**ИКОНОМИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА**

**КАТЕДРА „ИНФОРМАТИКА“**



**КУРСОВ ПРОЕКТ**

ПО  
Нейтив мобилни приложения

НА ТЕМА

Мобилно приложение за инспекция на обекти

Разработил: Проверили:

Силви Райчева , Ф.Н.125199 гл. ас. д-р Бонимир Пенчев

Специалност: Мобилни и уеб технологии гл. ас. д-р Стойчо Стоев

Курс: 6, група 42

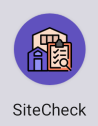
**ВАРНА, 2024**

# Идея на приложението

"SiteCheck" е мобилно приложение, което помага за дигитализацията на документи, изготвени по време на различни видове инспекции на територията на даден обект - изправност на електрически инсталации, проверка на пожарогасители, състояние на вентилационни системи и др. Потребителят е лицето, което извършва проверката. Техникът има възможност да разгледа списък с обекти(прочетени от файл), които трябва да посети за деня. След като пристигне на обект и извърши инспекция, потребителят може да добави информацията за проверката - адрес, помещения и резултат (успех/има нужда от подобрения), след което да изпрати на прекия си ръководител доклада по имейл. В случай, че инспекторът не е запознат с локацията, която трябва да посети, може да я потърси в модул Карти, част от приложението.

# Описание на приложението

## Икона и име на приложението

За приложението е избрана икона(фиг. 1) от Flaticons, а за име е избрано “SiteCheck” (фиг. 2)

Фигура 1. Икона на приложението

## 

Фигура 2. Име на приложението

## Навигация

Навигацията в приложението е осъществена чрез меню и BottomNavigationView и е разположена в MainActivity.



Фигура 3. BottomNavigationView

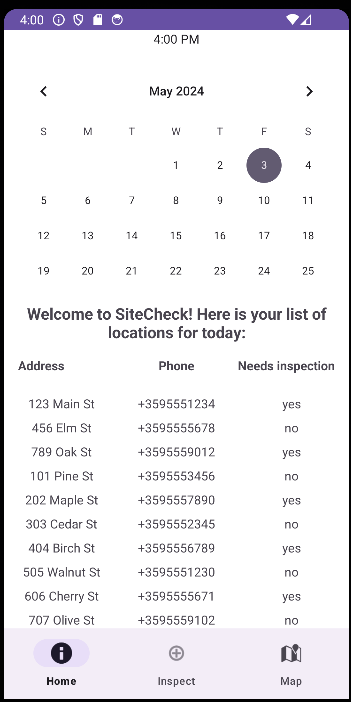
1. Екрани

Приложението се състои от Main Activity и три фрагмента (Home, Inspect, Map). Смяната между тях се прави чрез контрол BottomNavigationView(фиг. 4).



Фигура 4. Програмен код в Main Activity за осъществяване на навигацията в приложението

* 1. Екран “Home”

При стартиране на приложението се зарежда фрагмент Home(фиг. 5). Състои се от няколко вида контроли:

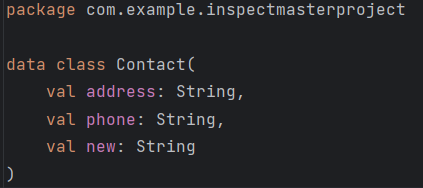
Фигура 5. Екран Home

* TextClock: визуализира времето към текущия момент
* CalendarView: визуализира текущата дата
* TextView: използвани са няколко елемента от този тип за поздравително съобщение към потребителя, както и за оформяне на заглавията на колоните на таблицата
* RecyclerView: визуализира данни, прочетени от файл в папка Assets, “data.txt”

