

University for Business and Technology in Kosovo

UBT Knowledge Center

Theses and Dissertations

Student Work

Winter 2-2021

ZHVILLIMI I APLIKACIONIT PËR PËRCJELLJEN E MOTIT I BAZUAR NË SISTEMIN OPERATIV ANDROID

Egzon Baftiu

University for Business and Technology - UBT

Follow this and additional works at: <https://knowledgecenter.ubt-uni.net/etd>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Baftiu, Egzon, "ZHVILLIMI I APLIKACIONIT PËR PËRCJELLJEN E MOTIT I BAZUAR NË SISTEMIN OPERATIV ANDROID" (2021). *Theses and Dissertations*. 2147.

<https://knowledgecenter.ubt-uni.net/etd/2147>

This Thesis is brought to you for free and open access by the Student Work at UBT Knowledge Center. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of UBT Knowledge Center. For more information, please contact knowledge.center@ubt-uni.net.



Programi për Shkenca Kompjuterike dhe Inxhinierise

**ZHVILLIMI I APLIKACIONIT PËR PËRCJELLJEN E MOTIT I
BAZUAR NË SISTEMIN OPERATIV ANDROID**

Shkalla Bachelor

Egzon Baftiu

Shkurt / 2021
Prishtinë



Programi për Shkenca Kompjuterike dhe Inxhinierise

Punim Diplome
Viti akademik 2014 – 2015

Egzon Baftiu

**ZHVILLIMI I APLIKACIONIT PËR PËRCJELLJEN E MOTIT I
BAZUAR NË SISTEMIN OPERATIV ANDROID**

Mentori: Can.PhD. Xhelal Jashari

Shkurt / 2021

Ky punim është përpiluar dhe dorëzuar në përmbushjen e kërkesave të
pjesëshme për Shkallën Bachelor

ABSTRAKTI

Aplikacionet softuerike në përgjithësi ndërtohen për të ju përshtatur kërkesave të klientit në mënyrë që të ju mundësohet një menaxhim më i lehtë i punëve mbi të cilat veprojnë. Një nga format e zhvillimit të aplikacioneve është edhe zhvillimi i aplikacioneve mobile të bazuara në android. Në këtë temë do të krijojmë një aplikacion mobil të bazuar në sistemin operativ android që për qëllim ka shfaqjen e informatave rreth motit, ky lloj aplikacioni është i nevojshëm dhe i rëndësishëm për përdoruesit që dëshirojnë ta shikojnë temperaturën në vende të ndryshme të botës dhe posaçërisht në Kosovë. Aplikacioni ofron mundësi të shumta si: temperatura maksimale, temperatura minimale, lagështinë, dhe presionin e ajrit. Aplikacioni i ndërtuar do të ketë të dhëna reale dhe të sakta, kjo do të bëhet duke e krahasuar rezultatet me aplikacione tjera të ngjashme. Një aplikacion i tillë është i nevojshëm për të gjithë sidomos për ata persona të cilët puna e tyre e përditshme është e varur nga moti apo edhe nga ata që vendosin për ti kaluar pushimet si verore apo dimërore. Në këtë punim gjithashtu janë bërë krahasime brenda aplikacioneve android për tablet, televizion dhe ora të mençura.

Fjalët kyçe: aplikacionet mobile, aplikacionet android, sistemi i motit, retrofiti, json.

MIRËNJOHJE/FALENDERIME

Do të doja të shpreh mirënjohjen time të sinqertë për të gjithë ata të cilët më ndihmuan që ta përfundojë këtë pjesë të rrugëtimit. Një falënderim i veçantë shkon për mentorin Xhelal Jashari i cili e ka mbikëqyrur këtë projekt. Ai më ofroi orë të panumërta me këshilla, udhëzime dhe mbështetje. Unë e falënderoj atë për angazhimin e tij gjatë gjithë momenteve të lehta dhe të vështira që i kam pasur deri në përfundimin e këtij punimi. Falënderim i njejtë u takon edhe kolegëve të mi nga UBT, si dhe profesorëve për kontributin e tyre në lëndë të ndryshme.

Këtë punim jua kushtoj prindërve të mi babait Naim Baftiu dhe nënës Fetije Baftiu, për mbështetjen e tyre të ngrohtë dhe të pafund.

PËRMBAJTJA

LISTA E FIGURAVE	V
-------------------------------	----------

LISTA E TABELAVE	V
-------------------------------	----------

1 HYRJE.....	Error! Bookmark not defined.1
---------------------	--------------------------------------

1.1 Objektivat e temës.....	2
-----------------------------	---

2 SHQYRTIMI I LITERATURËS	3
--	----------

2.1 Aplikacionet Android.....	3
-------------------------------	---

2.2 Analizimi i aplikacioneve mobile.....	4
---	---

2.3 Avantazhet e aplikacioneve mobile.....	5
--	---

2.4 Disavantazhet e aplikacioneve mobile	6
--	---

2.5 Aplikacioni në bazë të motit.....	8
---------------------------------------	---

2.6 Krahasimi i aplikacioneve android.....	9
--	---

3 DEKLARIMI I PROBLEMIT	11
--------------------------------------	-----------

4 METODOLOGJIA.....	12
----------------------------	-----------

5 REZULTATET.....	13
--------------------------	-----------

5.1 Zhvillimi dhe analizimi i projektit.....	13
--	----

5.2 Android Studio.....	14
-------------------------	----

5.2.1 Android Manifesti.....	15
------------------------------	----

5.2.2 Implementimi i Gradle.....	16
----------------------------------	----

5.2.3 Application programming interface.....	16
--	----

5.3 Veglat e përdorura.....	17
-----------------------------	----

5.3.1 Web faqja.....	17
----------------------	----

5.3.2 Java.....	17
-----------------	----

5.3.3 Json.....	18
-----------------	----

5.3.4 Retrofit.....	18
---------------------	----

5.3.5 Api key.....	18
--------------------	----

5.4 Zhvillimi i aplikacionit dhe ndarja e klasave.....	18
--	----

5.4.1 Main Activity.....	19
--------------------------	----

5.4.2	Api interface.....	19
5.4.3	Maps Activity.....	19
5.4.4	Coordinates.....	19
5.4.5	Weather Response.....	20
5.4.6	Weather.....	20
5.4.7	Gjenerimi i keys.....	20
5.5	Menaxhimi i riskut.....	21
5.5.1	Gabimi në llogaritjen e kohës së implementimit.....	22
5.5.2	Mungesa e njohurive në teknologji të caktuara.....	22
5.5.3	Dështimi i gjetjës së testueve potencial.....	22
5.6	Siguria dhe Privatësia.....	23
5.6.1	Kodi i përdorur për komunikimin me serverin.....	23
5.6.2	Implementimi i picassos.....	24
5.6.3	Kodi për marrjen e retrofitit.....	25
5.6.4	Kodi për marrjen e inputit të vendit.....	25
5.7	Rezultatet përfundimtare.....	26
6	DISKUTIME DHE PËRFUNDIME.....	27
7	REFERENCAT.....	28

LISTA E FIGURAVE

Figura 1. Historia e Aplikacioneve Android	4
Figura 2. Paraqitja grafike e zhvillimit të aplikacioneve.....	5
Figura 3. Mobile Application.....	8
Figura 4. Aplikacionet Android në device të ndryshëm.....	15
Figura 5. Paraqitja e layoutit në baze të kodit.....	25
Figure 6. Paraqitja e implementimit të picassos në layout	25
Figura 7. Pjesa e layoutit në bazë të kodit ku pranon input.....	26
Figura 8. Paraqitja e vendit të kërkuar.....	27

