Razvoj večplatformne aplikacije za prikaz naprav pametnega doma

Sebastjan Mevlja

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija
sm9299@student.uni-lj.si

Sinopsis Danes aplikacije uporabljamo na različnih napravah in na različne načine. Praviloma so avtohtone aplikacije hitrejše, ponujajo več funkcionalnosti ter omogočajo boljšo uporabniško izkušnjo. Slabost teh aplikacij je, da jih je potrebno razviti za vsako platformo posebej, kar je pogosto drago in zahtevno. Telekom Slovenije, d. d., razvija avtohtone aplikacije za upravljanje NEO pametnega doma za vsako platformo posebej. Zanimalo nas je, ali bi lahko več aplikacij nadomestili z eno večplatformno aplikacijo. Naredili smo primerjavo večplatformnih ogrodij in za razvoj aplikacije izbrali ogrodje Flutter. Izdelali smo večplatformno aplikacijo za upravljanje NEO pametnega doma, ki omogoča pregled povezanih naprav. Aplikacija podpira Android in iOS mobilni platformi ter spletne brskalnike. Postavili smo tudi avtomatiziran proces testiranja in gradnje aplikacije z uporabo platforme Github Actions.

Ključne besede:
Flutter
Android
iOS
splet
večplatformne aplikacije
pametni dom
CI/CD

Opomba: Prispevek temelji na: Mevlja, S. (2022). Razvoj večplatformne aplikacije za NEO pametni dom: diplomsko delo, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana: S. Mevlja.

Univerzitetna založba Univerze v Mariboru

ISBN 978-961-286-639-6
DOI https://doi.org/10.18690/um.feri.10.2022.17

1 Uvod

Danes aplikacije uporabljamo na različnih napravah in na različne načine. Lahko jih namestimo na svoje mobilne telefone, tablice in računalnike, lahko jih uporabljamo preko spletnega brskalnika. Za doseganje priljubljenosti aplikacij med uporabniki morajo te ustrezati vedno višjim standardom – morajo biti privlačne, enostavne ter odzivne. Praviloma so avtohtone[1] aplikacije hitrejše, ponujajo več funkcionalnosti ter omogočajo boljšo uporabniško izkušnjo. Njihova slabost pa je, da jih je potrebno razviti za vsako platformo posebej, kar je pogosto drago in zahtevno. Podjetja morajo torej podobno aplikacijo razviti za več platform, za to pa potrebujejo razvijalce z različnimi znanji.

Po podatkih portala statista.com [1] je junija 2022 med vsemi operacijskimi sistemi na trgu prevladoval Android z več kot 44 %, sledi Windows z 19 % ter iOS s 17 %. Pogosto želimo z aplikacijami podpreti več trgov in platform, vendar je lahko razvoj takšnih aplikacij zelo drag, zato se podjetja vedno pogosteje odločijo za razvoj večplatformnih[2] aplikacij. Večplatformne aplikacije hkrati pokrivajo več trgov in podjetja lažje dosežejo več uporabnikov. S tem podjetja prihranijo tudi denar, ker opravijo manj dela za razvojni proces in vzdrževanje, olajša se zagotavljanje konsistence med platformami in izboljša se uporabniška izkušnja.

Telekom Slovenije, d. d., razvija aplikacijo NEO [2] za vsako platformo posebej, kar zahteva veliko časa in denarja. Obstoječa aplikacija podpira mobilne naprave Android in iOS ter spletne brskalnike. Trenutno ne obstaja večplatformna aplikacija za upravljanje NEO pametnega doma. Aplikacija poleg upravljanja pametnega doma omogoča tudi spremljanje televizijskih kanalov in videotek.. Osredotočili se bomo na del aplikacije za upravljanje pametnega doma, saj je kompleksnejši.

