

Курсова работа

По дисциплина „Мобилни приложения“

На тема: „Метеорологично приложение“

Изготвил: Ерика Карамучева

Факултетен номер: 2101321067

Специалност: “Софтуерно инженерство” - редовно

Проверил: ас. Маргарита Гочева

***Съдържание***

1. [Увод](#Увод)
2. [Основни функционалности](#Функционалности)
3. [Използвани технологии и библиотеки](#Техонологии)
4. [Потребителски инструктаж](#Инструктаж)
5. [Архитектура на приложението](#Архитектура_на_приложението)
6. [Имплементация](#Имплементация)
7. [Заключение](#Заключение)
8. [Библиография](#Библиография)
9. [Списък на фигурите](#Списък_фигури)
10. ***Увод***

Целта на курсовия проект е да се изгради мобилно приложение, което да извлече данни за текущото време, за избран от потребителя град. Дава се възможност на потребителя да съхрани извлечените данни, по желание да ги редактира и евентуално да ги изтрие.

1. ***Основни функционалности***

● Намиране и извеждане на времето на град по име

● Записването на данни по желание на потребителя

● Възможност за редактиране на данните

● Възможност за изтриване на данните

1. ***Използвани технологии и библиотеки***

● Android Studio

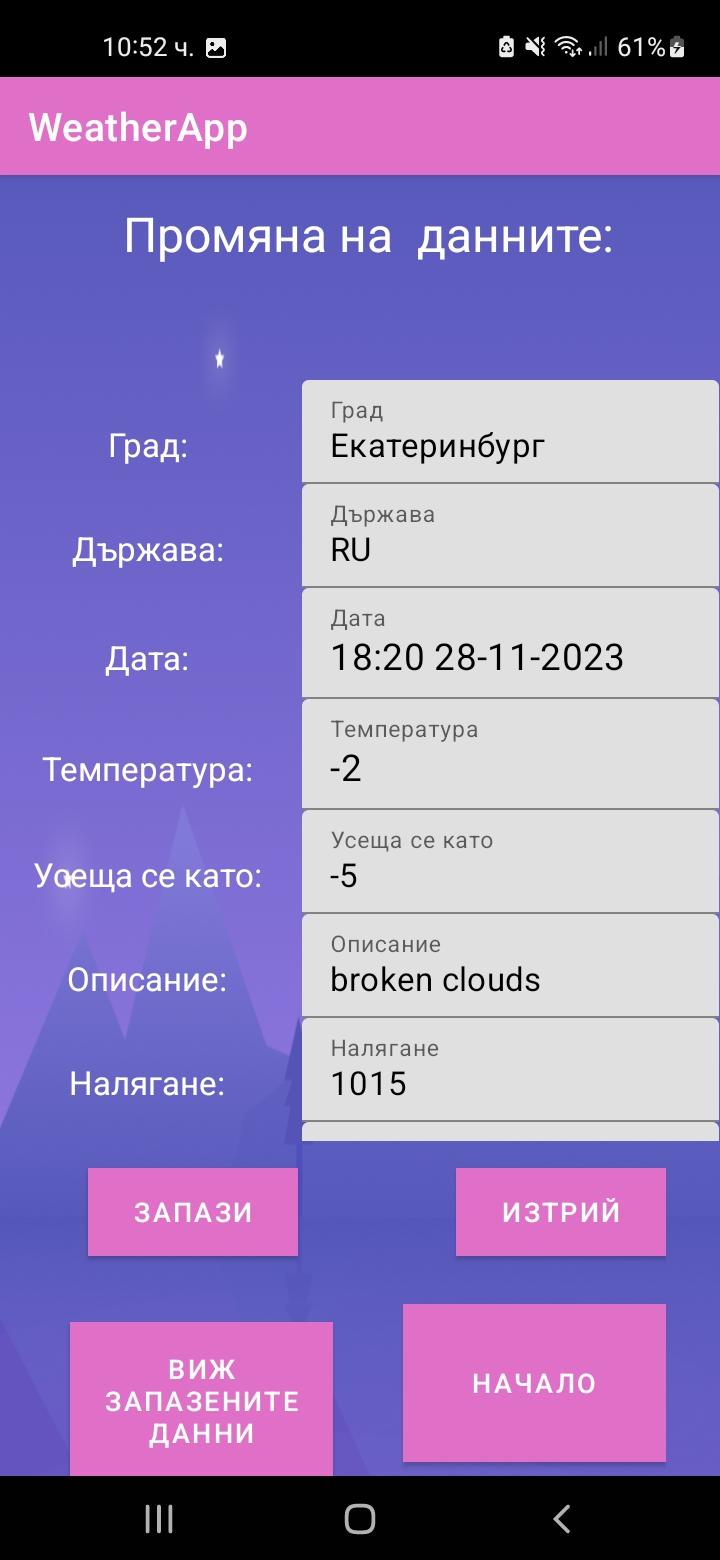
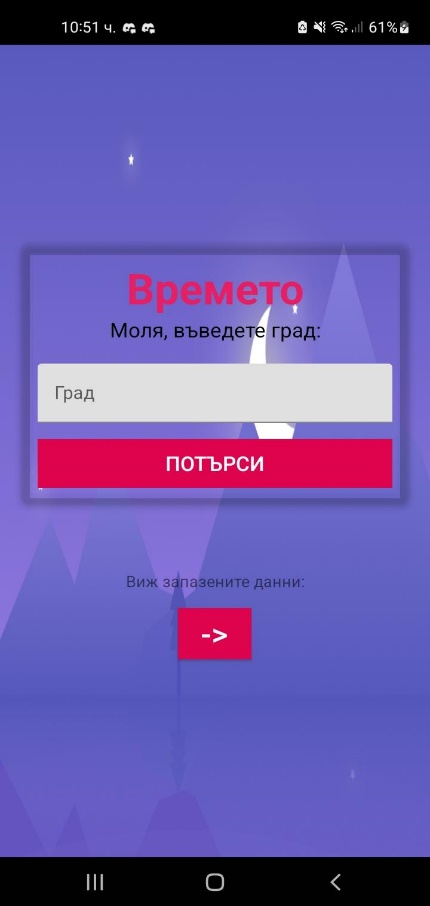
● Kotlin