LISTA E TABELAVE

Tabela 1. Krahasimi i aplikacioneve.....	9
--	---

1 HYRJE

Aplikacionet mobile kanë gjetur zbatim të madh në tregun e punës e sidomos në ato bizneset ku aplikacionet e tyre janë kryesisht të fokusuara në zhvillimin e aplikacioneve mobile. Sot në botë ekzistojnë aplikacione të ndryshme varësisht nga kërkesat e klientit të cilit i nevojiten, rëndësia e zhvillimit të aplikacioneve mobile për biznesin është e dukshme sot, pasi gjithnjë e më shumë përdorues kërkojnë të bëjnë gjërat virtualisht. Nga ana tjetër, edhe bizneset po ndryshojnë mënyrën e funksionimit të tyre. Ata i kuptojnë përfitimet e aplikacioneve mobile për bizneset dhe ofrojnë një mënyrë të thjeshtë për përdoruesit e fundit për të hyrë menjëherë në detaje të kompanisë dhe njëkohësisht ti mbajnë të lidhur me markën e tyre të preferuar dhe të qëndrojnë të aranzhuar. Aplikacionet mobile në android mund të zhvillohen në dy gjuhë në java dhe kotlin. Java është gjuhë programuese e cila fillimisht u zhvillua nga James Gosling dhe inxhinier të tjerë të kompanisë Sun Microsystems. Java si komponent thelbësore e Java platformës u lëshua në vitin 1995. Java edhe pse shumicën e sintaksës së saj e ka të derivuar nga gjuhët programuese C dhe C++, ajo në krahasim me këto dy gjuhë përmban një model të objektit më të thjeshtë si dhe ka përkrahje më të vogël për punë të drejtë-për-drejtë me resurset sistemore. Ne do ta zhvillojmë aplikacionin në gjuhën programuese Java pasi që kemi njohuri të arritura nga fakulteti dhe kemi provojë më të madhe në këtë gjuhë prandaj kemi vendosur ta zhvillojmë në këtë gjuhë. Sistemi i shpërndarjes është i njëjti sa i përket mënyrës se si funksionojnë njoftimet, por ka ndryshime në mënyrën se si ato mund të përshtaten. Shumë herë do të dëshironit të ofronit mbështetje dhe azhurnimet përkatëse të saj për përdoruesit. Aplikacionet mobile android në krahasim me aplikacionet mobile në IOS në nivelin e rritjes së përdoruesve për botën e biznesit ka gjetur zbatim ose përdorim më të madh aplikacionet në platformën Android. Për arsye sepse aplikacionet në platformën android janë shumë më use-friendly më të përdorshme. Qëllimi i aplikacionit tonë që do të zhvillohet është krijimi i një aplikacioni që do të shfaqë të dhëna të temperaturës në kohë reale në të gjitha vendet e botës duke pas parasysh komunikim me serverin në Londër ku ka një API open-source për zhvilluesit e aplikacioneve pa pagës. Mënyrën e zhvillimit të aplikacionit si

përdorimin e teknologjisë, veglave për krijim do ta shohim në hapat më tutje që janë të paraqitura në këtë punim diplome.

1.1 Objektivat e temës

Objektivat e kësaj teme janë:

- Të prezantoj një aplikacion mobile në platformën android.
- Të përmbaj të dhëna të sakta në aspektin e motit.
- Të implementoj zgjidhjen e një problemi të caktuar.
- Zhvillimi i aplikacionit me korrektësi dhe real.
- Të jap të dhënat e motit në të gjitha vendet e botës.
- Të shtjelloj të gjithë procesin nga marrja e të dhënave deri te implementimi i projektit.
- Shfrytëzimi i API për ndërtimin e aplikacionit tonë.
- Të jap përfundime se cili është rezultati dhe kush janë palët e interesit.

2 SHQYRTIMI I LITERATURËS

Në këtë kapitull do të analizojmë dhe do ta shqyrtojmë literaturën tonë se çka janë aplikacionet android , rëndësin e përdorimit të tyre si dhe analizimin e një aplikacioni mobile që do të na përcjell të dhënat e motit po ashtu avantazhet dhe disavantazhet e aplikacioneve mobile si dhe krahasimin ndërmjet aplikacioneve tjera që i kemi si television, tablet, orë e mençur.

Gjatë këtij kapitulli gjithashtu do të analizojmë se çka është weather map aplikacioni, komponentët e saj si dhe rëndësia e përdorimit të aplikacionit duke e pasur parasysh edhe benifitet bazë që i ofron ky aplikacion, pas kësaj do të bëjmë analizimin e zhvillimit të aplikacioneve të ngjashme me aplikacionin tonë si dhe çfarë teknologji është përdorur për zhvillimin e këtyre aplikacioneve, si dhe do të tregojmë mënyrën e funksionimit se si këto aplikacione mobile po operojnë në tregun tonë.

2.1 Aplikacionet Android

Aplikacionet mobile android janë të bazuar në sistemin operativ Linux dhe Kernel ku janë krijuar për të ekzekutuar në një pajisje android ose emulator. Termi gjithashtu i referohet një skedari APK që qëndron për paketën Android. Ky skedar është një file Zip që përmban kodin e aplikacionit, burimet dhe informacionin dhe meta informacionin. Aplikacionet Android mund të shkruhen në Kotlin, Java dhe C++ dhe ekzekutohen brenda makinës virtuale. Aplikacionet shpërndahen normalisht përmes tregjeve të aplikacioneve siç janë Google PlayStore, kështu që është e mundur të aktivizoni instalimin nga skedari APK ose përmes lidhjes USB në cilësimet e pajisjes . Për ta instaluar APK ajo duhet të ketë një emër unik package name për të ruajtur informacione meta. Pas instalimit ky emër regjistrohet në menaxherin e paketës. Për aplikacionet e shfaqura në Google Play emri i paketës mund të shihet në URL. Pasi të kemi krijuar një aplikacion dhe jemi në fazën përfundimtare me bo

run atë aplikacion në java, run bëhet përmes metodës public static void main dhe thirrët i gjithë kodi se çka përmban në të, mirëpo te android e kemi metodën onCreate dhe brenda kësaj metode thirrret gjithçka se çka kemi deklaruar brenda saj do të thotë kjo se pjesa e dukjes së aplikacionit ose më mirë të themi e pamjes (views) duhet të inicializohet brenda kësaj metode onCreate në hapat në vazhdim do të shohim më hollësisht mënyrën e përdorimit. Platforma android aplikacionet është e ndanë në shumë versione dhe ne do ti cekim disa prej tyre: versioni Alpha 2008, Beta 2009, Cupcake 2009, Donut 2009, Eclair 2009, Froyo 2010, Gingerbread 2010, Honeycomb 2011, IcecreamSandwich 2011, JellyBean 2012, KitKat 2013, Lollipop 2014 , Marshmallow 2015, Nought 2016 , Oreo 2017 , Pie 2018 dhe është versioni i fundit i android-it. Më poshtë do të paraqesim një figurë të versioneve.[1]



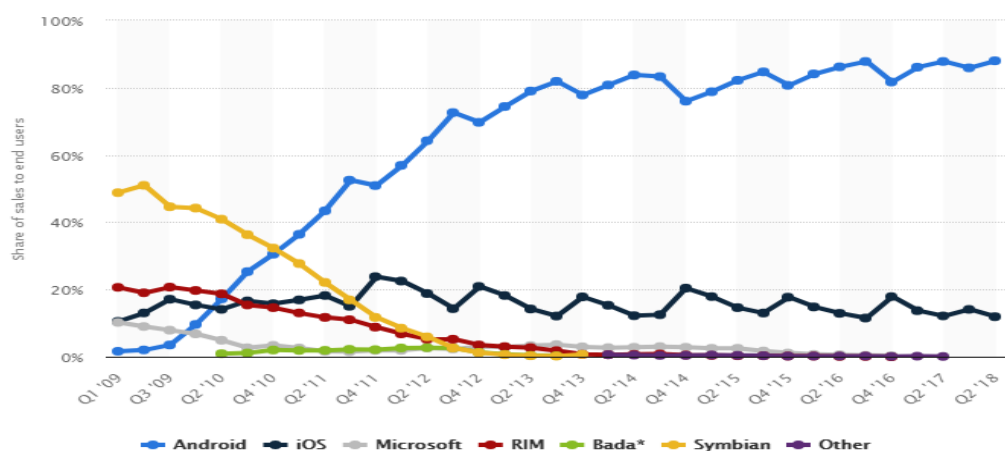
Figura 1. Historia e Aplikacioneve Android¹

2.2 Analizimi i Aplikacioneve Mobile

¹<https://ictinfozone.blogspot.com/2015/09/android-version-history.html>

Analiza është faza e parë e zhvillimit të një projekti softuerik , në të cilën mblidhen edhe strukturohen të gjitha kërkesat e klientit për softuerin që e projekton. Si i tillë , kjo fazë ndikon pashmangshëm në fazat e më vonshme të projektit , prandaj edhe duhet ti kushtohet vëmendje më e veçantë e përzgjedhjes së metodologjive shkencore përkatëse. Në këtë kapitull sqarohen kërkesat burimore për zhvillimin e këtij softueri dhe bëhet formatizimi i këtyre kërkesave sipas praktikave të zgjedhura të inxhinierisë softuerike. [1]