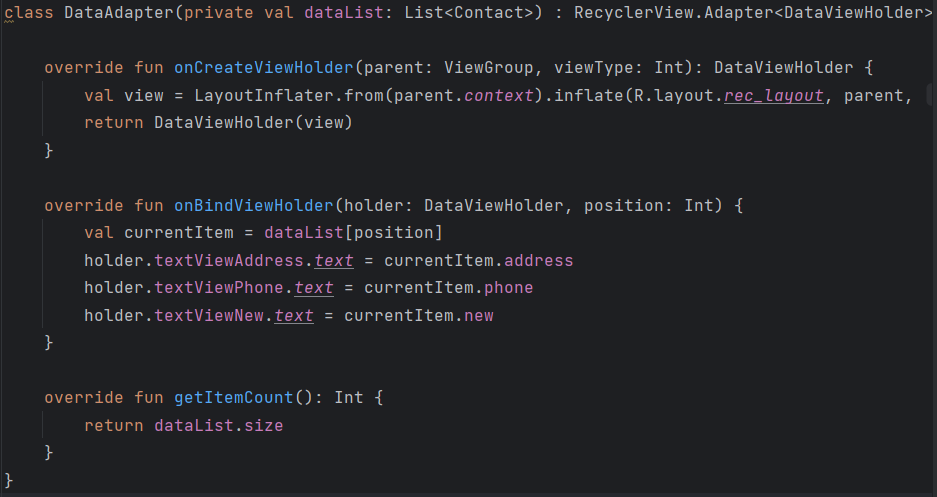
За визуализиране на данните от файла в RecyclerView, се използва Data Class Contact(фиг. 6) и клас DataAdapter(фиг.7). Визуализацията на

записите в RecyclerView се определя от rec\_layout.xml.

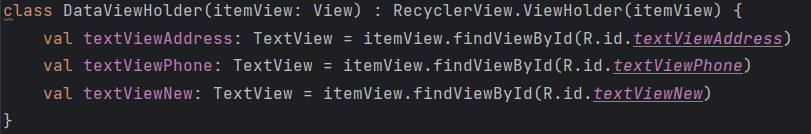
Функцията DataViewHolder(фиг. 8) обвързва елементите от RecyclerView с елементите в класа.



Фигура 6. Data Class Contact

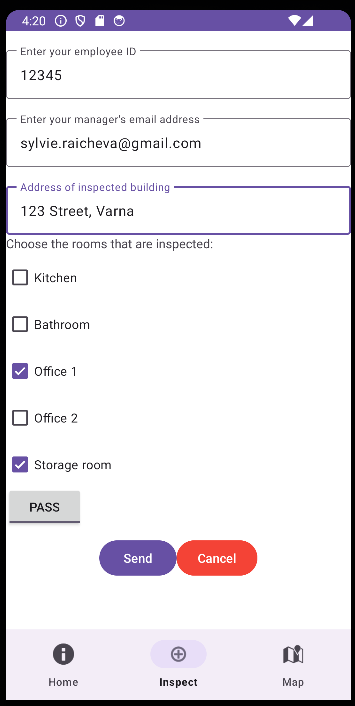


Фигура 7. Клас DataAdapter



Фигура 8.Функция за обвързване на елементите от RecyclerView с елементите в класа

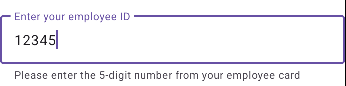
* 1. Екран “Inspect”

Фрагментът Inspect(фиг. 9) позволява на потребителя да въведе данните за инспекцията и да ги запази.

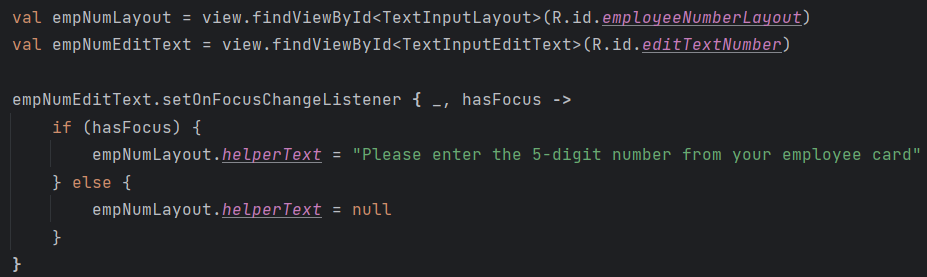
Фигура 9. Екран Inspect

Състои се от няколко вида контроли:

* За трите текстови кутии е използван TextInputLayout, в който е поставен TextInputEditText. Първата текстова кутия приема като вход номера на служителя, като за целта е използван input type number, за да се избегне въвеждането на символи различни от числа. Максималната дължина на входа е 5 символа. При фокусиране върху елемента, се показва помощен текст, като функционалността е реализирана чрез onFocus събитие.