● Android SDK

● GSON – За десериализация на обекти

● SQLite - базата данни, в която се съхраняват записите

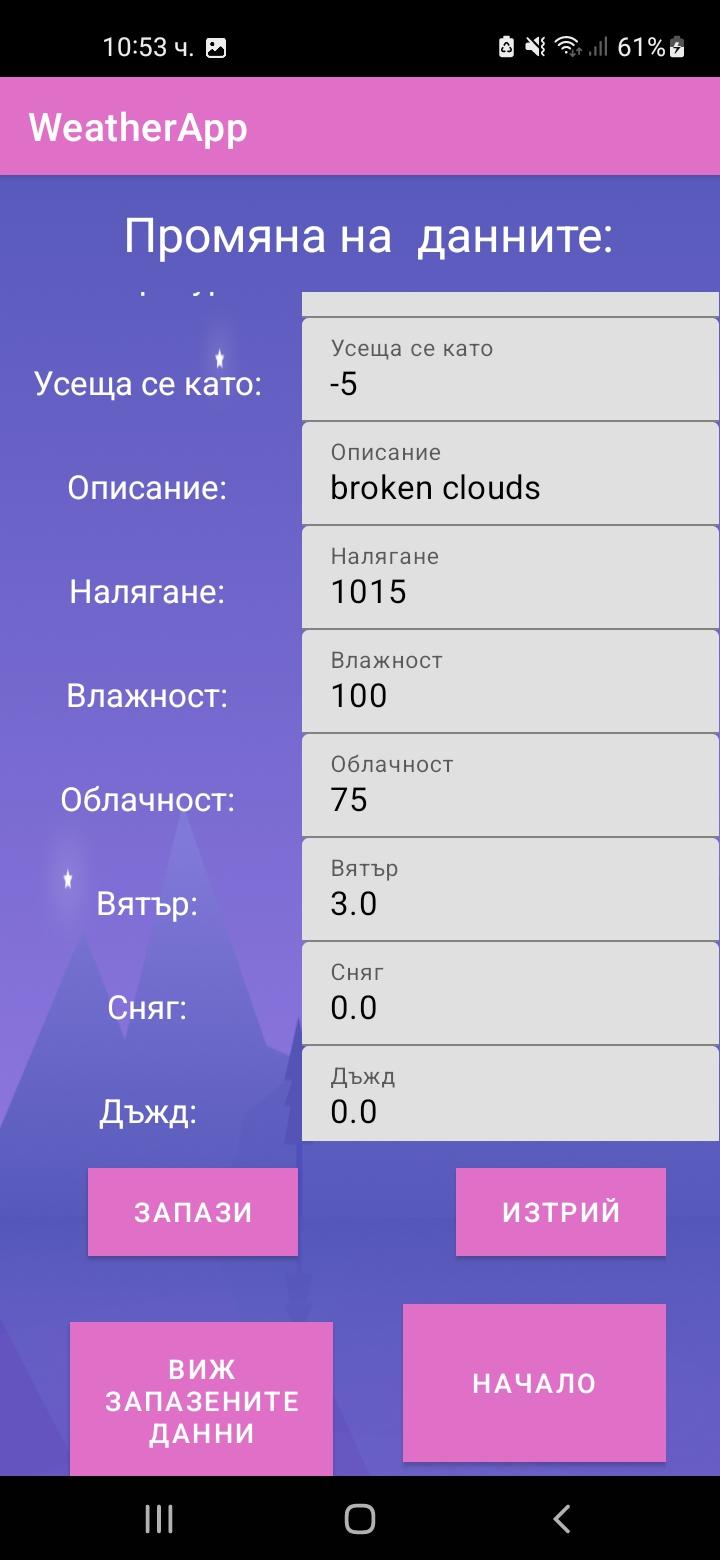
1. ***Потребителски инструктаж***

****

[c]