Para se të fillojmë të zgjidhim një problem, duhet të dimë se çfarë problemi është dhe çfarë analize duhet bërë. Si për shembull ndodhi një gabim gjatë analizimit të projektit - një problem tjetër gjatë instalimit të programit. Nëse gjatë instalimit të aplikacionit pajisja për ndonjë arsye nuk mund të vazhdojë atë proces, ajo automatikisht do ta shfaq një mesazh dhe do të ndaloj instalimin. Ne do ti shprehim disa arsye pse mund të shkoj keq gjatë instalimit të aplikacionit: versioni android për të cilin është shkruar programi që do të instalohet nuk përputhet me versionin android të instaluar në telefonin tuaj. Për shembull, ju keni një telefon me sistemin operativ version 3.0, dhe programi është shkruar për versionin 4.0.3. Nëse versioni i Android-it tuaj është më i lartë se versioni për të cilin është shkruar aplikacioni, nuk do të ndodhë asnjë gabim por nëse ndodh e kundërta do të kemi probleme sepse ne po tentojmë me një version më të vjetër me përdor një version më të ri gjë që telefoni juaj apo emulatori nuk e përkrah. Po ashtu mundësia është se ka viruse që ndërhyjnë në procesin e instalimit të aplikacionit tuaj. Ne do ti shfaqim më poshtë disa krahasime të platformave të ndryshme ndër vite duke i paraqitur ato në mënyre grafike në raport me aplikacione të ndryshme si Ios, Microsoft, Rim, Bada, Symbian, Other. [2]



© Statista 2018

Figura 2. Paraqitja grafike e zhvillimit të aplikacioneve²

2.3 Avantazhet e aplikacioneve mobile

Aplikacionet mobile janë zgjidhja më e mirë që kërkon zhvillim dhe përdorim të rregullt në industri, një aplikacion mobil lejon përdoruesit të vendosin preferenca të krijojnë llogari dhe të mbajnë informacion jetik në dorë. Në bazë të biznesit një aplikacion mobile është një mundësi shumë e mirë për të synuar një audiencë dhe për këtë arsye ndërtojnë fushata marketingu për përdorues të ndryshëm. Një tjetër aftësi e integruar e aplikacioneve mobile është aftësia e notifications tek përdoruesi që kanë instaluar aplikacionin në sistemin e tyre. Nga ana tjetër duke u dhënë botuesve mundësin që të dërgojnë mesazhe drejtpërdrejt përdoruesit. Lehtësia për të dërguar njoftimin është një nga arsyet kryesore pse shumë biznese dëshirojnë të kenë një aplikacion mobile në radhë të parë, sepse e ka menaxhimin e punës më të lehtë si për stafin e tij ashtu edhe për menaxhimin e biznesit. Një avantazh tjetër është i përdorimit mobil është mundësia e përdorimit të aplikacioneve edhe kur është offline. Arsyja është se aplikacionet janë të instaluar në një pajisje të lëvizshme, ato do të vazhdojnë të ofrojnë përmbajtje dhe veçori pa konektim të internetit. Aplikacioni ruan të dhënat në nivel lokal dhe më pas i vendos ato pasi të bëhet një connection kërkesat e jashtëzakonshme të

²<https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>

mëdha dhe të bëhet zgjerimi i tregut. Ne nesër nuk jemi të kufizuar të punojmë vetëm në tregun vendor , mundësit tona shtrihen në tregun global. Bizneset gjithnjë janë në kërkim të zhvilluesve android me provojë. Fleksibiliteti i punës dhe orarit, ne mund të punojmë me zhvillimin e aplikacioneve vetëm me gjysmë orari punë, orar të plotë, në zyre, freelancer, mundësit të tjera të cilat nuk na kushtëzojnë në kohë dhe vendbanim. Paga konkurruese, ne mund të sigurojmë një pagë të mirë madje edhe vetëm me disa orë pune në ditë. Zgjerimi i tregut të pajisjeve mobile , pajisjet mobile çdo ditë e më shumë po hynë në përdorim dhe po janë të rëndësishme në jetën tonë sociale dhe biznesore. Fuqizimi i këtyre pajisjeve paraqet edhe rritjen e kërkesave të zhvilluesve softuerik në industri. Për nga e kundërta e anës së mirë të aplikacioneve mobile i kemi edhe anën negative të aplikacioneve mobile, prandaj më poshtë do të sqarojmë edhe do ti shohim edhe disavantazhet e aplikacioneve mobile.[2]

2.4 Disavantazhet e aplikacioneve mobile

Disavantazhet e aplikacioneve mobile janë: Mbrojtja e pamjaftueshme e shtresës së transportit, autorizimi dhe vërtetimi i dobët, injeksioni nga ana e klientit, trajtimi i gabuar i audiences, rishikimet e sigurisë, rrjedhja e të dhënave në kanalin anësor malware. Ne duhet të kemi nevojë për kriptimin TLS/SSL me algoritme të forta brenda komunikimeve. Gabimi unik është lidhjet e paskriptuara të aplikacionit në një kompani të tretë për zhvillim të aplikacioneve. Këto dobësi janë vendosur nga ana e serverit. Ushtrimet më të mira që duhet ti ndjekim janë të njëjtat me aplikacionet në internet. Sidomos për zhvillimin e aplikacioneve identifikuesit e pajisjeve duhet të injorohen pasi pajisjet mund të hiqen dhe të ndërhyhet në të. Në shkarkimin e aplikacioneve mobile një numër i madh i shkarkimeve mund të jetë nga fëmijët duke e shikuar fotografën apo ikonën e aplikacioni mobile duke kërkuar lojëra. Në fakt duke e shikuar një mesatare të rriturit shkarkojnë më pak se një ose shumë aplikacione në muaj se sa fëmijët. Përditësimi dhe mirëmbajtja nuk ka vetëm investime në zhvillimin e parë të një faqe në internet ose aplikacione native mobile. Të dy gjithashtu duhet të azhurnohen rregullisht dhe sipas nevojës sa herë që ka ndonjë ndryshim të produktit ose shërbimit. Llogaria për përditësimet e Android dhe Ios dhe kjo mund të jetë për tre zhvillues të ndryshëm që duhet të ngarkohen me tre grupe azhurnimesh , jo vetëm me kaq por bizneset të cilat krijojnë aplikacione mobile kërkojnë kohë për të ri-aprovuar aplikacione të

azhurnuara dhe që konsumatorët të përfitojnë nga një azhurnim, ata ose duhet të shkarkojnë azhurnime softuerësh ose të ri-shkarkojnë një aplikacion të tërë. Tjetra në listën tonë të disavantazheve të aplikacioneve mobile është mënyra e ndarjes dhe e shpërndarjes së tyre. Kur bëhet fjalë për aplikacione vendase, përpjekjet tuaja të marketingut zbehën drejt vendeve të shumta të cilat ndoshta përfshijnë: ueb faqja juaj , aplikacioni juaj në google play , po ashtu edhe në app store, dhe mbase edhe më shumë nëse aplikacioni juaj renditet në dyqanet e tjera të aplikacioneve të jashtë atyre të përmendura më sipër. Kjo do të thotë që ju do të duhet të zhvilloni një strategji shumë efikase për secilin kanal të veçantë. Në krahasim aplikacionet progresive të uebit mund të ndahen me një url të vetme ose një kod QR. Një nga disavantazhet më të mëdha mund të jetë edhe kostoja e një aplikacioni mobile native(vendase) që kushton mijëra dollar. Dyqanet e aplikacioneve mobile marrin gjithashtu një komision përqindje nga shitjet deri në 30 % , pastaj ka kosto edhe të marketingut. Ndërsa aplikacionet native varen plotësisht nga përpjekjet e marketingut dhe vlerësimi i klientëve.[2]



Figura 3. Mobile Application³

2.5 Aplikacioni për motin

Aplikacionet mobile janë duke u bërë ndër aplikacionet më të nevojshme në botën e biznesit, në bazë të idesë që e kemi do të projektojmë një aplikacion për shfaqjen e temperaturave të vendeve të ndryshme të botës. Këto aplikacione të tilla kemi edhe në teknologji të ndryshme

³<https://www.indiamart.com/proddetail/android-mobile-application-development-services-20821307233.html>

si për shembull në web , në desktop , etj. Por qëllimi jonë është ta krijojmë në aplikacion mobile që mendoj se është më i përdorshëm dhe ka lehtësi të përdorimit për klientin.