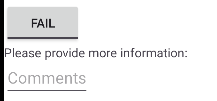
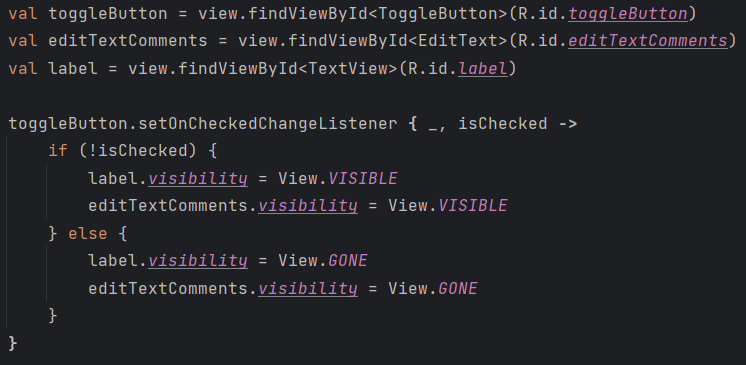


Фигура 10. onFocus събитие



Фигура 11. Реализация на onFocus

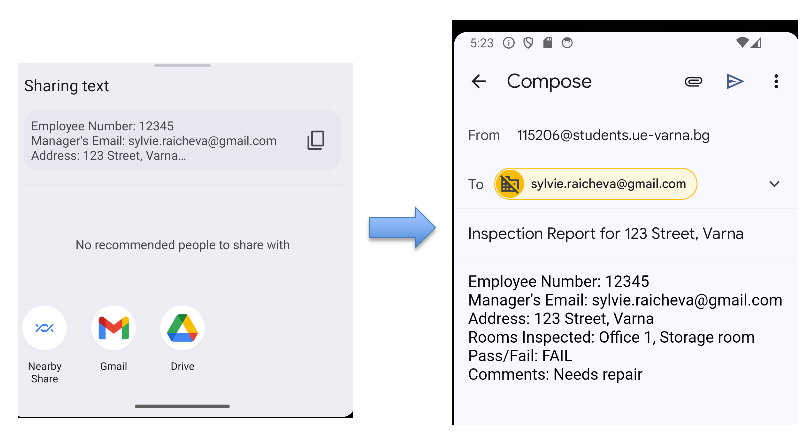
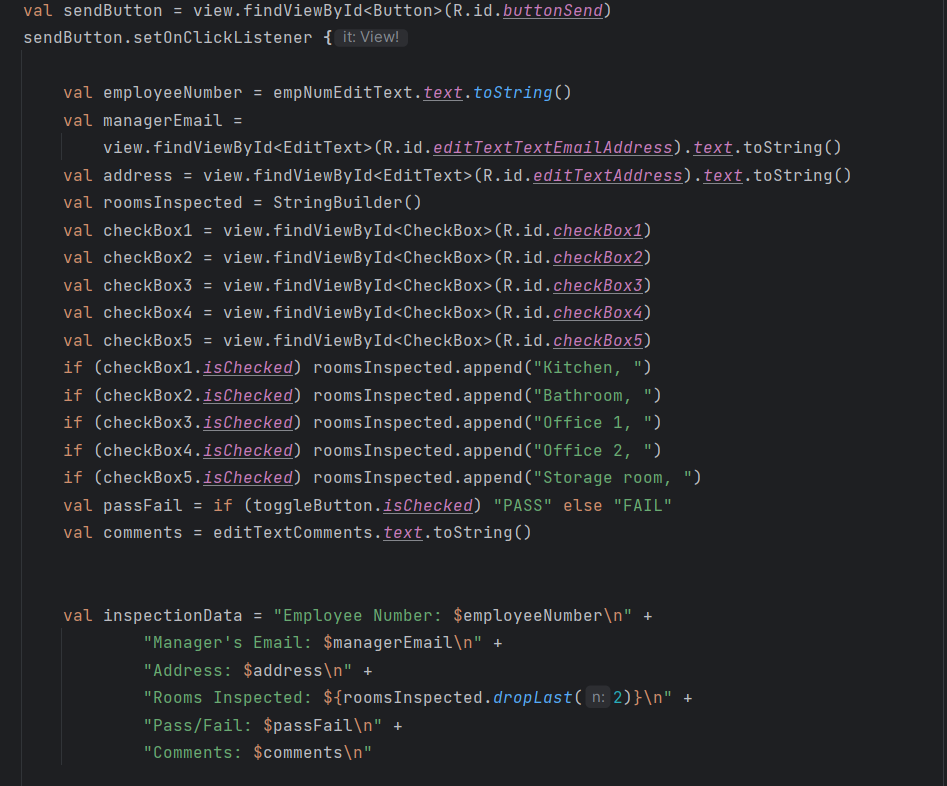
Втората текстова кутия приема като вход имейла на мениджъра на потребителя. За inputType е посочен textEmailAddress. Третата текстова кутия приема като вход адреса на обекта. За inputType е посочен text.