[b]

[a]



[d]

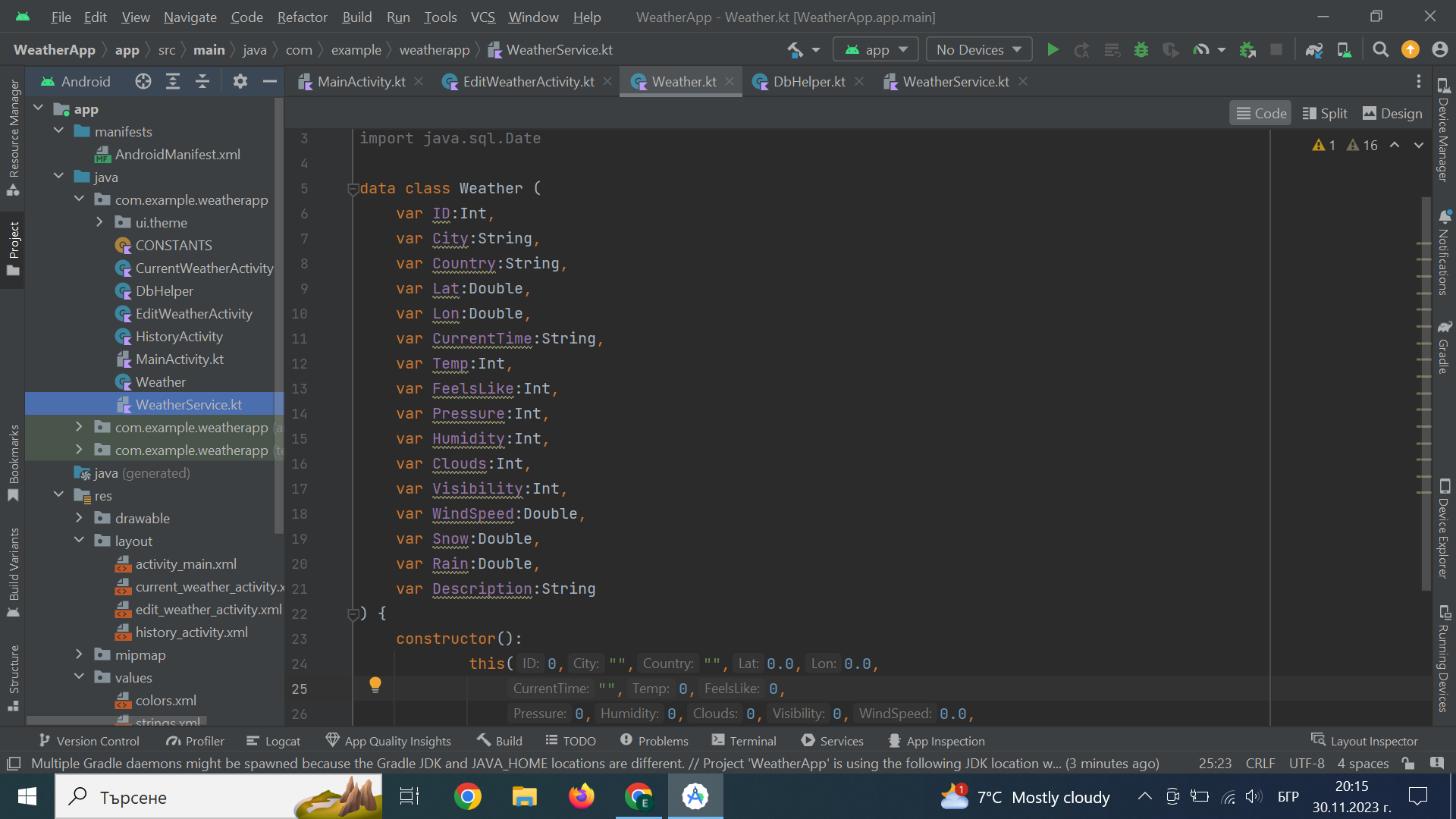
При реализацията на приложението е използван REST API на OpenWeatherMap [1].