Ne kemi menduar se zhvillimi i këtij aplikacioni është i rëndësishëm si dhe na ndihmon nëse jemi në ndonjë vend të panjohshëm dhe dëshirojmë ta shohim motin se çfarë temperatura kemi ose jemi duke planifikuar diçka që na duhet patjetër të jetë koha ashtu siç dëshirojmë dhe përmes aplikacionit mund të shohim matjen e saktë të temperaturave si dhe temperaturat maksimale që do të arrijnë brenda ditës si dhe ato minimale.[3]

Fillimisht duke hulumtuar në internet dhe duke kërkuar api të hapur (pa pagesë) që mund të shfrytëzojmë kemi hasur në ueb faqen openweather ku gjatë familjarizimit me ueb faqen dhe nxjerrjen e kërkesave kemi ardhur në përfundim se mund të zhvillojmë një aplikacion ku do të na shfaqen të dhënat si: temperatura maksimale, temperatura minimale, lagështia, shtypja e ajrit, dhe mundësia e pamjes së lokacionit me ndihmën e gps. Për zhvillimin e këtij aplikacioni do të përdorim editorin Android Studio i cili është i vetmi editor që e kemi për zhvillimin e aplikacioneve mobile në android. Për përdorimin e teknologjive ne do të flasim më në detaje te pjesa e rezultatit se çka kemi përdorur për zhvillimin e aplikacionit nga fillimi deri te përfundimi. Ky aplikacion mendoj se mund të jetë i përdorshëm edhe për disa klientë të cilët profesioni i tyre i punës është i lidhur ngushtë me kohën apo temperaturat. Si për shembull punëtorët që punojnë në bregdet , apo punëtorët që janë nëpër qendra të skijimit gjatë sezonës dimërore dhe po ashtu mendoj se shërben edhe për bujqit që e punojnë tokën. Më pastaj do të shohim pak ecurinë e zhvillimi të projektit në pika të shkurta ku te pjesa e rezultatit do të flasim më gjerësisht.

2.6 Krahasimi i aplikacioneve android

Pas analizimit të aplikacionit më lartë që do ta krijojmë do të paraqesim krahasimin e aplikacioneve android që kemi zgjedhur në editorin android studio. Më poshtë do të paraqesim një tabelë të disa funksioneve që do ti përkasin aplikacioneve android që janë mobile aplikacionet, aplikacionet televizive, aplikacionet për orë të dorës etj. Këto aplikacione të përbashkëta e kanë gjuhën programuese që përdoret java dhe kotlin.

Aplikacionet	Mobile dhe Tablet	TV	Orë e mençur
Gjuha e zhvillimit	Java/Kotlin	Java/Kotlin	Java/Kotlin
Web	Po	Jo	Jo
Çmimi	400 \$	50 \$	100 \$

Tabela 1. Krahasimi i aplikacioneve.

Pas krahasimeve në bazë të aplikacioneve dhe çmimit vijmë në përfundim që aplikacionet mobile android është aplikacioni me çmim më të lartë në krahasim me aplikacionet TV dhe Smart Watch për arsye sepse aplikacionet mobile dhe aplikacionet për tablet janë më të përdorshme sepse shërbejnë për restaurante, hotele, marrjen e porosive nëpër markete kanë gjetur zbatim më të madh në krahasim me aplikacionet tjera siç janë tv dhe që ka një çmim 50\$ po ashtu edhe smart watch që ka çmimin më të vogël se mobile për 100\$.

Sa i përket krijimit të aplikacioneve ueb në android nuk kemi mundësi të krijojmë nuk ofron kurrfarë zgjidhje. Po ashtu për zhvillimin e këtyre aplikacioneve përdoren dy gjuhë programuese por ne kemi zgjidhur gjuhën programuese Java që mund të implementojmë aplikacionin tonë sepse pasi që kemi njohuri me Java. Gjuha programuese Kotlin është një gjuhë e re e krijuar në vitin 2011 dhe me avancimin e teknologjisë kotlin aplikacionet mobile android do të kalojnë vetëm në gjuhën programuese kotlin.

3 DEKLARIMI I PROBLEMIT

Si në çdo sistem softuerik i kemi edhe deklaratimet e problemit që e kemi zhvilluar në aplikacionin tonë. Gjatë zhvillimit të këtij projektit duke kaluar nëpër faza të shumta kemi pasur probleme të ndryshme. Si për fillim duke menduar rreth krijimit të aplikacionit idesë se si duhet ta bëjmë çfarë teknologjie duhet përdorur, cilat janë veglat për përdorim të aplikacionit etj. Duke hulumtuar dhe analizuar kemi arritur në përfundim se për zhvillimin e aplikacionit kërkesat tona ishin të gjegjë një api free ku jep të dhëna pa pages, gjatë hulumtimit kemi parë disa api të hapur ku jep të dhëna për motin në kohë reale përmes ueb faqes open weather. Kur kemi hyrë në ueb faqe ku ofron api të hapur për shkëmbimin e informatave ndërmjet klientit dhe serverit kemi definuar se çka mund të projektojmë dhe si do të doket aplikacioni jonë. Gjatë analizimit të informatave të API për komunikim kemi të detyruar ta krijojmë një api key ku është unik dhe ti je identifikuar me atë çelës ku nuk mund të jetë i ngjashëm me një çelës tjetër dhe as nuk mund ta përdor dikush tjetër atë çelës.

Procesimi tjetër i deklaratimit të problemit në aplikacionin tonë është nëse përdorimin e teknologjive nuk e kemi bërë import tek gradle. Kjo do të thotë se detyrimisht duhet ta implementojmë te pjesa gradle si për shembull implementimin e picasso, implementimin e retrofitit, implementimin e konvertimit të gson, si dhe ky aplikacion për komunikim me serverin kërkon internet dhe duhet bërë permission internetin që na mundëson përdorimin çdo herë të aplikacionit.

Deklarim tjetër i problemit është nëse kërkojmë një request nga serveri mirëpo komunikimin me të e bëjmë jo në mënyrë të duhur. Kërkesa për komunikimin me klientin janë dy dhe ato janë id e krijuar dhe tjetra është qyteti vendi ku dëshirojmë ta shohim. Me këto të dyja ne mund të komunikojmë me serverin për marrjen e inputit të dhënave të atij vendi. Ndërsa si problem që mund ta definojmë është nëse njërin nga këto të dyja nuk e kemi në link, nuk kemi mundësi të komunikojmë me serverin sepse api jonë nuk arrin të komunikoj me serverin nëse nuk e ke plotësuar kushtin e kërkuar për komunikim që janë dy parametra id dhe emri i qytetit.

4 METODOLOGJIA

Metodologjia, planifikimi dhe menaxhimi janë parakushte kyçe që përcaktojnë suksesin e përgjithshëm të një projekti. Për rrjedhojë, është një rëndësi vitale që qysh në fillim të zhvillimit duhet të definohen struktura e projektit edhe hapat që do të ndiqen në realizimin e suksesshëm të tij. Është e ditur se në çfarëdo projekti, problemet janë pothuajse të paevitueshme.

Për t'u informuar dhe kuptuar më shumë në lidhje me temën dhe aplikacionet të ngjashme dhe teknologjitë e tyre, ne kemi bërë hulumtime në artikuj shkencorë dhe materiale të ndryshme duke përdorur libra si dhe informata nga interneti, për tu informuar më shumë rreth këtyre aplikacioneve mobile, në metodologjinë tonë kemi përfshirë disa metoda siç janë:

Përmes shfletimit të literaturës kemi analizuar aplikacione të ndryshme të ngjashme që përdoren në botë, teknologjinë që këto sisteme përdorin.

Analizimi i të dhënave, kemi hulumtuar materiale të ndryshme rreth funksionimit të aplikacioneve të tjera të ngjashme me aplikacionin që do ta zhvillojmë ne, kemi analizuar gjithashtu se si këto aplikacione sjellin benefite dhe ju ndihmojnë përdoruesve si për shembull studentëve, personave të cilët udhëtojnë, janë gjatë rrugës apo punojnë etj.

Krahasimi i aplikacioneve, metodë kjo që është përdorur për të kuptuar përparësitë dhe mangësitë e këtij aplikacione në pajisje të ndryshme.

