Начать

SkyCast - приложение для отслеживания погоды

SkyCast
Weather Apps

Выполнили студенты 245 гр.
Лапин Кирилл
Сокол Илья
Москвитин Дмитрий

SkyCast
Weather App

Область

Предметная область

SkyCast
Weather App

Область

Основные цели:

- Актуальная информация о текущей погоде
- Актуальная информация о прогнозе погоды
- Отслеживание погоды в различных городах

SkyCast
Weather App

Область

- Получение текущей погоды
- Почасовой прогноз погоды
- Прогноз на несколько дней
- Визуализация метеорологических характеристик при помощи иконок, графиков и карт
- Отслеживание погоды в нескольких местах
- Возможность сохранять и удалять из избранные места

SkyCast
Weather App

Область

Источники данных:

- API погодных служб, такие как WeatherAPI или OpenWeatherMap
- Данные с метеорологических станций
- Данные с метеорологических спутников

SkyCast
Weather App

Область

Метеорологические характеристики:

- Температура
- Осадки
- Влажность
- Ветер
- Атмосферное давление
- Облачность
- Видимость

SkyCast
Weather App

База данных

SkyCast
Weather App

Список сущностей:

Используется две сущности базы данных с соответствующим описанием

№	Сущность	Назначение
1	user_entity	Перечень пользователей, зарегистрированных в приложении
2	chosen_city	Перечень избранных городов пользователей

SkyCast
Weather App

Список атрибутов сущности:

Рассмотрим список атрибутов таблицы user_entity

Ключевое поле	Атрибут	Назначение
ПК	id	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому пользователю. Значения автоматически генерируются СУБД при вставке новой записи в таблицу.
	username	
	password	

SkyCast
Weather App

Список атрибутов сущности:

Рассмотрим список атрибутов таблицы chosen_city

Ключевое поле	Атрибут	Назначение
ПК	id	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому городу. Значения автоматически генерируются СУБД при вставке новой записи в таблицу.
	city_name	
ВК	user_id	

SkyCast
Weather App

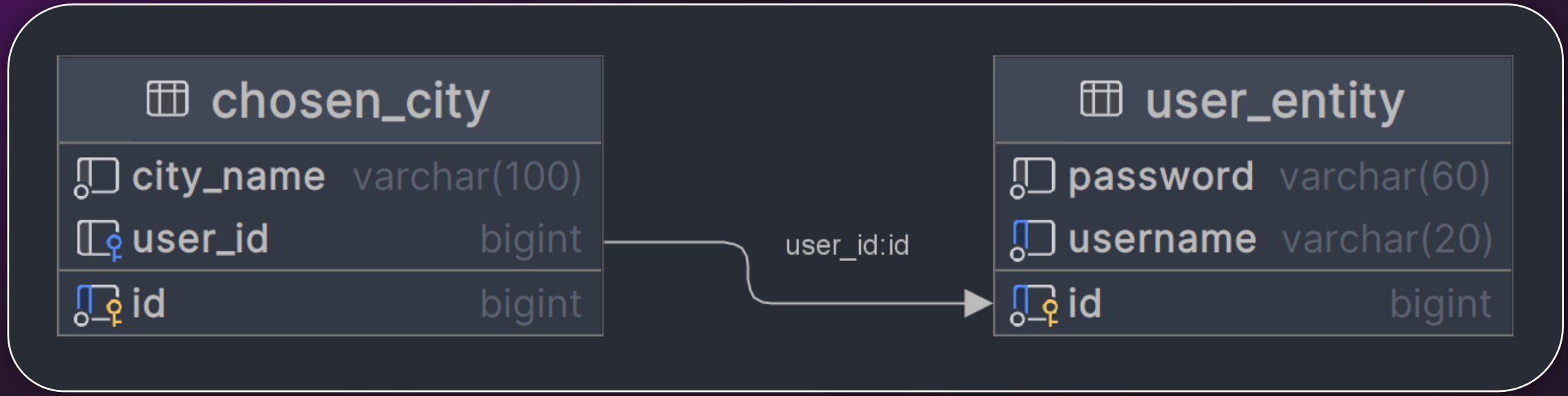
База данных

Список связей сущностей:

Рассмотрим классы сущностей и их свойства

№	Сущности, участвующие в связи	Тип связи	Обоснование
1	user_entity – chosen_city	1:N	Каждый пользователь может добавить несколько избранных городов, но каждый избранный город может относиться только к одному пользователю