След стартиране на приложението за потребителя има две възможности [a]. Едната е да въведе град, за който иска да извлече текуща информация, а другата- да види вече запазените данни [c]. При въвеждане на име на град, след избиране на бутона “Потърси” на потребителя се извежда списък с текущите характеристиките на времето [b]. Данните, които се визуализират са:

* Името на града, заедно с държавата, в която се намира
* Времето (дата и час) на извличане на данните
* Изображение, което съответства на описанието ( в зависимост от това дали в описанието е описано, че вали дъжд, сняг има облаци или е слънчево пред потребителя се визуализира различно изображение)
* Текущата температура в градус Целзий
* Усеща се като
* Описание
* Налягане, измерено в hPa
* Влажност, измерена в %
* Облачност- в %
* Вятър в m/s
* Сняг в mm.
* Дъжд в mm.

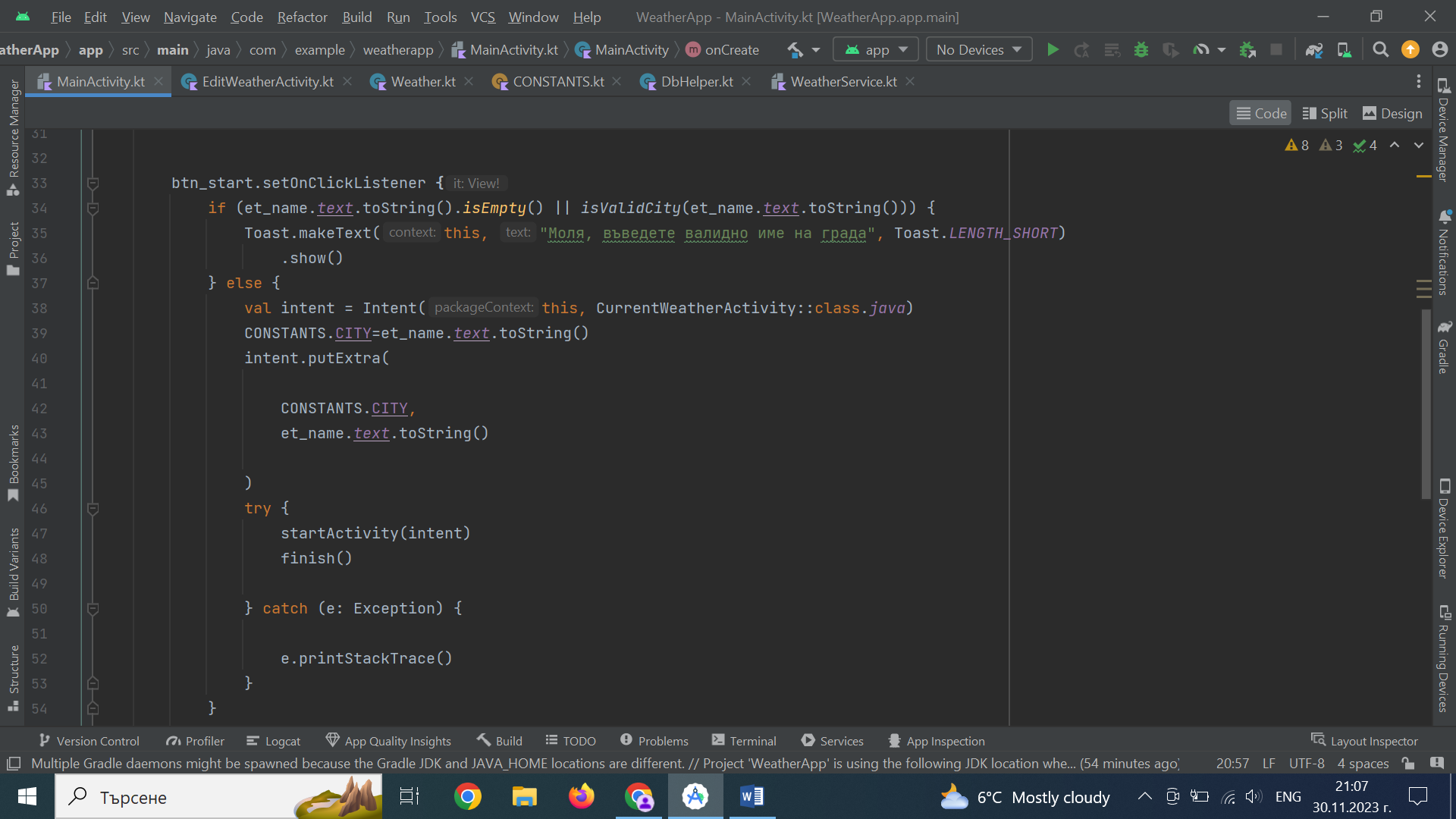