Zhvillimi i aplikacionit, pasi që e kemi analizuar aplikacionet e ngjashëm dhe funksionet e këtyre aplikacioneve më të përdorura ne kemi ardhur në përfundim se mund të zhvillojmë një aplikacion për pajisje mobile i cili na përshtatet dhe i përmbush kërkesat tona.

5 REZULTATET

Në këtë kapitull do të paraqesim hapat për zhvillimin e aplikacionit për menaxhimin e motit, po ashtu do ti paraqesim të gjitha veglat e përdorura si dhe teknologjitë të cilat na kanë ndihmuar për zhvillimin e aplikacionit.

Gjatë këtij kapitulli do të shtjellojmë më në detaje projektin tonë dhe do të tregojmë ecurinë e realizimit të aplikacionit si dhe përdorimin e teknologjive. Do të flasim për zhvillimin dhe analizimin e projektit, android studion, android manifest, implementimi i gradle, application programming interface, zhvillimi dhe ndarjen e klasave të aplikacionit, gjenerimi i key, menaxhimi i riskut, sigurinë, dhe paraqitjen e disa pjesëve të kodit të cilat janë më të rëndësishme.

5.1 Zhvillimi dhe analizimi i projektit

Aplikacion që do ta ndërtojmë do të jetë në aplikacionin mobil duke e përdorur android studio ku na mundëson zhvillimin e aplikacionit tonë. Pasi që e kemi analizuar dhe planifikuar se a mund të realizohet edhe ka rezultate të pritshme dhe kush janë palët e interesit kam vendosur ta krijoj aplikacionin për shfaqjen e motit. Fillimisht për zhvillimin e këtij aplikacioni më ka nevojitur API ku përmban të dhëna statistikore reale në të gjitha vendet e botes edhe unë këto të dhëna i kam hulumtuar nga interneti në ueb faqen openweathermap.org ku kam krijuar accountin (llogarin) time dhe pasi që e kam krijuar më ka gjeneruara një key (çels) personal ku përmes atij key ne kemi marrur informata nga vende të ndryshme të botes që i kemi thirrur për shumë vende ku çelësi jon ka kontaktuar me serverin në londër. Pasi që i kemi marrur informatat neve si klient na ka kthyer një informacion në Json file ku për neve i kemi klasifikuar dhe i kemi ndarë në attribute ku në ato informacione kemi pasur: koordinatat latitude longitude, emrin e qytetit, temperaturën maksimale, temperaturën minimale, shtypjen e ajrit, nje ikon në të cilën nëse koha ka qenë me diell e kemi pasur ikonën e diellit ose në të kundërtën me gjendjet tjera atmosferike në varësi me motin si për shembull borë shi, frymë etj. Pasi që e kemi marrur API dhe e kemi menduar se si do të duket aplikacionit jonë, kemi filluar me pjesën e butonave, text view, ku i kemi përdorur dhe i kemi lidhur njëri me tjetrin me constraint layout pasi që i kemi lidhur dhe e kemi parë pamjen tonë se si

do të jetë kemi filluar me pjesën logjike të kodit ku komunikimi me serverin fillon prej `base_url` se bashku me id tonë të cilin e kemi gjeneruar për marrjen e të dhënave, ku neve na obligohet që të shkruajmë emrin e vendit ku po dëshirojmë ta shikojmë temperaturën dhe përmes `onClick` të butonit ku pranohet input e marrim emrin e qytetit nëse nuk është gabim emri i qytetit e thirr funksionin ku i kemi deklaruar `base_url`, edhe id tonë, dhe na shfaqë të dhënat në kohë reale se si janë në server, nëse nuk e gjen vendin na shfaqë një error ku tregon se emrin e qytetit apo vendit nuk e ka në rregull duhet rregulluar. Po ashtu kur të na shfaqen të dhënat e një vendi përmes tekstit që është `weather in london` or `weather in kosovo`, me një klik të atij text view që i qasemi na shfaqet lokacioni përmes GPS që kalon prej një aktiviteti në aktivitetin e google maps dhe mund ta shohim për shembull `london in map` ose ku e kemi caktuar ne si përdorues `addMarker` që është një objekt i `MarkerOptions`.

5.2 Android Studio

Në këtë editor që quhet Android Studio ne do të krijojmë aplikacionin tonë fillimisht para se të hyjmë në fazën e zhvillimit të projektit i kemi disa hapa në të cilën ne duhet t'i selektojmë se çka po dëshirojmë tek zhvillimi i aplikacionit ne duhet të kemi për obilgim zgjidhjen e versionit se cilin version po dëshirojmë me ndërtu projektin po ashtu duhet tek manifesti të bëhet konfigurimi dhe implementimi se çka kemi për të shfrytëzuar në mënyrë që mos të na shfaqet ndonjë problem gjatë rrugëtimit të zhvillimit në aplikacionin tonë, mandej duhet të zgjedhim se apo dëshirojmë me zhvillu aplikacion në mobile, tv, tablet apo ku kemi vendosur ne, duhet të zgjedhim aktivitetin se çfarë dëshirojmë të zhvillojmë kemi dy aktivitete që do të shfrytëzojmë dhe që do të lidhen njëri me tjetrin si për shembull google maps aktiviteti me empty aktiviteti, ky aktivitet google maps ka për qëllim shfaqjen e lokacionit të vendit për shembull nese dojmë me pa një vend përmes gps latitude dhe longitude që përmes kordinatave x,y e gjejm vendin e kërkuar, po ashtu e patëm edhe aktivitetin empty ku ne do të planifikojmë se si do të duket aplikacioni jonë që do të japim pamje të re se çfarë na pëlqen neve, pasi që e kemi zgjedhur aktivitetin duhet të fillojmë me pjesën e layout ku kemi të bëjmë me constraint layouts.[4]



Figura 4. Aplikacionet Android në pajisje të ndryshme.⁴

5.2.1 Android Manifest

Android Manifest është një xml file ku ka të bëjë me versionet encodimin paketën e aplikacionit tonë çka kemi krijuar , na tregon cila është klasa kryesore ku thirret aplikacioni jonë qysh do të doket pamja e ka një label, një theme, një icon etj. Po ashtu na tregon se ku kemi aktivitete që do të thotë ndërlidhja e klasës me klasën tjetër në android klasave i themi aktivitete. Mendoj se një deklaram i problemit është se tek android manifesti duhet të aplikohet permission për internet në rast se nuk e aplikojmë permission të internetit ne nuk mund të komunikojm me serverin. Patjetër duhet aplikuar permission të internetit për marrjen e informatave në server. Njëkohësisht duhet aplikuar edhe permission për access fine location për zhvillimin e aplikacionit tonë. Te ky xml file kemi edhe meta data që na shërbejn për shfaqjen e lokacionit të vendit ku kërkojmë e kemi çelësin tonë të implementuar këtu që e kemi gjeneruar nga developer konsolle përmes emailit tonë.

⁴<https://www.android.com/tv/>

5.2.2 Implementimi i Gradle

Gjatë rrugëtimit tonë në këtë aplikacion kemi edhe probleme tjera që na shfaqen një ndër to është edhe implementimi në gradle i veglave të cilat i kemi shfrytëzuar për ndërtimin e aplikacionit tonë. Nëse ne nuk i aplikojmë veglat tona siç janë: Gson, Retrofitin, Retrofit-Converter, Picasso, ne nuk mund të zhvillojmë aplikacionin tonë që e kemi menduar. Sepse për zhvillimin e aplikacionit tonë na duhet të përdorimi këto më lartë secila i kanë veprimet e veta që patjetër duhet implementuar për ti shfrytëzuar resurset e tyre përndryshe aplikacionit jonë nuk din si të veprojnë ndryshe, për implementimin e këtyre ne kemi hulumtuar në google dhe i kemi gjetur se si duhet implementuar me çfarë sintakse.[12]

5.2.3 Application Programming Interface

Çfarë nënkuptojmë me API ?