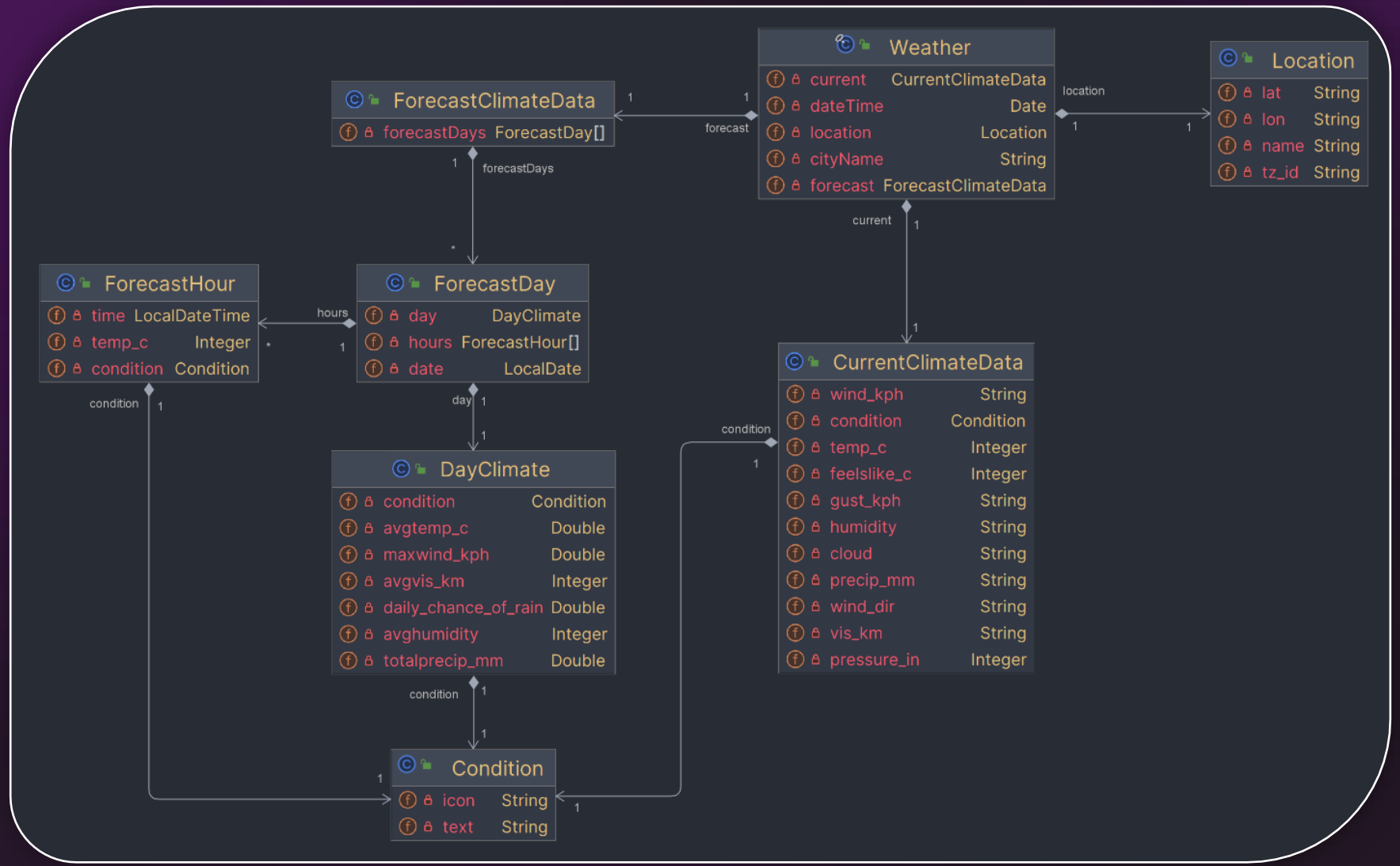
В результате получили:



SkyCast
Weather App

Архитектура

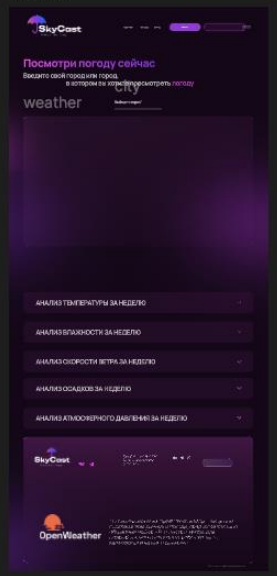
Архитектура



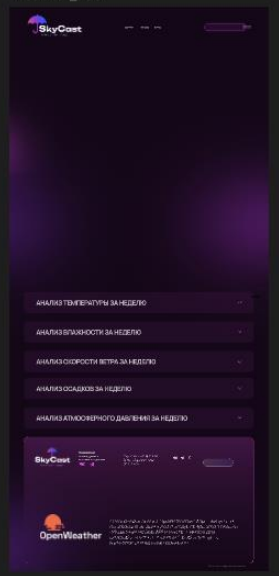
Архитектура

Main (+master)