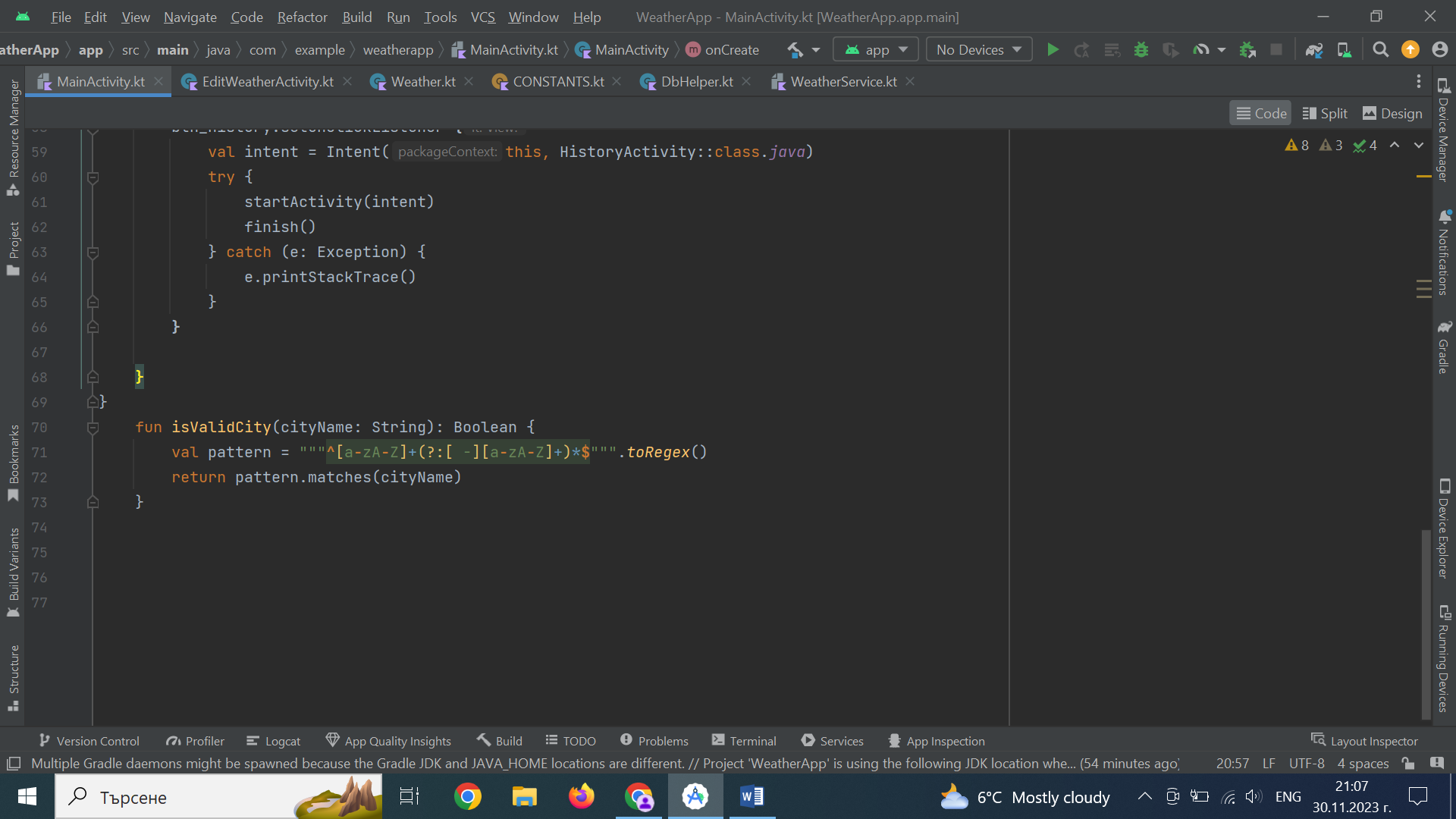
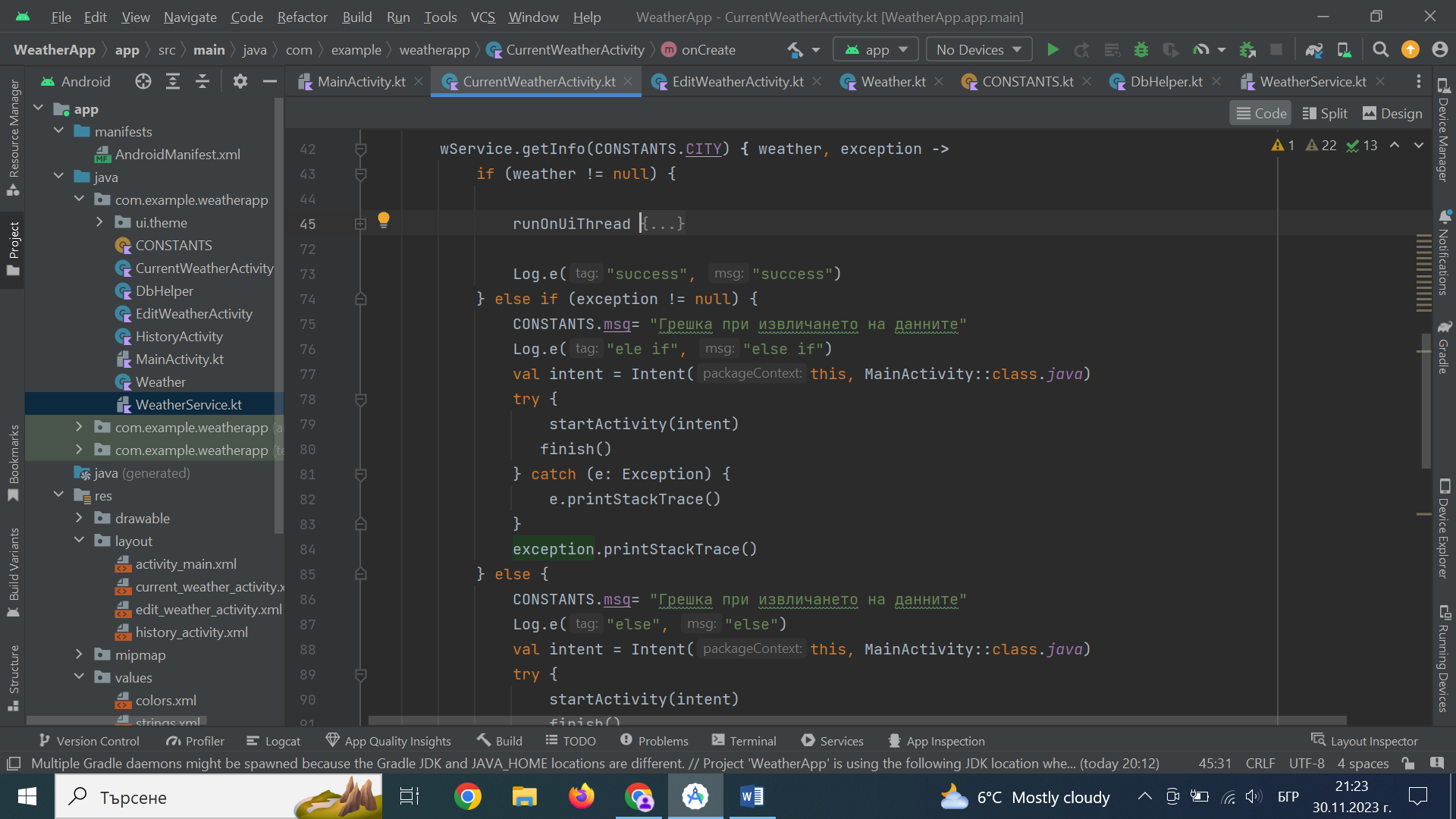
Тук пред потребителя има няколко възможни варианта- да запази информацията посредством бутона “Запази”; да избере бутона “Виж запазените данни”, който от своя страна ще пренасочи потребителя към екран, на който се визуализират записите в БД [c]; или да се върне в началния екран посредством бутона “Начало”, с което той може да потърси информация за друг град.

