**Спецификација**

за изработка на проектна задача

**Вовед**

Потребно е да се изработи Андроид апликација чија основна Андроид компонента е сервис (без GUI и интеракција со корисникот). Сервисот ќе се вклучи при иницијалното инсталирање на апликацијата, но и при бутирање на телефонот и истиот ќе биде резистентен на исклучување од корисникот и/или оперативниот систем. Основна задача на сервисот е да прави тестирање на квалитетот на мрежата, без било какво инволвирање на корисникот. Конкретната задача што треба да ја извршува апликацијата треба да ја побара и добие од backend (поточно од web API). Одговорот што ќе го добие од API-то е во форма на JSON кој го дефинира типот на задачата (JOB-от), па пример тип на JOB може да биде PING, како и дополнителни параметри, на пример на која IP адреса да прави ping, на колкав периодичен интервал, инт. Резултатите од задачата, апликацијата треба да ги испрати до backend-от, при што треба истите да содржат и timestamp од моментот на тестирањето. Ако во моментот на тестирањето нема мрежа, апликацијата треба да е способна да ги зачува неиспратените податоци до моментот кога ќе има мрежна конекција со backend-от, но и да има контрола на количеството податоци што ги чува за да истото не нарасне премногу големо.

**Начин и динамика на изработка на проектната задача**

Проектот ќе се работи индивидуално. Секој треба да креира и развива своја верзија. Проектот е задолжителен за сите студенти и ќе се работи во фази (milestones) во текот на неколку недели. За секоја од фазите ќе биде понуден патоказ и можни проблеми, како и принципи за нивно надминување. **Студентите може да понудат свои различни решенија кои се компатибилни со спецификациите дефинирани во овој документ**. Предвидени се следните генерални фази на изработка:

Фаза 1: да се креира апликација која ќе биде отпорна на исклучување и која ќе е способна да се подигне при вклучување на уредот. Дополнително да се осигура дека апликацијата нема да покрева графичка активност ниту при нејзино инсталирање, ниту при вклучување на уредот.

Фаза 2: да се оспособи апликацијата да комуницира со backend-от со цел да превземе job. На почетокот ќе дефинираме само еден тип на job, а тоа е jobType: PING со соодветни параметри, но апликацијата треба да има потенцијал и способности да се проширува (модуларно) со други типови на job-ови. Во соодветниот модул, апликацијата треба да почне да го извршува job-от без да го блокира работењето на уредот (позадински).

Фаза 3: да се оспособи апликацијата да ги испраќа резултатите од тестирањето до backend-от. Ако не постои конекција, истите да може да ги зачува и да ги испрати кога ќе има конекција. Да може да го контролира количеството на зачувани податоци до одредена горна граница чувајќи ги само најсвежите записи. Записите да се отпорни за ребутирање на уредот.

Фаза 4: проширување на множеството JOB-типови.

**Фаза 1**

Со цел да направите апликација која е отпорна на исклучување и прекин на работата, било заради дејствија на корисникот или заради дејствија на Андроид оперативниот систем (кој ги исклучува апликациите и сервисите според одредени правила и ситуации – најчесто кога нема ресурси како на пример RAM), препорачуваме да ја проучите апликацијата опишана на следниот линк:

<https://fabcirablog.weebly.com/blog/creating-a-never-ending-background-service-in-android>

особено подобрената варијанта од следниот линк:

<https://fabcirablog.weebly.com/blog/creating-a-never-ending-background-service-in-android-gt-7>

соодветните кодови за двете варијанти може да се симнат од следните линкови:

<https://github.com/fabcira/neverEndingAndroidService>

и

<https://github.com/fabcira/neverEndingProcessAndroid7->

Во првата недела од проектот, потребно е да се симне кодот и да се инсталира. Треба детално да се проучат механизмите кои се употребени во ова апликација од аспект на преживливост на процесот и од аспект на остварување на оваа задача кај различните верзии за употребеното Android API. За оваа цел, дополнете го лог-от кој го генерира апликацијата и темелно тестирајте различни сценарија. Пробајте го кодот на различни верзии на оперативниот систем и изберете една работна околина на која ќе работите до крајот на работилницата.

