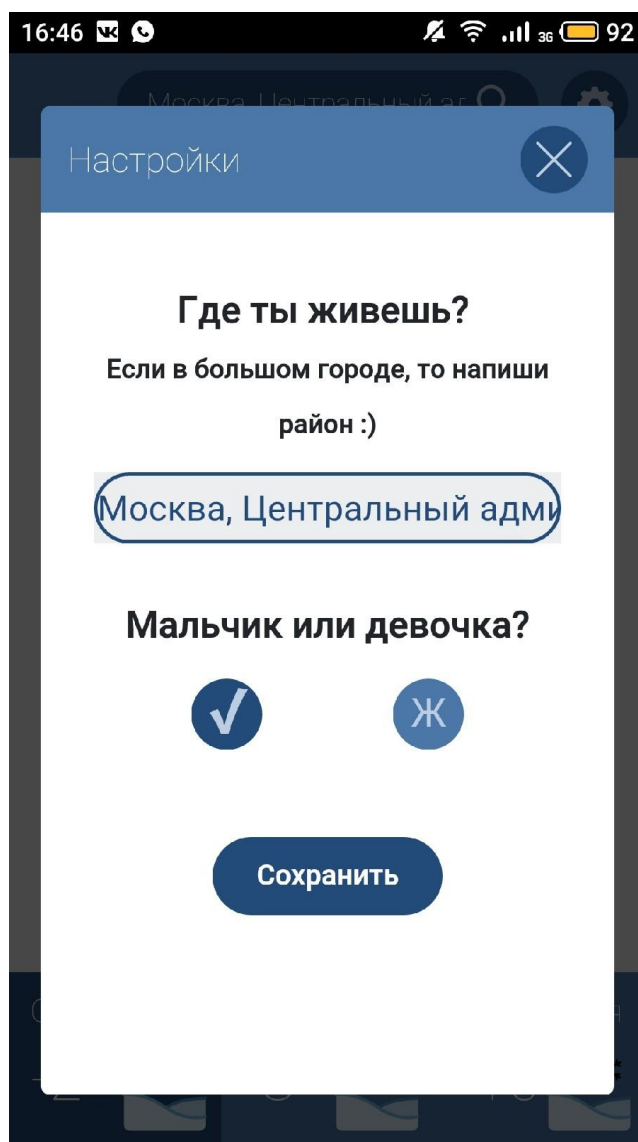


Рисунок 6. Окно настроек пользователя

Для изменения постоянной геолокации, нужно ввести название географического объекта в поле ввода под заголовком “Где ты живешь?”. Для примера введем “Амстердам”.

Для изменения гендера нужно нажать на кнопку находящуюся под заголовком “Мальчик или девочка?”.

Далее для сохранения настроек нужно нажать на кнопку “Сохранить”, после чего отправляется запрос на сервер (OpenStreetMap), который возвращает совпадения со строкой поиска, далее всплывет еще один роруп для выбора конкретного географического объекта:

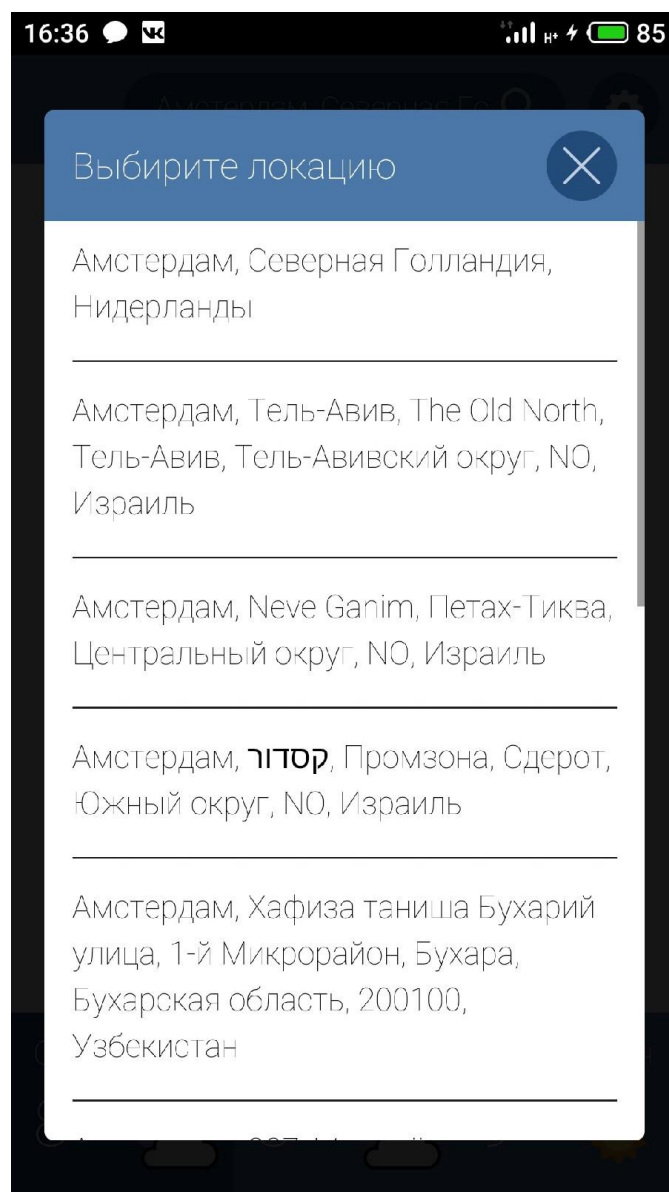


Рисунок 7. Выбор конкретной геолокации

После выбора интересующего нас географического объекта, настройки сохраняются в локальном хранилище устройства. Далее приложение отправляет запрос на сервер, чтобы получить новые данные (комплекты одежды и информацию о погоде) на семь дней вперед. При этом уже существующие данные в хранилище обновляются и мы видим новую информацию.

### 3.2.4 Быстрый поиск

Для того чтобы каждый раз не открывать окно настроек существует быстрый поиск по географическим объектам. Строка быстрого поиска

находится в верхней центральной части экрана:



Рисунок 8. Окно быстрого поиска

Для поиска требуется один раз нажать на поле ввода, после чего откроется клавиатура. Далее нужно ввести новый географический объект, допустим Лондон, после чего нажать на иконку лупы в правой части поля ввода. Далее откроется роупер с выбором конкретной геолокации, по аналогии с рисунком №7. Далее выбираем интересующий нас географический объект. В отличие от изменения постоянного географического положения, быстрый поиск не заносит новое положение в настройки пользователя. Он только показывает комплекты на семь дней вперед.



Рисунок 9. Быстрый поиск и кнопка “Назад”

Чтобы вернуться обратно, достаточно нажать на кнопку “Назад”, находящуюся в левом верхнем углу, выглядящую как стрелочка, направленная влево (Рисунок 9).

### 3.2.5 “Календарь-слайдер”

Для выбора доступных для просмотра дней, был создан “календарь-слайдер”. Данный блок находится внизу экрана.

При нажатии на карточки слайдера, переключается картинка комплекта одежды. Выбранный день выделен более темным цветом, в отличие от других. На карточке слайдера присутствует информация о дне и погоде:

- Дата
- Температура (в градусах Цельсия)
- Погодная иконка

Слайдер можно крутить вправо и влево, всего в нем присутствует 7 карточек с информацией.



Рисунок 10. “Календарь-слайдер”

## 4. Сравнение с аналогами

### 4.1 MeteoDress

#### Плюсы:

- Прогноз с интервалом в 3 часа
- Больше одежды
- Более точный алгоритм

#### Минусы:

- Прогноз только на 5 дней
- Платный
- Только под устройства IOS
- Работает не на всех версиях IOS

#### **4.2 Прогноз погоды от Yandex**

##### **Плюсы**

- Высокая точность прогноза
- Высокая детализация

##### **Минусы**

- Нет наглядного примера как одеваться

#### **4.3 MeteoLook (Мой проект):**

##### **Плюсы:**

- Бесплатный
- Прогноз на 7 дней
- Работает на всех устройствах Android версии 4.0.x и выше

##### **Минусы:**

- Мало одежды
- Нет детализации по часам
- Алгоритм несовершенен



## **Заключение**

Данное приложение может быть использовано людьми любого возраста и пола, живущими в любой точке мира. Особенно актуально для детей и подростков, а также для часто путешествующих.

Результатом работы стало мобильное приложение, подбирающее одежду согласно погодным условиям. Получившийся продукт имеет ряд преимуществ перед аналогами. Так же есть и ряд недостатков, которые в процессе доработки проекта будут исправлены, а именно будет совершенствоваться алгоритм подбора одежды, будет добавлено больше одежды.

## **Ссылки**

По ходу написания проекта обращался за помощью к следующим ресурсам:

<https://ru.stackoverflow.com/>

<https://learn.javascript.ru/>

<https://cordova.apache.org/>

<https://angular.io/>

API, используемые в приложении:

<https://www.openstreetmap.org>

<https://darksky.net/dev>