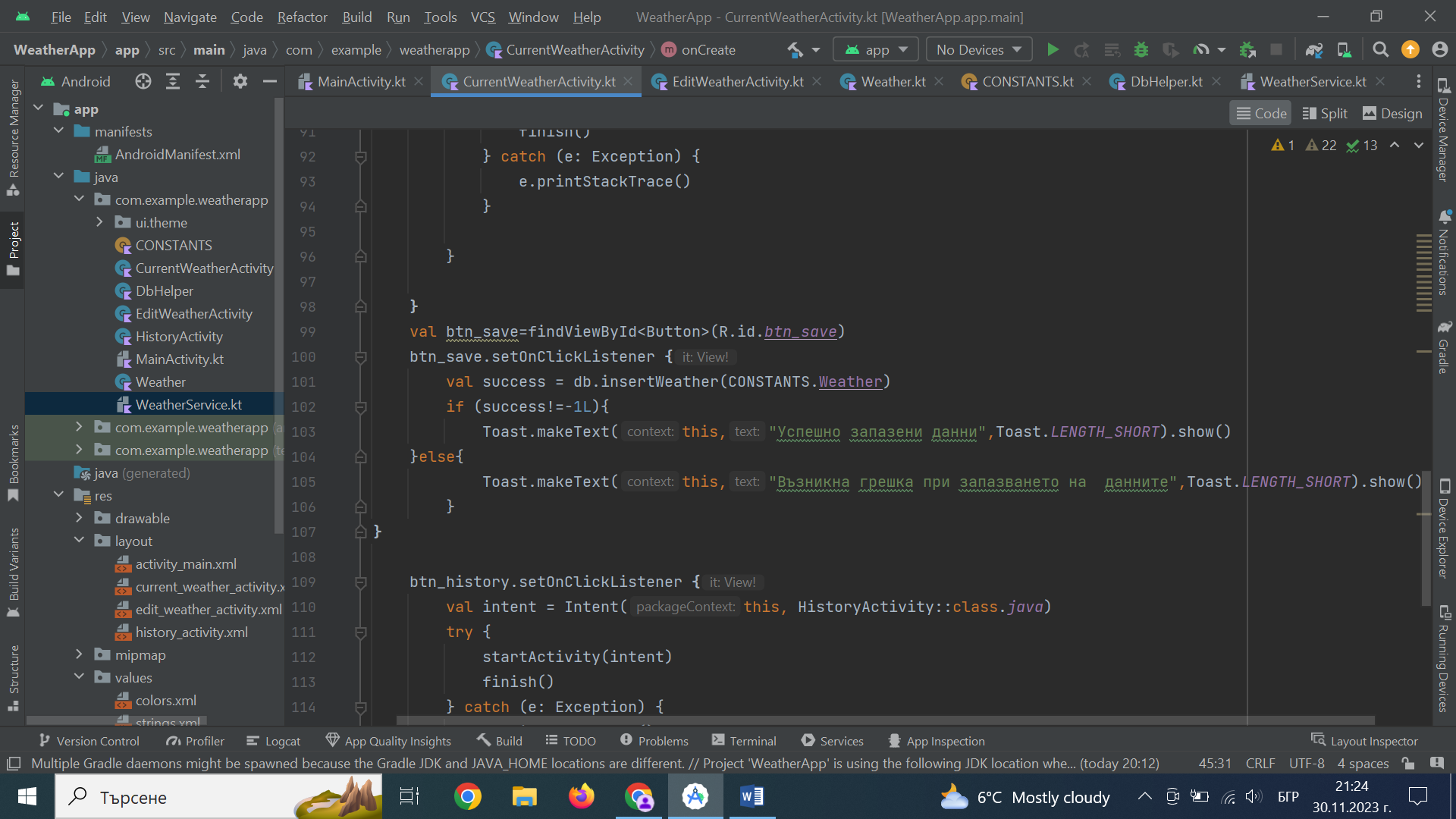
В изгледа със запазените данни пред потребителя се визуализира само част от запазената информация- име на града, дата на извличане на данните, градуси и два възможни бутона за всеки запис- за редакция и за изтриване. Когато потребителя избере бутона за редактиране, пред него се отваря нов изглед [d], който съдържа всички полета, т.е. цялата информация, която се извлича от API. Потребителя може да направи желаните от него промени и да ги съхрани посредством бутона “Запази”, да изтрие записа с бутона “Изтрий”, да се върне една стъпка назад- “Виж запазените данни”[c] или да се върне в самото начало на приложението [a].