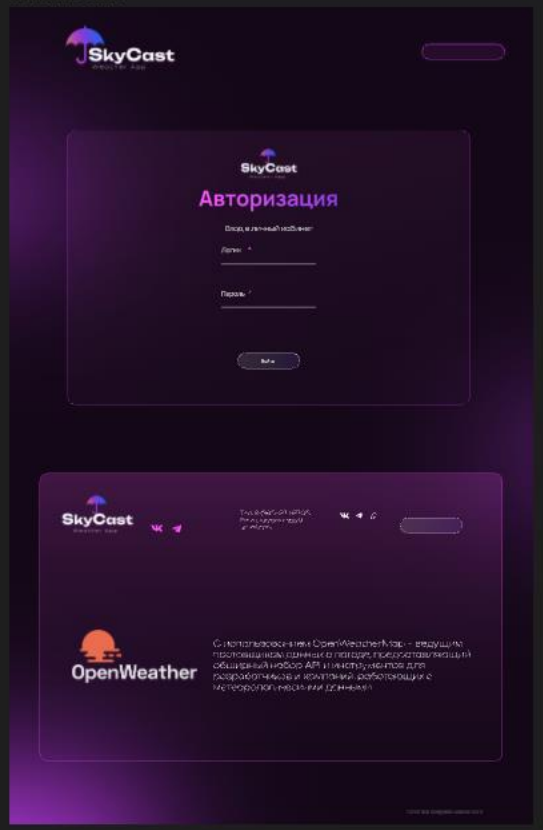
Weather



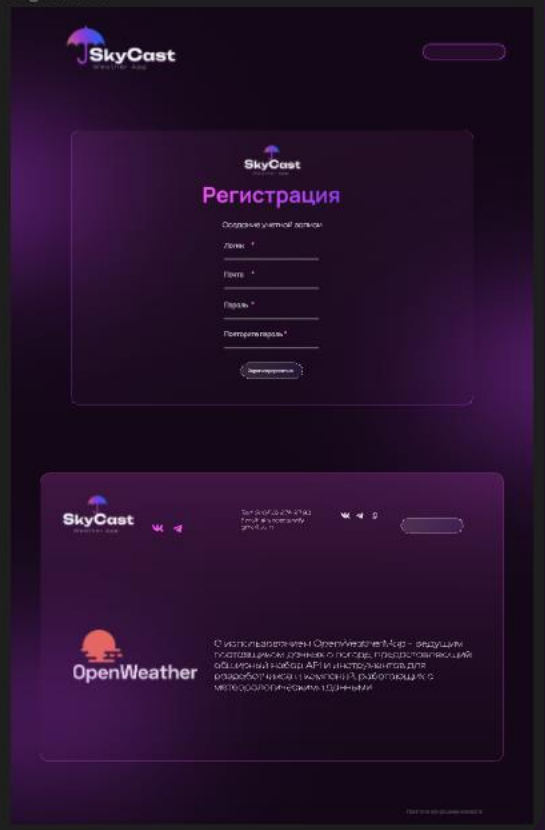
Weather_user



authorization