API, Application Programming Interface, është një paketë udhëzimesh dhe standardesh për softuer që lejon komunikimin e ndërsjellët p.sh. kur një uebsite përdor një dispozitiv për të shpërndarë një link në twitter ose facebook apo ndonjë rrjet social tjetër. Me API kemi parasysh shërbimet në ueb, gjë që bashkërenditet me tendencën për të përdorur API për të mbështetur shpërndarjen e materialeve dhe të dhënave midis komuniteteve dhe aplikacioneve. Materiali i krijuar në një vend mund të postohet dhe përditësohet në vende të ndryshme në Internet, TV, etj. API mund të kenë disa forma, publikimi i të cilave mundëson ndërtimin e aplikacioneve, dispozitivëve, website-ve dhe mjeteve të tjera, duke u bazuar në informacionin dhe shërbimet e agjencisë, në mënyrë që audienca të marrë atë që i nevojitet nga vende të ndryshme. API nuk janë eksperimentalë. Më shumë se gjysma e trafikut të kompanive të mëdha, si Twitter dhe eBay, kalojnë nëpërmjet API. API ndihmon në shtrirjen e informacionit të agjencisë tek përdoruesit e tjerë. API ofron akses në kohë reale të të dhënave, në mënyrë që një aplikacion të mund të aksesojë automatikisht informacionin më të fundit pa qenë nevoja që dikush të kopjojë nga uebsite i agjencisë çdo përditësimi. API dhe të dhënat e api e bën më të lehtë përdorimin e të aplikacionit. Ai është një mjet gjithnjë e më i rëndësishëm për publikimin e materialeve. Sidoqoftë, me një paketë të dhënash të disponueshme nëpërmjet API-t, është e rëndësishme që e gjithë paketa të mund të aksesohet duke përdorur “bulk download”.[14]

5.3 Veglat e përdorura

Për zhvillimin e aplikacionit tonë weather map dhe posedimin me të dhëna nga përgjigja e serverit për ndërtimin e këtij projekti janë përdorur një numër bukur i madh veglash dhe teknologji të ndryshme duke përfshirë si në vijim:

5.3.1 Web

Duke hulumtuar për zhvillimin e aplikacionit se ku mund të shfrytëzojmë API të hapur dhe pa pagesë ne kemi hasur në këtë ueb faqe open weather për krijimin e një aplikacioni të tillë siç do ta zhvillojmë ne.[10]

5.3.2 Java

Java është gjuha jonë programuese që do ta përdorim në zhvillimin e aplikacionit duke e pas parasysh se Android Studio na lejon të zhvillojmë projekte me java dhe kotlin. Për këtë program do të zgjedhim gjuhën të cilën e kuptoj dhe e kam mësuar pasi që njohurit fillestare i kemi marrur në fakultet dhe do të jetë më e lehtë për zhvillim.

5.3.3 Json

Json është një teknologji shumë e përdorshme ku në rastin tonë është mesazhi jonë që na kthen serveri pasi që e kemi bërë kërkesën (request) që do të thotë e kthen në mesazh të lexueshëm për userin. Po ashtu për formatim të json-it. Unë do të përdori një uebfaqe tjetër për formatim të json-it ku ajo ueb faqe e rregullon formatin e json dhe quhet Json formatter ku në këtë faqe më ndihmon pak se cilën mund ta bëjë klasë, cilën objekt etj.

5.3.4 Retrofit

Retrofiti është një teknologji tjetër që bashkëvepron me jsonin, përdorimi i saj është një teknologji tjetër për java dhe android. Kjo e bën relativisht të lehtë marrjen e attributeve me JSON përmes një shërbimi të bazuar në rest. Në retrofit ne konfigurojmë se cilin konvertues

përdoret për serializimin me të dhëna. Nga json i kthejmë në retrofit të dhënat për projektin tonë.

5.3.5 API key

Gjenerimi i një key (celsi) private ku përmes çelësit ne mund të kontaktojmë me serverin për marrjen e informatave dhe i krijojmë dy metoda ku përmes tyre thirret gjithçka se çka ka në përgjigje të serverit (Call<Response>()). Ky duhet të jetë unik dhe nuk mund të përdoret në aplikacione tjera.

5.4 Zhvillimi i aplikacionit dhe ndarja e klasave

Procesi i zhvillimit të aplikacionit është mjaftë i ndërlikuar, mirëpo me ndihmën e json formaterit kemi ardhur në përfundim të zgjidhjes së algoritmit tonë se si do ta zhvillojmë aplikacionin të ndarë në klasa. Pasi që kemi krijuar logjikën e zhvillimit të aplikacionit do ta analizojmë se cilat vegla nevojiten dhe duhet përdorur në aplikacionin tonë, në mënyrë që gjatë rrugëtimit të aplikacionit mos të kemi probleme madhore, pasi që dihet në çdo projekt problemet janë të pa evituara. Duke menduar edhe për testimin e aplikacionit tonë kemi menduar se si pjesë testuese të marrim duke i dhënë input të pa saktë do të thotë emrin e qytetit gabimisht për arsye të testimit se a është në nivel apo jo. Nëse neve gjatë ekzekutimit të kodit nuk na paraqitet ndonjë problem me emulator atëherë aplikacioni jonë bëhet run. Dhe kur të jemi të gatshëm që të shkruajmë emrin e qytetit nëse e shkruajmë emrin e qytetit saktë se si është nuk na paraqet problem asgjë dhe na shfaqen informatat e sakta, mirëpo nëse e shkruajmë emrin e qytetit gabimisht apo e shkruajmë ndonjë vend që nuk ekziston me gabime të qëllimshme atëherë na gjenerohet një mesazh që e kemi gabim emrin e qytetit të dhënë si për shembull nëse e marrim edhe e shkruajmë Kosova me dy shkronja më tepër kjo neve nuk na shfaqë të dhënat se serveri jonë nuk ka idenë se ku gjendet ky vend dhe lajmëron një tekst që emri i kërkuar nuk gjendet.

5.4.1 MainActivity

MainActivity është një klasë ku ekzekutohet e para në aplikacionin tonë dhe e ka metodën onCreate ku përmes saj thirret gjithë aplikacioni në këtë klasë ka atributet text view ku i

kemi përdorur për të ju referuar temperaturës minimale, temperaturës maksimale, temperaturës momentale, image view ku për neve image view është ikona e diellit, shiut etj, po ashtu e kemi edhe butonat, progress bar që shërben sa është duke komunikuar me serverin të bëhet load të na shfaqë një rrumbullak, deri në momentin kur ti shfaqë të dhënat dhe kryen komunikimin dhe më pastaj zhduket. Kjo klasë ka edhe metodat e veta që i kemi referuar API, `getWeatherFun(String city,String appid)` që i ka këto dy parametra, po ashtu e ka edhe metodën `getWeatherDataLocation(String latitude, String longitude)` ku përmes kësaj metode thirret aktiviteti tjetër google maps që i merr të dhënat e GPS.

5.4.2 ApiInterface

Ky interface përmban dy metoda, këto dy metoda shërbejnë për komunikimin me serverin marrjen e të dhënave mirëpo i kan dallimet e veta. Njëra metodë shërben për marrjen e të dhënave temperaturës max, temperaturës minimale, shtypjen e ajrit, etj ku duhet me pas gjithsesi një çels privat edhe një vend `base_url` se ku po don me kqyr temperaturën. Ndërsa metoda tjetër shërben edhe kjo për marrjen e të dhënave mirëpo kjo metodë tregon vendin pozitën gjeografike do të thotë latitude dhe longitude. Metoda përmban latitude, longitude si dhe id tonë të aplikacionit të cilin e kemi zhvilluar. Ky interface do të përdoret në klasën Main Activity ku do të zëvendësohet nga Json ne Retrofit.

5.4.3 MapsActivity

Kjo klasë Maps Activity përmban dy attribute për latitude dhe longitude kuptohet se këto dy attribute janë double po ashtu edhe kjo klasë e ka të gjeneruar metodën `onCreate` që thirret aplikacioni përmes saj mirëpo në të ka edhe një intent që do të thot ky aktivitet derivohet nga një intent tjetër që e ka një latitude dhe një longitude në këtë rastin tonë e kemi shfrytëzuar në klasën Main Activity. Po ashtu e ka edhe një metodë të veten që quhet `onMapReady()` ku shfaqja e të dhënave të vendit tonë bëhet nga kjo metodë sepse e ka metodën `addMarker` të objektit `MarkerOptions` dhe ja marrim vendin.

5.4.4 Coordinates

Klasa Coordinates është një klasë e cila përmban vetëm dy attribute double latitude, dhe double longitude një konstruktor default dhe e ka enkapsulimin getters and setters asgjë më tepër, mirëpo kjo klase krijohet një objekt i klases Coordinates që na kryen punë tek klasa WeatherResponse.