1. ***Архитектура на приложението*** 
   * **CONSTANTS**- клас, в който се съхраняват константите на приложението като текущо избран град, id на селектиран запис и други.

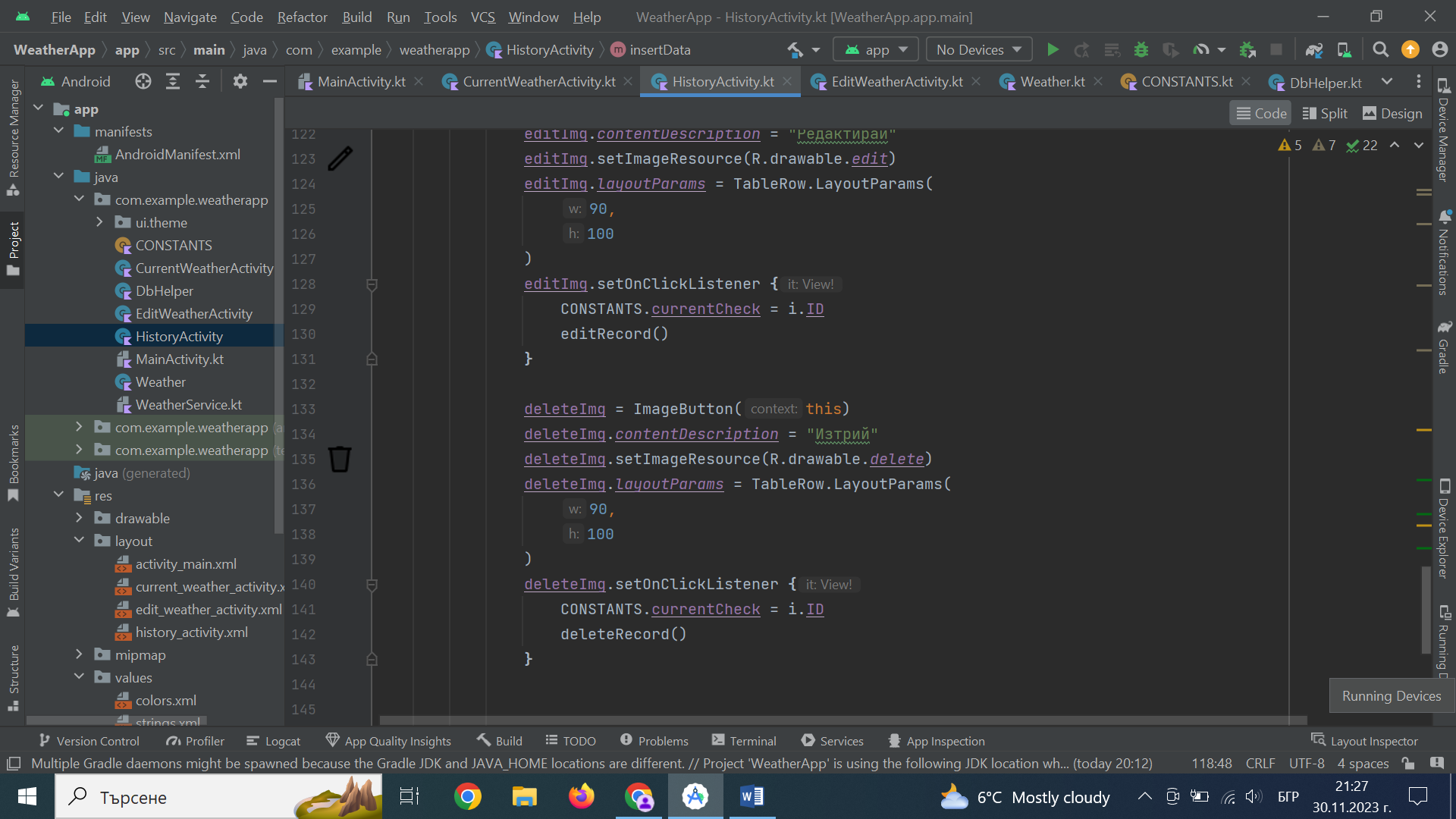
* **CurrentWeatherAactivity**- клас, който реализира логиката за извличането на данните и задава действията, на които отговарят бутоните от изгледа, който показва текущото време.
* **DbHelper**- клас, отговорен за създаването на базата данни, нейната работа и поддържането й.
* **EditWeatherActivity**- клас, който визуализира данните за избран запис, дава възможност на потребителя да ги редактира и осъществява извикването на методите към базата данни за тяхното актуализиране.
* **HistoryActivity**- клас, който визуализира записаните в БД записи. Предлага няколко възможни действия.
* **MainActivity**- реализира функционалността на началния екран.
* **Weather**- клас- модел, който използваме за да окажем необходимите ни полета
* **WeatherService**- класа, чрез който се извличат данните от API

1. ***Имплементация***

* След като потребителя е въвел името на града и е избрал „Потърси“ се извършва проверка дали наистина е въвел нещо в полето и дали въведеният низ е коректен.
* След като всичко е минало успешно. Извикваме функцията за извличане на данни от OpenWeatherMap API, където подаваме името, което потребителя е въвел. Ако всичко е минало успешно, визуализираме получените данни. В противен случай връщаме потребителя в начален екран и извеждаме съобщение.
* Потребителят избира “Запиши“, където с помощта на клас DbHelper се опитваме да запазим данните в SQLite базата. Извеждаме съобщение дали всичко е минало успешно.



* При избор на редакция извличаме данните за избрания запис. Това става като на всеки бутон за редакция и за изтриване имаме setOnClickListener метод, чиято роля е да запази id- то в константите, откъдето можем да видим кой точно запис иска да модифицира потребителя и съответно да извикаме по още един- помощен метод, съответно за изтриване или редактиране на данните.



1. **Заключение**

Курсовият проект дава добра основа за разработване на мобилни приложения с помощта на Kotlin и Android Studio, полезни за бъдещо развитие.

Приложението предоставя интуитивен и лесен за интерфейс, с помощта на който потребителите лесно могат да получат необходимата за тях метеорологична информация.

1. ***Библиография***

[1] „OpenWeatherMap API,“ [Онлайн]. Available: https://openweathermap.org.

1. ***Списък на фигурите***

[a]- Начален екран на приложението

[b]- Екран с текущите характеристики на времето

[c]- Екран, визуализиращ всички записи от базата данни

[d]- Екран за редакция на запис