* За избор на инспектираните помещения са използвани пет CheckBox контрола
* За визуализиране на избора от проверката се използва ToggleButton. В случай, че резултатът е FAIL се появява контрол EditText, в който потребителят може да добави допълнителни коментари, като например причина за отказа. Функционалността е реализирана чрез OnCheckedChange събитие.

Фигура 12. OnCheckedChange събитие

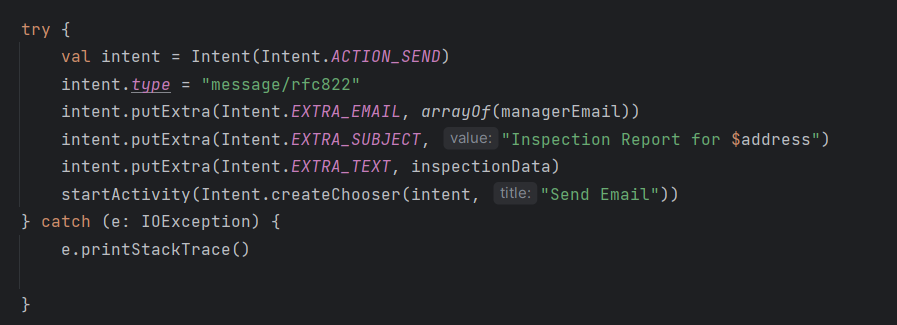
Фигура 13. Реализация на OnCheckedChange

* При клик върху бутон Send(onClick събитие) се прихваща информацията попълнена в полетата и се запазва в променливата inspectionData. Чрез Intent се стартира имейл клиент по избор на потребителя, който отваря имейл адресиран към имейла на мениджъра и съдържание, съхраненото в inspectionData.