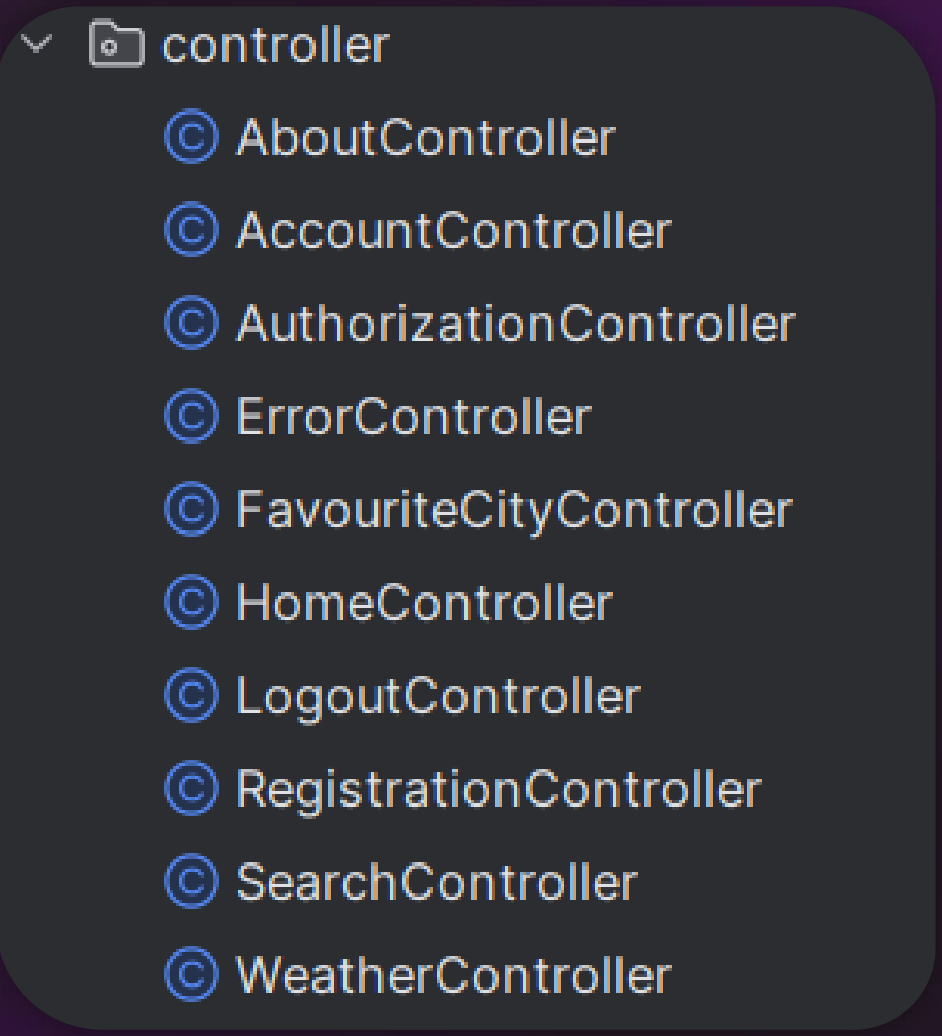


registration

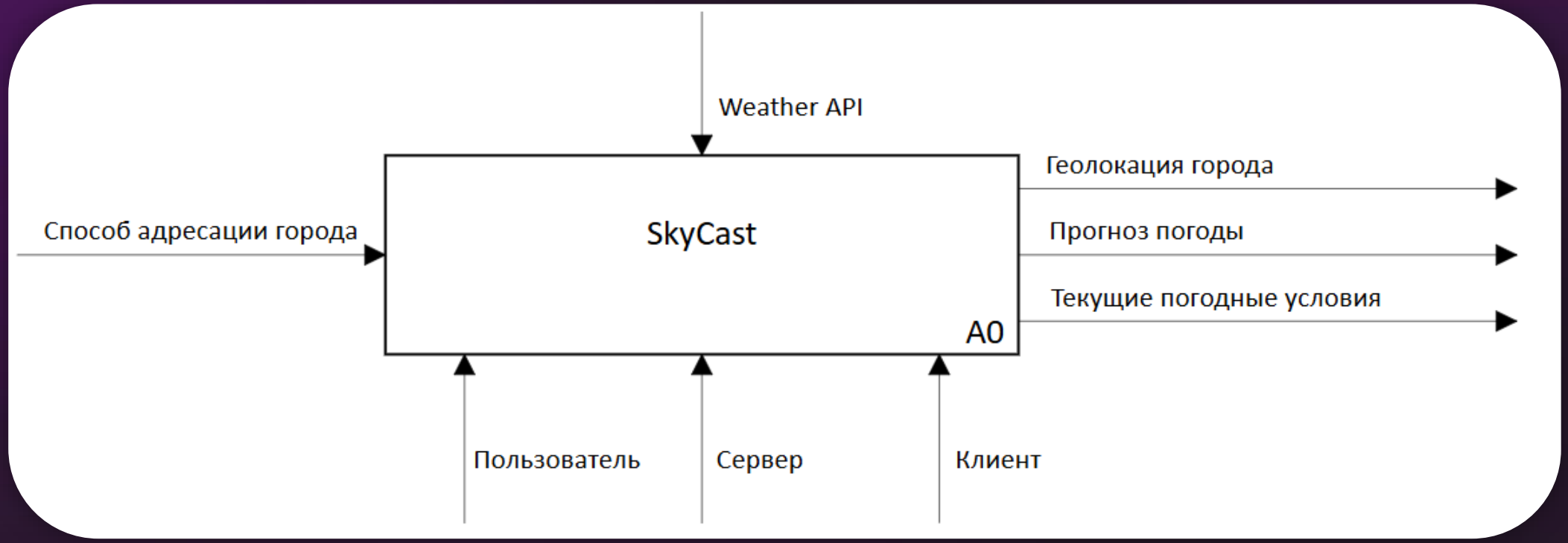


Контроллер:

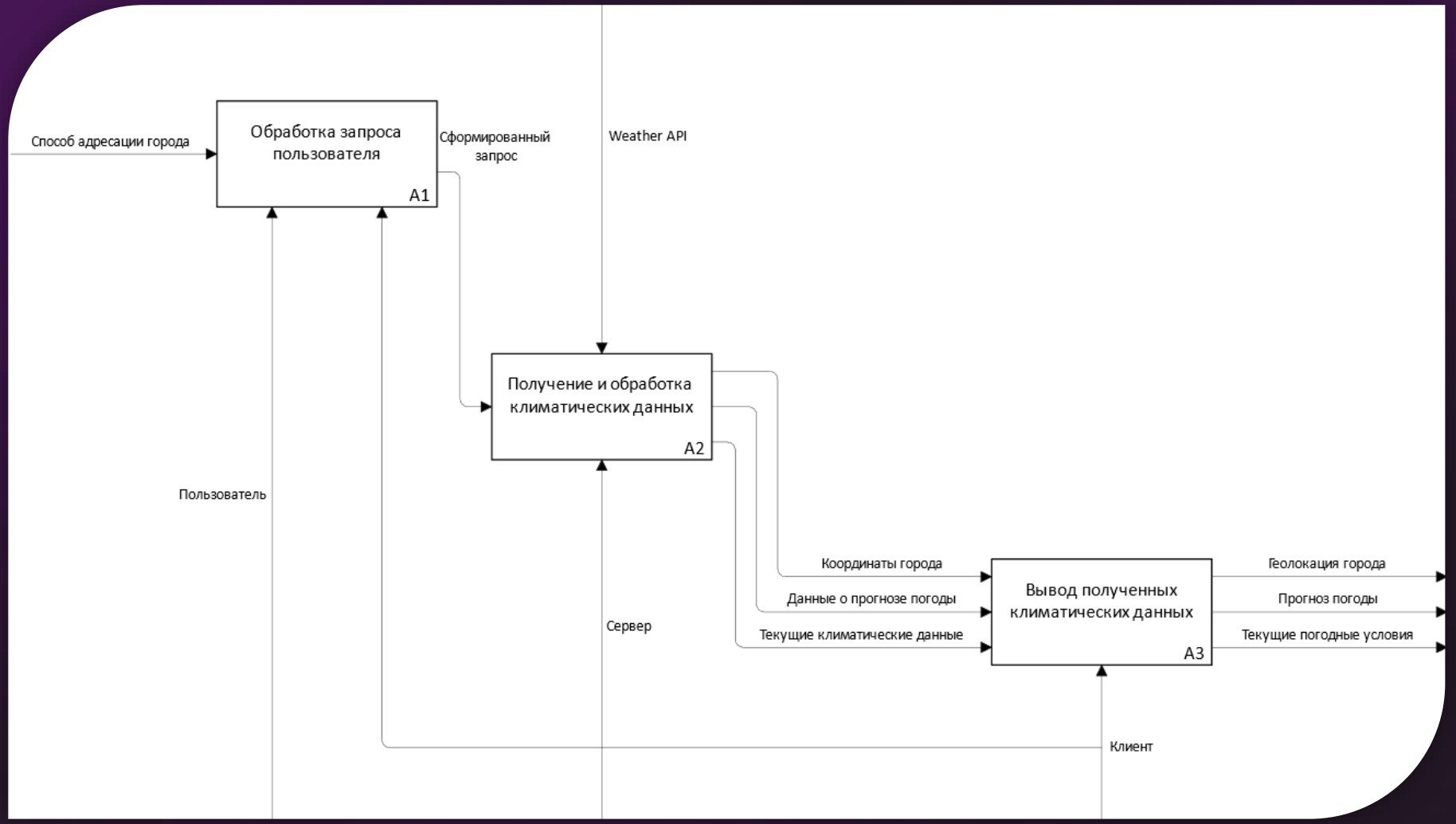
Контроллер отвечает за обработку пользовательских запросов