Glavni cilj je preveriti, ali je mogoča izdelava večplatformne aplikacije z uporabniškim vmesnikom v slovenščini za upravljanje NEO pametnega doma Telekoma Slovenije, d. d. Aplikacija bo omogočala prikaz naprav pametnega doma ter podpirala Android in iOS mobilne naprave ter spletne brskalnike. Za uporabo aplikacije bo potrebna prijava z NEO uporabniškim imenom in geslom. Najtežji del naloge bo predstavljala integracija obstoječega paketa za upravljanje pametnega doma OBLO [3], saj ta ne podpira večplatformnega razvoja in je razvit za vsako platformo posebej. Primerjali bomo najbolj priljubljena ogrodja za razvoj večplatformnih aplikacij in izbrali najustreznejše. Za preverjanje delovanja aplikacije bodo pripravljeni avtomatski testi. Postavili bomo tudi avtomatiziran proces testiranja in gradnje aplikacije.

2 Ogrodja za razvoj večplatformnih aplikacij

V današnjem konkurenčnem in hitro razvijajočem okolju razvijalci nenehno iščejo orodja za razvoj aplikacij, ki bi jim olajšala delo in zmanjšala stroške. Kot odgovor na to povpraševanje se je na trgu pojavilo veliko ogrodij za razvoj večplatformnih aplikacij. Primerjavo med avtohtonimi in večplatformnimi aplikacijami prikazuje tabela 1. Opisali bomo najpomembnejše prednosti večplatformnih aplikacij.

[1] avtohtona – izvorno "native"
[2] večplatformna – izvorno "cross-platform"

Tabela 1: Primerjava med avtohtonimi in večplatformnimi aplikacijami [9].

kategorija/vrsta aplikacije	avtohtona	večplatformna
arhitektura	odvisna od platforme	enotna
cena razvoja	višja	nižja
ponovna uporaba kode	nizka	visoka
dostop do strojne opreme	popoln	omejen
UI in UX	odvisno od platforme	konsistentno
zmogljivost	višja	nižja
ciljni trg	ena platforma	več platform
trajanje razvoja	daljše	krajše
število razvijalcev	več	manj

2.1 Ponovna uporaba programske kode

Največja prednost večplatformnih aplikacij je zagotovo ponovna uporaba kode. Razvijalci programsko kodo napišejo enkrat, čeprav aplikacija deluje na več platformah. To odpravlja ponavljanje in posledično prihrani čas ter stroške.

2.2 Cenejši in krajši razvoj

Ogrodja za razvoj aplikacij za več platform ponujajo dobro ravnovesje med kakovostjo in ceno, saj omogočajo, da z manj razvijalci, ki dobro obvladajo svoje področje, pokrijemo več platform. Manj časa, sredstev in truda, porabljenega za razvoj aplikacij, je neposredno sorazmerno z nižjimi stroški.

2.3 Enostavnejša namestitev in vzdrževanje

Ker se večina izvorne kode napiše samo enkrat, imajo razvijalci manj dela s pisanjem in vzdrževanjem kode.
To pomeni enostavno in hitro uvajanje sprememb, vzdrževanje, objavljanje posodobitev ter popravljanje napak.
Aplikacije na več platformah so posodobljene hkrati, kar je pomembno za konsistentnost.

2.4 Večji trg

Večplatformne aplikacije dosegajo širši tržni doseg, saj z njimi lažje dosežemo več uporabnikov, ki uporabljajo različne platforme.

2.5 Konsistenten uporabniški vmesnik

Večplatformne aplikacije nudijo enotni uporabniški vmesnik na različnih platformah, hkrati pa spoštujejo standarde, specifične za platformo. Če uporabnikom ponudimo konsistentno izkušnjo, bodo znali aplikacijo uporabljati na kateri koli platformi.