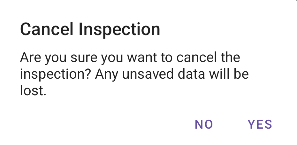


Фигура 14. Прихващане на въведените данни

Фигура 15. Стартиране на Gmail чрез Intent



Фигура 16. Intent реализация

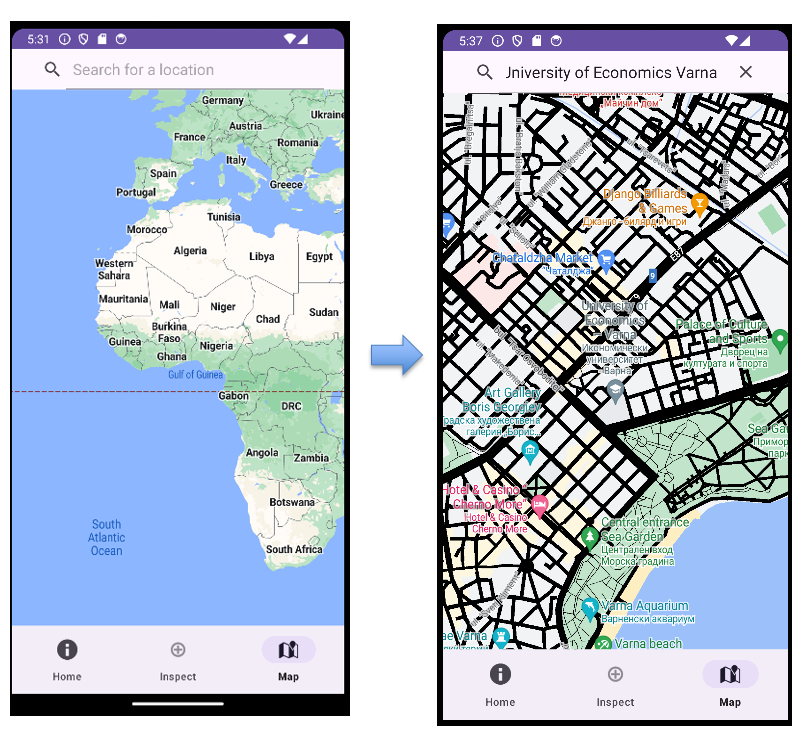
* При клик върху бутон Cancel(onClick събитие), се визуализира диалогов прозорец с два избора – да се потвърди излизането от страницата или да се откаже, при което потребителят да остане в същата страница без да се изтрият попълнените данни.

Фигура 17. Програмен код на диалогов прозорец

Фигура 18. Диалогов прозорец с два избора

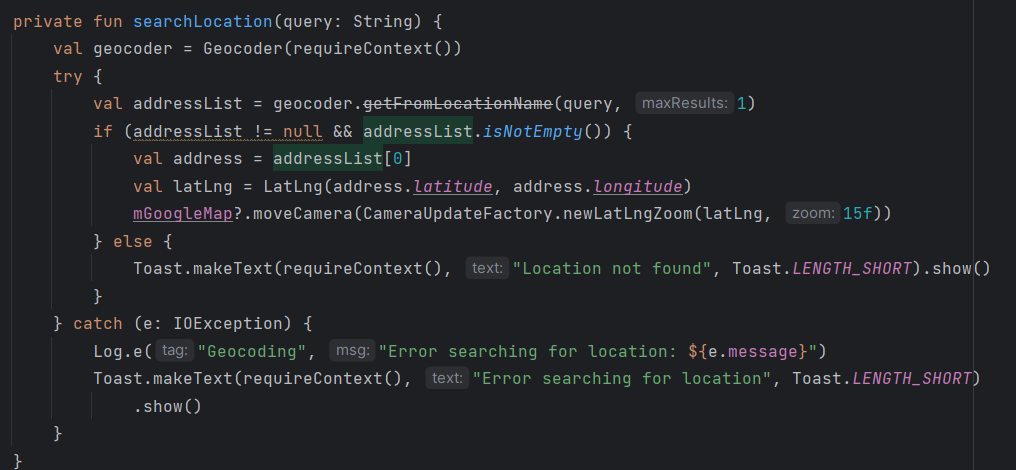
* 1. Екран “Map”

Фрагментът Map дава възможност на потребителя да потърси определена локация, която да се визуализира на картата. Функционалността е реализирана чрез Google Maps API. Използвани са SearchView и Fragment контроли, като потребителят въвежда локацията в SearchView и тя се визуализира на картата (във фрагмента).



Фигура 19. Екран Map преди и след търсене на локация

Функцията searchLocation използва класа Geocoder, за да превърне името на локацията въведена от потребителя в географски координати (ширина и дължина). След това, измества камерата на получените координати на картата като по този начин се задава фокус на локацията.



Фигура 20. Търсене на локация