5.4.5 Weather Response

Klasa weatherResponse është një klasë e cila ka atributet visibility, base, main, coordinates, edhe një varg i weather ku do të ruhen atributet e klases weather. Te kjo klase nuk ka tjetër përveç getters and setters për atributet e dhëna .

5.4.6 Weather

Klasa weather është një klasë e cila ka atributet e veta të marrura nga përgjigjja e serverit ku ato attribute jane : id, main, description, edhe nje icon. Ku klasa main ka në vete atributet e veta që janë temp, temp max, temp min, humidity, pressure.

5.4.7 Gjenerimi i keys

Gjenerimi i qelsit (keys) në api duhet të jetë unik për një përdorues dhe nuk mund të përdoret për shumë user (përdorues) sepse nuk ka qasje. Nëse nuk e aplikojmë çelësin unik ne nuk mundë të komunikojmë me serverin i cili detyrimisht duhet të pranon një api key tonën që e kemi gjeneruar përmes emailit tonë. Atë qels ne duhet ta implementojm në manifest të projektit që në momentin kur aplikacioni jonë të bëhet run ekzekutohet çelësi dhe emri i kërkuar i vendit. Dhe mendoj se kjo është një deklarimi i problemit në raste të mos përdorimit të çelësit tonë privat.

5.5 Menaxhimi i riskut

Çdo projekt ka edhe rreziqe nga të papriturat që mund të dalin gjatë implementimit të tij dhe secili projekt mund të ketë rreziqe të ndryshme. Për një projekt të suksesshëm është të jetë e rëndësishme që të shqyrtohen të gjitha rreziqet e mundshme me të cilat mund të ballafaqohemi gjatë rrugës , në mënyre që kur ato të ndodhin të mos jenë të papritura. Përveç

kësaj , bazuar në rreziqet e mundshme duhet të shkruhen edhe strategjitë për mënjanimin e tyre. Prandaj edhe në rastin tonë janë hulumtuar rreziqet që mund të dërgojnë në ngecje apo dështim të projektit. Duke marr për bazë natyrën e projektit, përgatitjen profesionale të zhvilluesit , si dhe nga hulumtimet fillestare rreth kësaj , janë identifikuar këto rreziqe të mundshme:

5.5.1 Gabimi në llogaritjen e kohës së implementimit

Një ndër rreziqet e para është konsideruar të jetë planifikimi i gabuar i kohës që merr zhvillimi i projektit. Janë dy faktorë që kanë mundur të ndikojnë në paraqitjen e këtij problemi: mungesa e njohurive të zhvilluesit në disa prej teknologjive që janë planifikuar të përdoren dhe ndryshimi i kërkesave teknike në faza të vonëshme.

5.5.2 Mungesa e njohurive në teknologji të caktuara

Një sfidë në mbarëvajtjen e këtij projekti ka qenë edhe nevoja për përdorimin e teknologjive me të cilat zhvilluesi nuk ka qenë mjaft i familjarizuar.

5.5.3 Dështimi i gjetjes së testuesve potencial

Duke marrë parasysh se periudha e testimit të aplikacionit , sipas një planifikimi i bie të jetë gjatë kohës së mos funksionimit të aplikacionit tonë, atëherë është konsideruar se gjetja e problemit mund të jetë me api.

Planifikimi i mirë i rreziqeve të mundshme ka ndikuar të jetë pozitivisht në mirëmbajtjen e projektit , pasi që me disa nga këto rreziqe edhe ka ndodhur të ballafaqohemi. Strategjia e përgatitur ka rezultuar efektive pasi pjesa absolute e rreziqeve janë mënjeluar në saje të kësaj strategjie.

5.6 Siguria

Aplikacionet e bazuara në teknologjinë e Android Mobile kanë përparësi të shumta , si lehtësia e qasjes kudo , krijimi i mundësisë së përdorimit në pajisje të ndryshme , ndërtimi i ndërfaqeve që ofrojnë eksperiencë të admirueshme tek shfrytëzuesit etj. Por kjo veti vjen me koston e ekspozimit ndaj rreziqeve të sigurisë. Për më tepër, me arritjen e standardeve të reja

të teknologjisë është rritur edhe gama e sulmeve kibernetike. Pra siguria është faktor i rëndësishëm për çdo softuer dhe duhet ti kushtohet trajtim të jetë i veçantë. Prandaj edhe gjatë zhvillimit të projektit të marrjes së temperaturave janë ndërmarrë disa hapa për të garantuar sigurinë e saktë të sistemit. Siguria së bashku me privatësinë janë dy elemente të rëndësishme në një aplikacion, me qëllim që të ruhet integriteti i përdoruesit të aplikacionit dhe mos keqpërdorimi i kodit. Ky aplikacion që do të ofroj elemente të temperaturave përmes key-s ky key është personal dhe duhet ruajtur se me atë identifikohet zhvilluesi i aplikacionit dhe është vetëm për një përdorues.

5.6.1 Kodi i përdorur për komunikimin me serverin

Në këtë pjesë të kodit që e kam shkruar më poshtë komunikojmë me serverin duke i pranuar dy parametra njëra është id jonë e krijuar dhe tjetra është vendi i qytetit ku dëshirojmë ta shohim temperaturën. Mirëpo për thirrjen e këtyre dyjave i kemi implementuar me një interface ku mbishkruhet metoda në klasë dhe e zhvillojmë kodin. Në vazhdim informatat të cilat i marrim server na kthen përgjigjen në kelvin jo në grad Celsius dhe ne duhet ti konvertojmë në grad Celsius duke e zbritur me minus 273.15 .

```

public void getWeatherDataFun(final String city,final String appid) {
    ApiInterface mApiInterface = this.getInterface();

    final Call<WeatherResponse> mService =
mApiInterface.getWeatherData(city, appid);
    progressBar.setVisibility(View.VISIBLE);
    Log.i("Request ", mService.request() + "");
    mService.enqueue(new Callback<WeatherResponse>() {
        @Override
        public void onResponse(Call<WeatherResponse> call,
Response<WeatherResponse> response) {
            progressBar.setVisibility(View.INVISIBLE);
            WeatherResponse = response.body();

            if (weatherResponse != null) {

                Weather weatherInfo = weatherResponse.getWeather()[0];
                mCity.setText("Weather in " + weatherResponse.getName());
                cityName = weatherResponse.getName();
                String iconID = weatherInfo.getIcon();
                String URL = "https://openweathermap.org/img/w/" + iconID +
".png";

                Picasso.get().load(URL).into(weathericon);

                Main mainInformation = weatherResponse.getMain();
                double kelvinTemp = mainInformation.getTemp();
                int celsiusTemp = (int) (kelvinTemp - 273.15);

                Coordinates = weatherResponse.getCoord();
                latitude = coordinates.getLat();
                longitude = coordinates.getLon();

                mTemperature.setText(celsiusTemp + "°C");
                mdesription.setText(weatherInfo.getMain() + ", " +
weatherInfo.getDescription());

                double kelvinTempMin = mainInformation.getTemp_min();
                int celsiusTempMin = (int) (kelvinTempMin - 273.15);

                double kelvinTempMax = mainInformation.getTemp_max();
                int celsiusTempMax = (int) (kelvinTempMax - 273.15);

                mpressure.setText(String.format("%.2f",
mainInformation.getPressure()) + " hpa");
                mtemp_min.setText(celsiusTempMin + "°C");
                mtemp_max.setText(celsiusTempMax + "°C");

            } else {
                Toast.makeText(MainActivity.this, "Could not locate city ! "
+ city name.getText(), Toast.LENGTH_LONG).show();
            }
        }
    });
}

```



Figura 5. Paraqitja e layoutit në bazë të kodit.

5.6.2 Implementimi i Picassos

Picasso është një librari e imazheve për aplikacionet android janë të krijuar për tu mirëmbajtur dhe kujdesur për ngarkimet e përpunimeve të imazheve. Thjeshtëson procesin e shfaqjes së imazheve nga vendet e jashtme. Po ashtu imazhet shtojnë edhe kontekstin dhe aftësinë vizuale shumë të nevojshme për aplikacionet android. Picasso implementohet vetëm me një rresht të kodit . Ku duhet implementuar tek manifesti I aplikacionit si gradle në mënyrë që të importohet në klasat tona për përdorim . Në aplikacionin tonë e kemi përdorur Picasso për shfaqje të ikonave të kohës në rast se e kemi kohën me shi apo me diell në vendin e kërkuar Picasso e shfaqë ikonën adekuate në varësi se si është temperatura. Për implementimin e Picassos në aplikacionin tonë e kemi të definuar më poshtë.


```
implementation 'com.squareup.picasso:picasso:2.71828'
```



Figura 6. Paraqitja e implementimit të Picassos në layout.