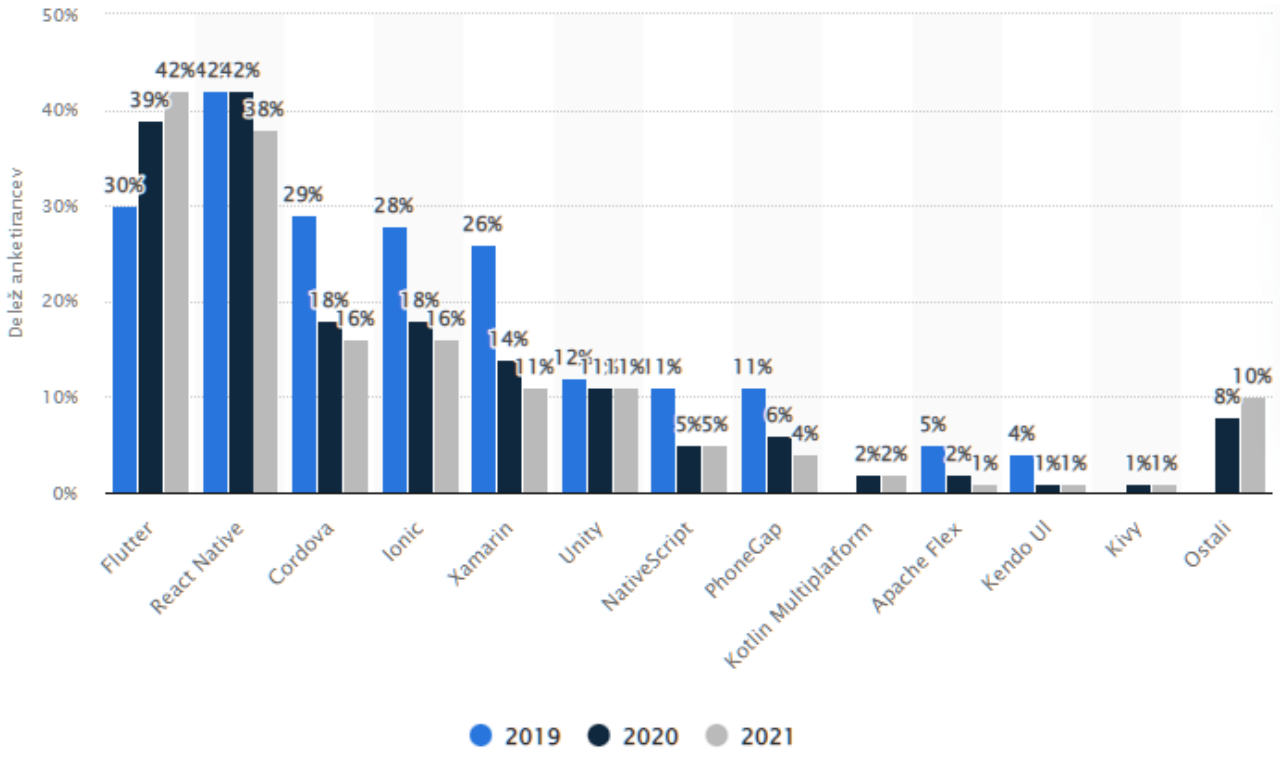
3 Primerjava večplatformnih ogrodij

Obstaja veliko ogrodij za razvoj aplikacij za več platform, ki imajo svoje prednosti in slabosti. Ker vsa ogrodja niso primerna za vse vrste aplikacij, smo pripravili primerjavo najbolj priljubljenih ogrodij, ki jo prikazuje tabela 2.