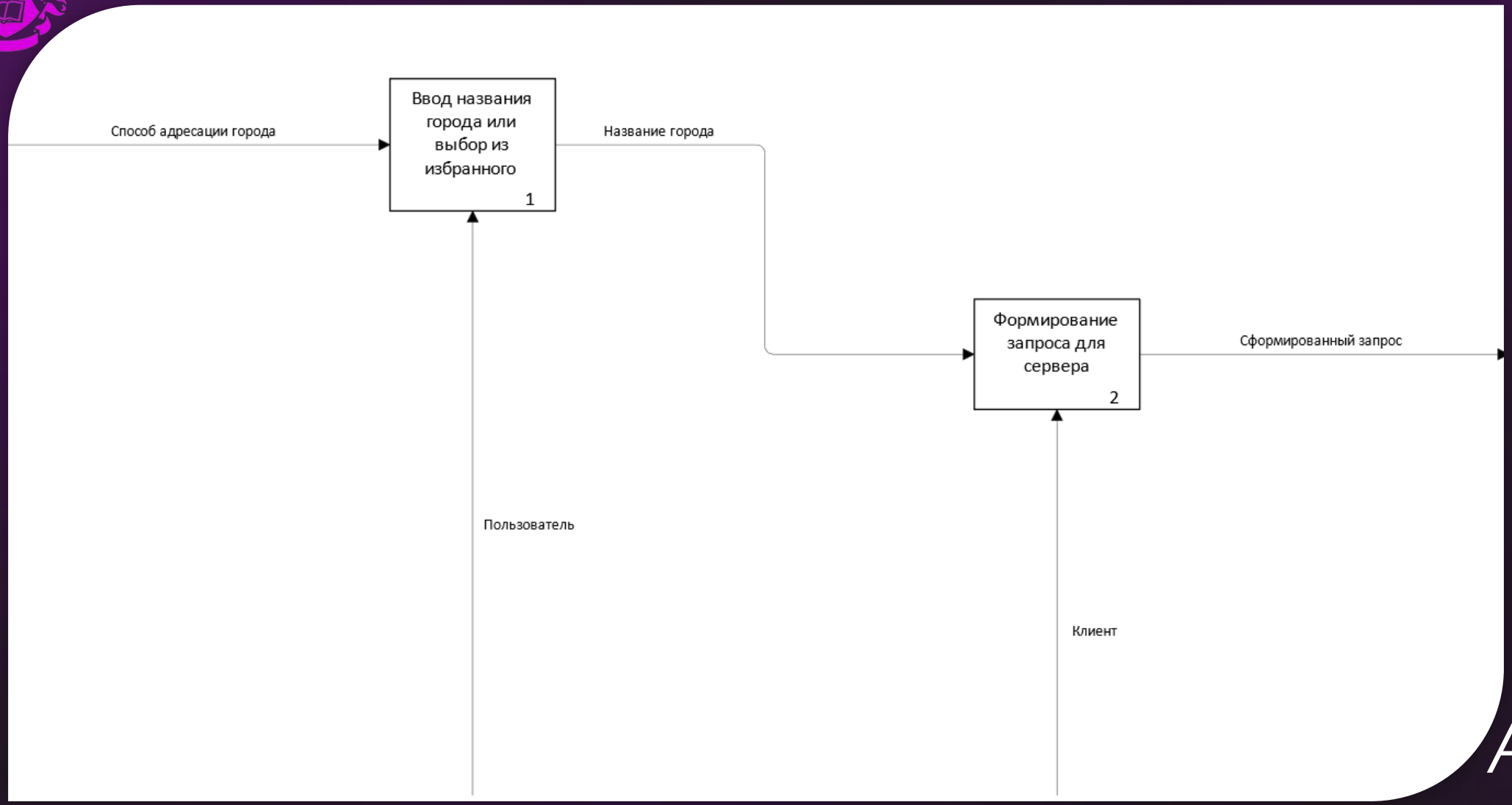
Архитектура



Архитектура

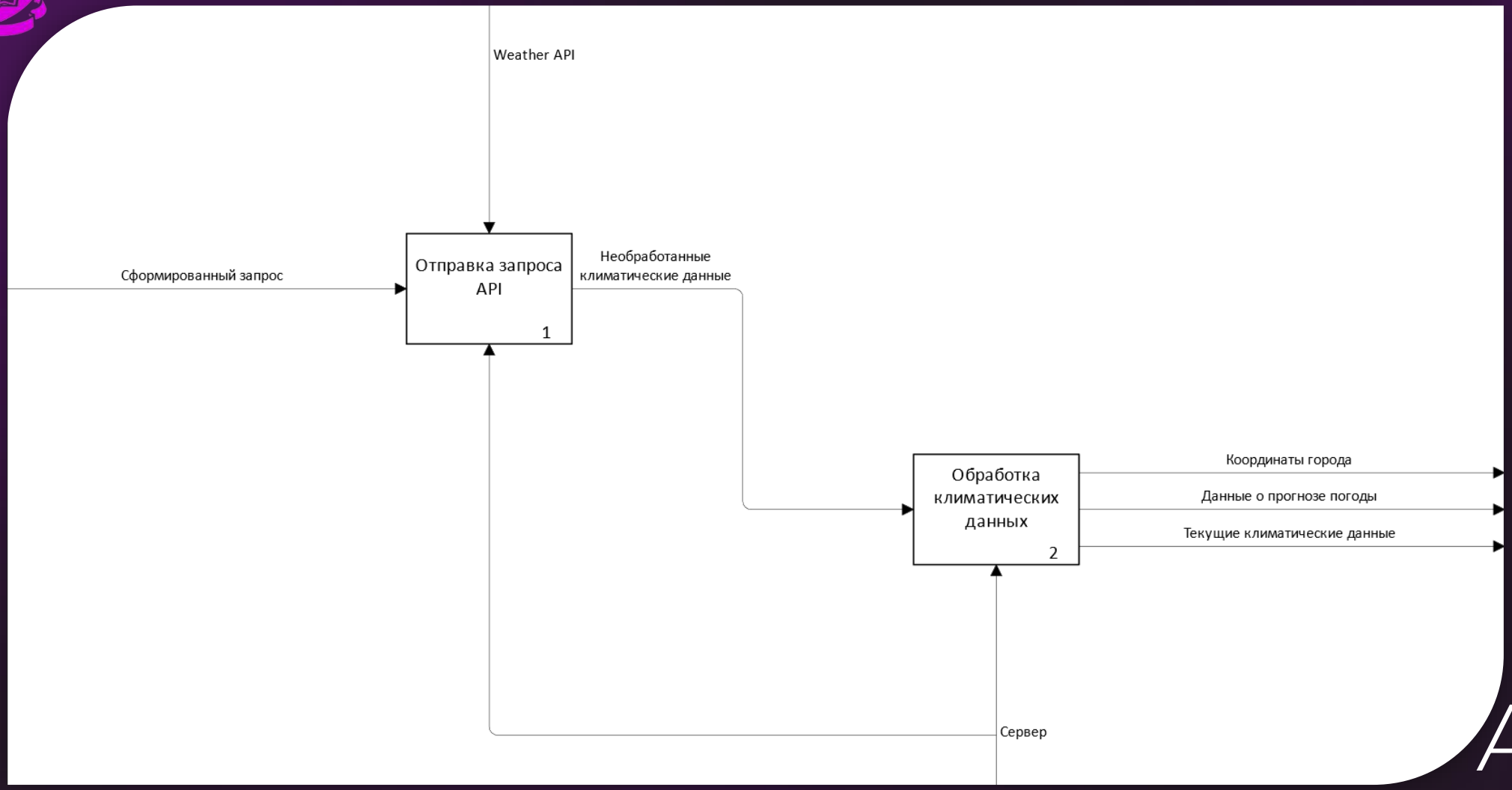


A0


Способ адресации города
Ввод названия города или выбор из избранного
1
Название города
Формирование запроса для сервера
2
Сформированный запрос
Пользователь
Клиент