5.6.3 Kodi për përdorimin e Retrofitit

Kodi më poshtë shërben për marrjen e të dhënave përmes api. Api na kthen përgjigje në formatin e gson dhe për neve duhet të jetë i lexueshëm ajo pjesë e kodit mirëpo neve na duhet patjetër ta konvertojmë gson në retrofit duke përdorur klasën retrofit ku i ka metodat e veta të implementuara builder dhe pranon një base url ku përmes këtij base url mund ta konvertojmë duke përdorur addConverterFactory dhe si parametër pranon një gson.

```
public ApInterface getInterface(){
    Gson gson = new GsonBuilder().setLenient().create();

    Retrofit retrofit = new
    Retrofit.Builder().baseUrl(BASE_URL).addConverterFactory(GsonConverterFactory.c
    reate(gson)).build();

    ApInterface = retrofit.create(ApInterface.class);
    return apInterface;
}
```

5.6.4 Kodi për marrjen e inputit të vendit

Kodi i zhvilluar më poshtë na shërben për pranimin e një teksti që ai tekst për neve do të jetë vendi i kërkuar që do të bëjmë kërkim që të na shfaqë të dhënat e temperaturës në kohë reale. Nëse te inputi i dhënë e shkruajmë vendin gabim të qytetit apo shtetit aplikacioni jonë do të lajmëron një mesazh ku do të tregon që nuk e ka gjet atë vend sepse nuk ka pasur mundësi të arrijë komunikimin serveri me kërkesën tonë të dhënë gabim.

```

button_submit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String cityName = city_name.getText().toString();
        if(cityName.isEmpty()){
            Toast.makeText(MainActivity.this,"Please fill a city name
before submitting",Toast.LENGTH_LONG).show();
        }
        else {
            getWeatherDataFun(cityName,APP_ID);
        }
    }
});

```

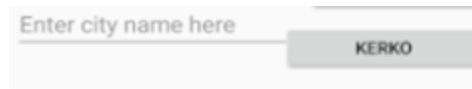


Figura 7. Pjesa e layoutit në bazë të kodit ku pranon input.

5.7 Funkcionaliteti i Aplikacionit

Në aplikacionin tonë të krijuar mënyra funksionimit të aplikacionit është shumë e lehtë , është e lehtë për arsye sepse në momentin kur ne e vendosim si input një emër të vendit ku dëshirojmë ta shikojmë dhe komunikimi jonë është i rregullt me serverin nuk paraqet pengesa dhe i shfaqë të dhënat në mënyrë korrekte pa dyshim që nuk kemi ndonjë problem apo error. Në momentin kur aplikacioni jonë bëhet run e ekzekutojmë na shfaqet emulatori ku thirret metoda kryesore statike onCreate() dhe çka ka brenda metodës gjithçka ekzekutohet, nëse nuk kemi ndonjë problem rreth komunikimit me serverin dhe serveri jonë na përgjigjet sipas informatave që i ka për vendin atëherë do të na doket layouti jonë si në figurën më poshtë. Duke e shfaqur temperaturën maksimale , temperaturën minimale, ikonën se a është me shi me borë apo me diell, si dhe duke e treguar edhe shtypjen e ajrit dhe lagështin.



Figure 8. Paraqitja e vendit të kërkuar

Në rast se në aplikacionin tonë nuk vendosim emrin e inputit apo vendit të kërkuar në mënyrën e duhur. Aplikacioni jonë nuk do të jep informata për atë vend vetëm se do të shfaqet një tekst ku do të tregon se emrin e kërkuar nuk gjendet në dhomën e serverëve duke kërkuar si informacion vendin. Prandaj duhet të kemi kujdes që mos të bëjmë gabime drejtshkrimore ndaj vendit të kërkuar.

6 DISKUTIME DHE PËRFUNDIME

Me këtë aplikacion që e kam zhvilluar përfundimi i këtij është se e kemi realizuar një aplikacion të besueshëm po ashtu edhe i kërkuar për klientët apo përdoruesit e ndonjë aplikacioni për matjen e temperaturës në vendin aktual apo edhe në vendet e ndryshme. Korrektësia dhe Performanca e këtij aplikacioni janë dy elemente bazë ku prioriteti i tyre është në nivel të lartë. Këto aplikacione mund të instalohen paraprakisht në telefona gjatë platformave të prodhimit , ose të dorëzohen si aplikacione në internet duke përdorur përpunimin nga ana e serverit ose nga ana e klientit për të siguruar një përvojë të ngjashme me aplikacionin brenda një shfletuesit të ueb. Zhvilluesit e softuerit të aplikimit gjithashtu duhet të marrin konsideratë një varg të gjatë madhësish të ekranit, specifikimeve të pajisjeve dhe konfigurimeve për shkak të konkurrencës së fortë në softuerin mobile dhe ndryshimeve brenda secilës prej platformave. Zhvillimi i aplikacioneve mobile ka qenë në rritje të vazhdueshme, në të ardhurat dhe vendet e krijuara. Një raport i analistit të vitit 2013 vlerëson se ka 529,000 punë të drejtpërdrejta në ekonominë e aplikacioneve brenda BE-së, më pas 28 anëtarë përfshirë edhe mbretërinë e bashkuar, 60 përqind e të cilave janë zhvillues të aplikacioneve mobile. Si pjesë e procesit të zhvillimit, dizajni i ndërfaqes së përdoruesit mobile UI është gjithashtu thelbësor në krijimin e aplikacioneve mobile. Ndërfaqja e përdorimit mobile konsideron kufizimet, kontekstet, ekranin hyrjen dhe lëvizshmërinë si skica për dizajnin. Përdoruesi është shpesh fokusi i ndërveprimit me pajisjen e tij dhe ndërfaqja përfshin komponentë si të pajisjes ashtu edhe të softuerit. Hyrja e përdoruesit lejon që përdoruesit të manipulojnë një sistem, dhe dalja e pajisjes lejon që sistemi të tregojë efektet e manipulimit të përdoruesit. Aplikacioni imë mendoj se është një aplikacion që ka një korrektësi dhe performancë të mirë, po ashtu edhe i përdorshëm për personat të cilët janë të interesuar për ndonjë aplikacion që ta shikojnë motin, për arsye sepse ne çdo herë dëshirojmë ta dimë temperaturën e vendit tonë apo ndonjë vendi tjetër se ku e kemi plan ta shikojmë. Po ashtu për ekzekutimin e këtij aplikacioni nevojitet permission me internet për arsye se nuk kemi mundësi të komunikojmë përmes api me serverin. Prandaj fillimisht duhet konektohem me internet dhe më pas ta përdorim aplikacionin tonë për nevojat tona.

7 REFERENCAT

- [1] A. Developer
<https://developer.android.com>
- [2] S. Amol dhe K. J. Delaney, "Introduction of Mobile Application ", Chicago, 2007.
- [3] A. Application, «Android Developer,» [Në linjë].
Available: <https://developer.android.com/reference/android/app/Application>.
- [4] A. Studio, «Android Studio,» [Në linjë].
Available: <https://developer.android.com/studio>.
- [5] B. Ryan, "Google is working on a mobile OS, and it's due out shortly", 2007.
- [6] M. Dan, Android Developer Blog, 2008.
- [7] Q. Pearnce James, "Google Strong Mobile Related Patent Portfolio", 2007.
- [8] L. Sean, "Mobile App usage grows 35%, TV and web not so much, 2012.
- [9] C. Eric, "Android Market Update ", 2009.
- [10] O. Weather, «Open Weather,» [Në linjë]. Available: <https://openweather.org/>.
- [11] J. Formatter, «Json Formatter,» [Në linjë]. Available: <https://jsonformatter.org/>.
- [12] L. Layout, «Android Devs,» [Në linjë].
Available: <https://developer.android.com/guide/topics/ui/layout/linear>.
- [13] R. L. f. Calls, «Retrofit Library,» [Në linjë]. Available: <https://square.github.io/retrofit/>.
- [14] G. Convertor, «Gson Convertor,» [Në linjë].
Available: <https://github.com/google/gson>.
- [15] L. L. a. Mascots, Linux Online, 2010.
- [16] A. D. GitBook, «Introduction of Android Developer,» 2020.