Tabela 2: Primerjava najbolj priljubljenih ogrodij [9].

kategorija/ogrodje	Flutter	React Native	Apache Cordova
Android podpora	da	da	da
iOS podpora	da	da	da
Windows podpora	da	da	da
Linux podpora	da	ne	ne
Mac podpora	da	da	ne
podpora za splet	da	da	ne
hitrost aplikacije	najvišja	srednja	najnižja
hiter ponovni zagon	da	da	ne
velikost aplikacije	srednja	velika	majhna
kompleksnost	srednja	srednja	nizka
sistemske funkcije	vse	večina	nekatere
vizualna konsistenca	da	ne	da
vtičniki in knjižnice	srednje	veliko	malo
programski jezik	Dart	JavaScript	JavaScript
plačljivo	ne	ne	ne
cena razvoja ($/h)	15–50	15–25	30–80
Priljubljenost	visoka	srednja	nizka
gradniki vmesnika	lastni	sistemski	lastni
poraba virov	nizka	srednja	visoka

Razna ogrodja za razvoj večplatformnih aplikacij so v zadnjih letih zelo napredovala in so vedno bolj priljubljena. Najbolj priljubljen je Flutter, ki je prehitel React Native. Uporablja ga 42 % razvijalcev večplatformnih aplikacij, medtem ko React Native 38 % [6]. To je tudi razvidno iz slike 1, ki prikazuje rezultate analize priljubljenosti večplatformnih ogrodij.



Slika 1: Priljubljenost ogrodij za razvoj večplatformnih aplikacij.
Vir: [4].

3.1 Izbira večplatformnega ogrodja

S svojo aplikacijo želimo podpreti Android in iOS mobilne naprave ter spletne brskalnike. Zaradi omejitev podpore različnih platform ogrodje Apache Cordova ne ustreza zahtevam. Zaradi večjega števila podprtih platform, hitrejših in manjših končnih aplikacij, dostopa do več sistemskih funkcij naprav, vizualne konsistence, manjše porabe virov [5] in višje priljubljenosti [4] smo za razvoj aplikacije izbrali ogrodje Flutter.