Вториот чекор на оваа фаза побарува отстранување на GUI-то. Овој чекор не е круцијален за крајната цел на проектот, но е дополнителен предизвик што се јавува како потреба во праксата. За оваа цел може да најдете многу различни решенија. На пример едно такво решение е дискутирано на следниот линк <https://stackoverflow.com/questions/10909683/launch-android-application-without-main-activity-and-start-service-on-launching>

За конкретниот пример кој го разгледуваме, едно можно решение е следното:

1) во манифестот направете промена на темата на активноста во **android:theme="@android:style/Theme.NoDisplay"**

2) во MainActivity направете промена во дефиницијата на класата во следниот облик

**public class** MainActivity **extends** Activity {

3) во MainActivity тргнете ја линијата (коментирајте ја) во onCreate методот

*// setContentView(R.layout.activity\_main);*

4) во MainActivity повикајте finish()

Тоа може да се направи на крајот од onCreate() или на крајот од onResume(). Воочете ги разликите. Тестирајте.

Постојат и други опции и други решенија за апликација без GUI. Слободно пробајте различни варијанти.

Откога ќе го завршите овој дел потребно е да ни го испратите вашиот проект (или линк од github) на [pero@feit.ukim.edu.mk](mailto:pero@feit.ukim.edu.mk) и на [danield@feit.ukim.edu.mk](mailto:danield@feit.ukim.edu.mk) , заедно со објаснување која варијанта сте ја употребиле и што е специфично во вашето решение.

Рок за изработка на оваа фаза е две недели **(до 1.5.2022-23:59)**.

**Фаза 2**

Направете backup на вашата верзија од Фаза 1. Сега потребно е апликацијата да се надгради со цел да комуницира со backend-от и да превземе job. Повикувањето на backend-от треба да се направи при активирање на апликацијата, но и периодично на фиксна периода од 10 минути. Пред да се повика backend-от, треба да се провери дали има мрежна конективност.

За тестирање направивме едноставен ASP.NET Core backend кој ќе го активирате на вашиот компјутер. Backend-от може да го симнете од <https://e-kursevi.feit.ukim.edu.mk/mod/resource/view.php?id=9362> и потребно е да го распакувате на вашиот компјутер. Го активирате со двоклик на EXE-то Backend\_v1.exe. Backend-от активира end-point кој треба да го повикате со вашата андроид апликација. Истиот може да го проверите преку вашиот пребарувач на <http://localhost:5000/getjobs>

Одговорот што ќе го добиете треба да биде JSONArray од дефиниции за job-овите кои треба да ги извршува вашиот андроид код. Во овој момент постои само една дефиниција од следниот облик:

**[**

**{**

**"date": "2020-04-18T17:33:16.1768632+02:00",**

**"host": "10.0.2.2",**

**"count": 3,**

**"packetSize": 100,**

**"jobPeriod": 120,**

**"jobType": "PING"**

**}**

**]**

Како што може да се види, станува збор за job-типот PING, што значи апликацијата треба да извршува наредба PING кон host-от дефиниран во JSON-от (10.0.2.2), на периодичен интервал од 120 секунди со користење на 3 пакети со големина 100 бајти.

Неколку практични совети за оваа фаза:

* На соодветно место во вашиот код може да направите бесконечен циклус кој ќе одговара на превземањето на job на секои 10 минути, при што во меѓувреме може тредот да го ставите да спие.
* Мрежната конективност може да ја проверите на пример со користење на ConnectivityManager.
* Отворањето на мрежна конективност треба да се направи со позадински тред (на пример со AsyncTask).
* Одговорот од backend-от парсирајте го со користење на JSONArray и JSONObject класите.
* Наредбата PING во андроид може да ја извршите преку Runtime.getRuntime().exec   
  еве линк за еден пример <https://www.codeproject.com/Questions/1098186/How-to-get-unix-timestamp-while-pinging-in-android>
* Опциите на PING се повеќе на број и се дефинирани од линукс кернелот. Еве линк за опциите на користење: <https://www.geeksforgeeks.org/ping-command-in-linux-with-examples/>
* Повикувањето на backend-от кај андроид емулаторот треба да оди преку loopback адресата 10.0.2.2 (еве објаснување <https://developer.android.com/studio/run/emulator-networking> и <https://stackoverflow.com/questions/28590366/ioexception-java-net-connectexception-failed-to-connect-to-127-0-0-1port-5000> )
* Резултатот од PING-от засега испишете го во лог-от.
* Кодот направете го што е можно по-модуларен, со цел да е спремен да прими и изврши повеќе различни дефиниции во JSONArray-от што го добивате како одговор од backend-от.