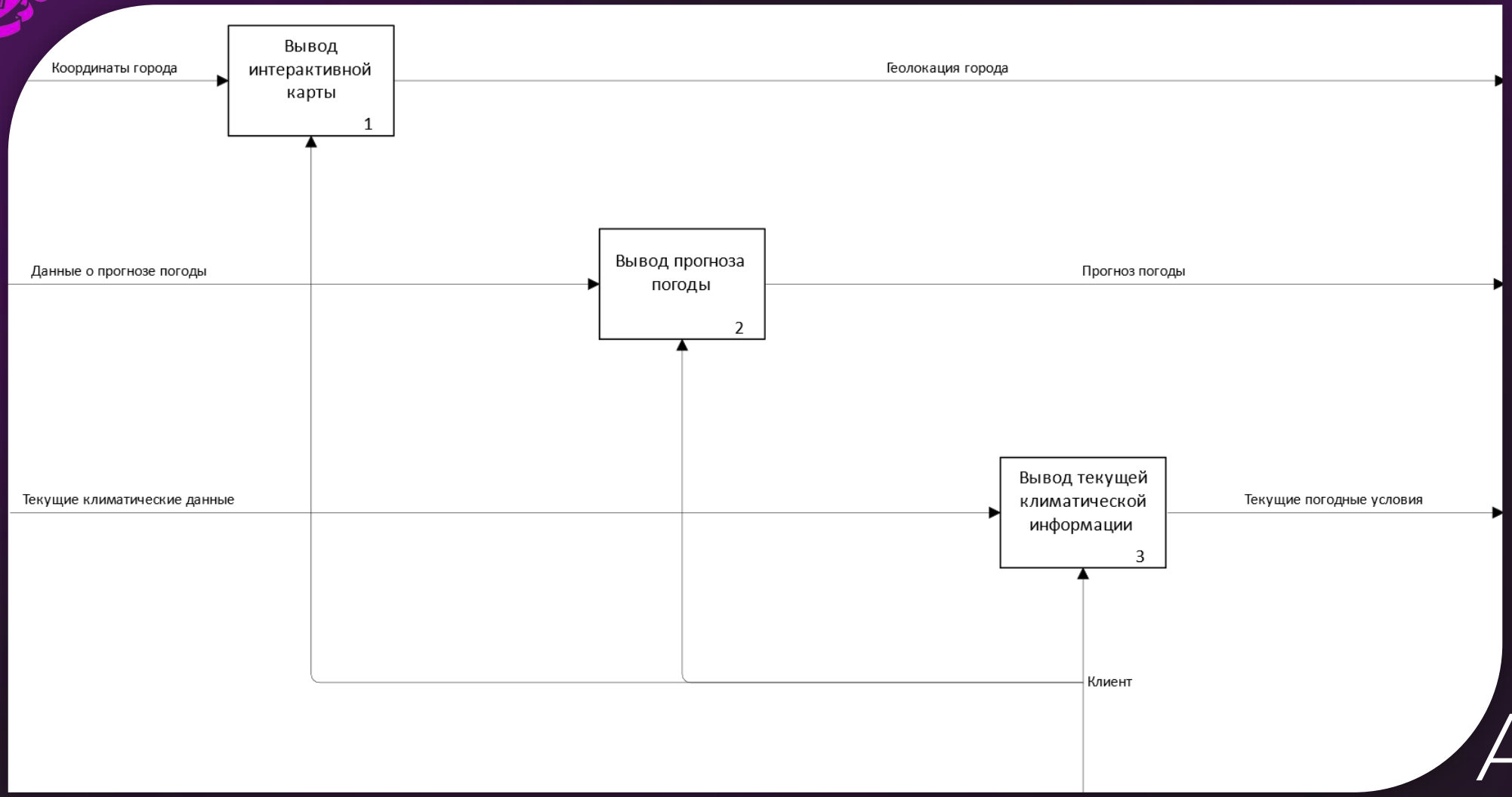
A1

Архитектура



A2

Архитектура



А3

SkyCast
Weather App

Технологии

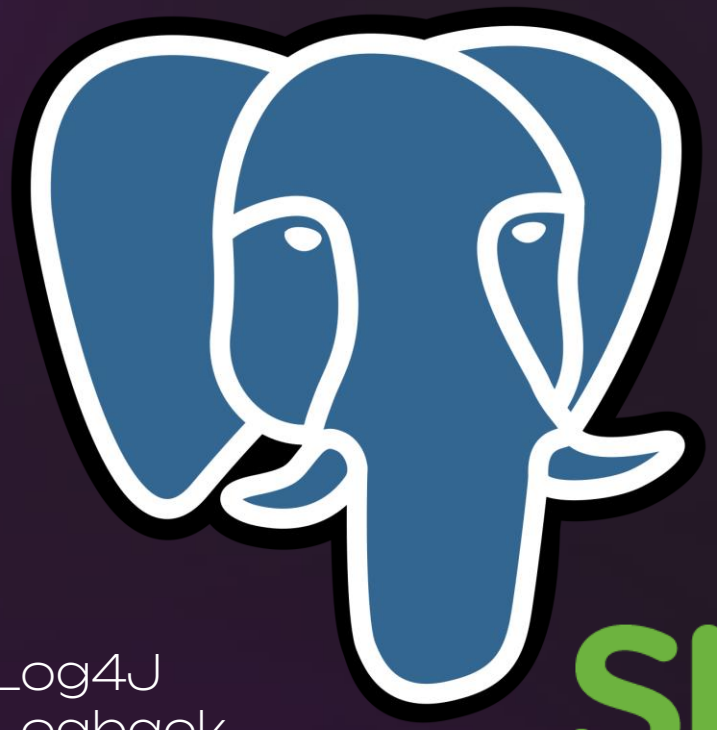
Стек технологий

SkyCast
Weather App

Технологии

Технологии Backend разработки:

- Spring Boot
- Spring Boot DevTools
- Spring Web
- Spring Security
- Spring Validation
- Spring Data JPA
- Project Lombok
- H2 Data Base
- PostgreSQL
- Apache Tomcat
- HikariCP
- Log4J
- Logback
- Jackson



SkyCast
Weather App

Технологии

Технологии тестирования:

- AssertJ
- Junit 5
- Mockito

mockito

JUnit 5

SkyCast
Weather App

Технологии

Технологии Frontend разработки:

- Thymeleaf
- HTML 5
- CSS 3
- JavaScript

Thymeleaf

SkyCast
Weather App

Технологии

Технологии Frontend разработки:

- Adobe Photoshop
- Figma
- Pixso

SkyCast
Weather App

Команда

SkyCast
Weather App

Команда

Илья Сокол

Backend-разработчик
Тестировщик

Кирилл Лапин

Frontend-разработчик
Веб-дизайнер

Дмитрий Москвитин

Аналитик

SkyCast
Weather App

Команда

KANBAN: Методология гибкой разработки

это гибкий подход к разработке, который фокусируется на визуализации и управлении потоком работы.

Интересные задачи

SkyCast
Weather App

Задачи

Трудные задачи

SkyCast
Weather App

Добавление

UserService

```
/**
* Используется для добавления города с названием <code>cityName</code>
* в список избранных городов пользователя <code>user</code>
* @param user пользователь, в список которого необходимо добавить избранный город
* @param cityName название добавляемого города
* @return измененная сущность <code>UserEntity</code>
* @throws CityIsAlreadyFavourite если добавляемый город уже является избранным
* для <code>user</code>
*/
@Transactional
public UserEntity addChosenCity(UserEntity user,
                                String cityName) throws CityIsAlreadyFavourite {

    // Получаем список избранных городов пользователя
    List<ChosenCity> chosenCities = user.getChosenCities();

    // Если город уже является избранным для данного пользователя
    if (isFavouriteCity(chosenCities, cityName)) {
        throw new CityIsAlreadyFavourite("Данный город уже является избранным
          для пользователя" + user.getUsername());
    }
```

UserService

```
    // Создаем новый кортеж на основе названия города
    ChosenCity chosenCity = new ChosenCity(cityName);

    // Добавляем избранный город пользователю
    user.addChosenCity(chosenCity);

    // Сохраняем изменения в базе данных
    chosenCityRepository.save(chosenCity);

    return user;
}
```

SkyCast
Weather App

Удаление

UserService

```java
/**
 * Используется для удаления города с названием <code>cityName</code>
 * из списка избранных городов пользователя <code>user</code>
 * @param user пользователь, из списка которого необходимо добавить избранный город
 * @param cityName название удаляемого города
 * @return измененная сущность <code>UserEntity</code>
 * @throws CityIsNotFavourite если удаляемый город не является избранным для
 * <code>user</code>
 */
@Transactional
public UserEntity deleteChosenCity(UserEntity user,
                                   String cityName) throws CityIsNotFavourite {

    // Получаем список избранных городов пользователя
    List<ChosenCity> chosenCities = user.getChosenCities();

    // Если город не является избранным для данного пользователя
    if (!isFavouriteCity(chosenCities, cityName)) {
        throw new CityIsNotFavourite("Данный город не является избранным для
          пользователя" + user.getUsername());
    }

    // Получаем из БД сущность города по названию и идентификатору пользователя
    Optional<ChosenCity> chosenCityToDeleteOptional = chosenCityRepository
            .findByCityNameAndUser(cityName, user);
```

UserService

```java
    // Получаем объект из сущности
    ChosenCity chosenCityToDelete = chosenCityToDeleteOptional.get();

    // Дополняем объект для корректного удаления
    chosenCityToDelete.setUser(user);

    // Удаляем город из избранных городов пользователя
    user.removeChosenCity(chosenCityToDelete);

    // Сохранение изменений в БД
    userRepository.save(user);

    // Удаляем город из базы данных
    chosenCityRepository.deleteById(chosenCityToDelete.getId());

    return user;
}
```

SkyCast
Weather App

Результат

. . . а что СДЕЛАНО?

SkyCast
Weather App

Результат

Преимущества:

- Современный стек технологий
- Интуитивный интерфейс
- Актуальные данные
- Гибкая архитектура

Недостатки:

- Ограниченная функциональность
- Зависимость от API

SkyCast
Weather App

Демонстрация

Демонстрация приложения