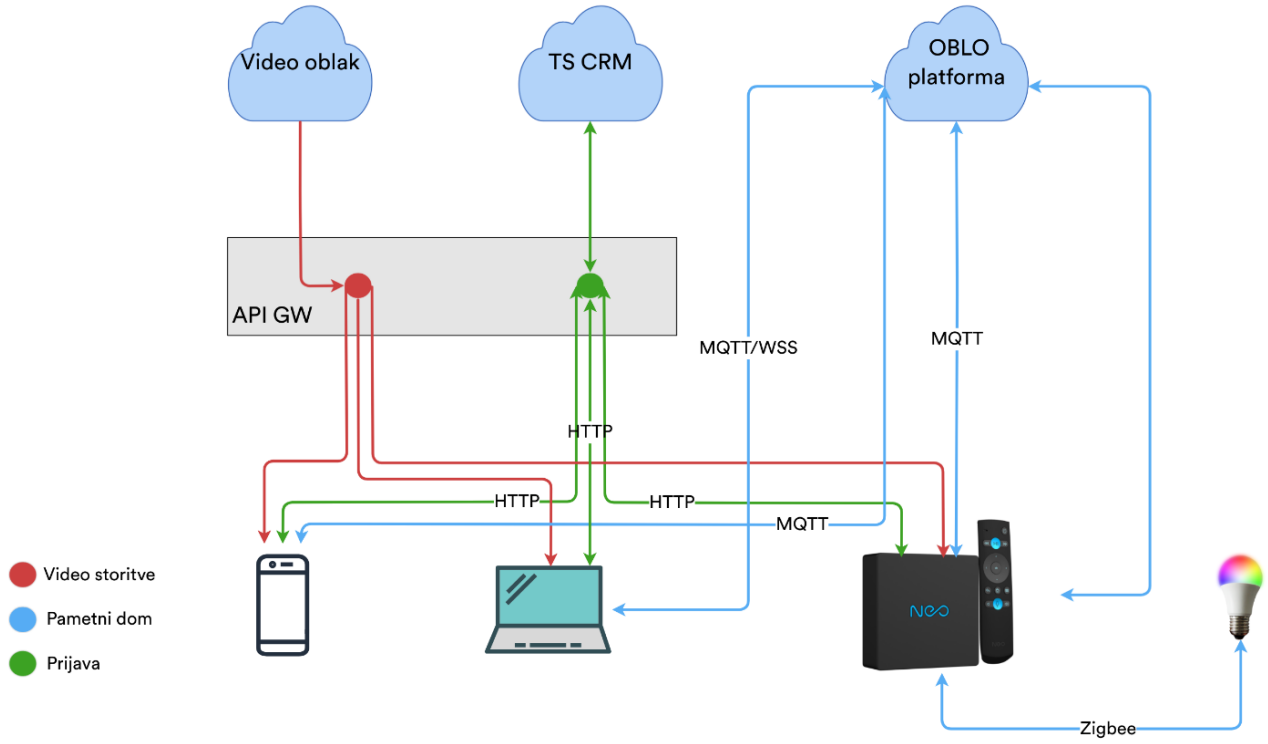
4 Orodja in tehnologije

4.1 Ogrodje Flutter

Flutter je brezplačno in odprtokodno ogrodje, namenjeno razvoju aplikacij za več platform. Omogoča hiter razvoj avtohtonih aplikacij za mobilne naprave, spletne brskalnike ter računalnike iz ene izvorne kode. Razvijalcem nudi vse, kar je potrebno za ustvarjanje aplikacij za več platform. Ponuja vse potrebne gradnike, izrisovalni pogon ter vmesnike za testiranje. Razvili so ga pri podjetju Google in ga tudi redno vzdržujejo. Njegova edinstvena lastnost je, da namesto uporabe sistemskih gradnikov, ki so na voljo v brskalniku oziroma na mobilni napravi, te upodobi sam s pomočjo lastnega 2D izrisovalnega pogona Skia. Za razvoj uporablja programski jezik Dart in omogoča integracijo z obstoječo avtohtono kodo (Kotlin in Swift). Na trgu obstaja že več kot pol milijona aplikacij, izdelanih v ogrodju Flutter [7].

4.2 Platforma NEO

NEO je platforma za pametno življenje Telekoma Slovenije, d. d.. Na enem mestu povezuje rešitve za dom in zabavo: spremljanje in napredno iskanje vsebin, upravljanje pametnih naprav, opravljanje nakupov preko televizijskega zaslona, glasovno upravljanje v slovenščini in igranje iger. Uporabniki lahko do platforme NEO dostopajo preko mobilne naprave, računalnika ali digitalnega sprejemnika v kombinaciji s televizijo. Vse naprave za spremljanje TV vsebin, avtorizacijo uporabnikov in upravljanje naprav pametnega doma pošiljajo sporočila na zaledni sistem, ki je sestavljen iz aplikacijskega vmesnika Stargate API Gateway in platforme za upravljanje pametnega doma OBLO. Digitalni sprejemnik NEO Smartbox je pogoj za uporabo pametnega doma, saj je med drugim tudi sprejemnik za Zigbee naprave. Za komuniciranje med napravami pametnega doma se uporablja protokol MQTT in platforma OBLO, ki ponuja vmesnike za pametni dom. Shema platforme NEO je prikazana na sliki 3.



Slika 3: Shema platforme NEO.
Vir: [9].

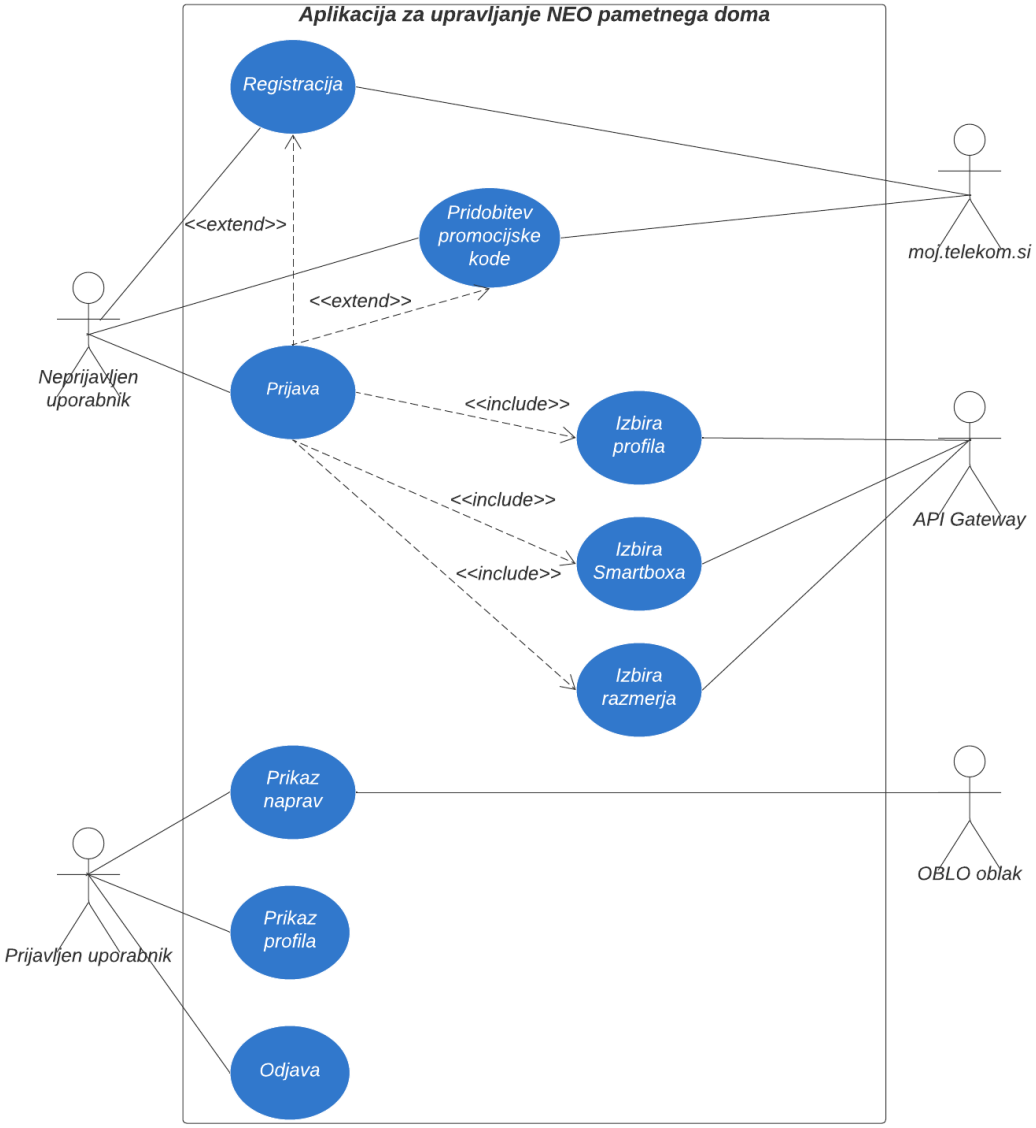
5 Razvoj aplikacije

Za razvoj smo izbrali ogrodje Flutter. Aplikacija podpira Android in iOS mobilne naprave ter spletne brskalnike. Med razvojem uporabniškega vmesnika smo sledili NEO oblikovalskim pravilom, zato je tema aplikacije temno modra z odtenkom vijoličaste. Namen razvoja aplikacije je bil preveriti, ali je možno razviti večplatformno aplikacijo za upravljanje NEO pametnega doma. Osrednji in bistveni namen aplikacije je prikaz naprav pametnega doma z uporabo protokola MQTT.

Med razvojem aplikacije je bil poudarek na uporabi dobrih praks in čitljivosti kode. Na začetku smo na podlagi uporabniških zgodb razvili statičen uporabniški vmesnik. Nadaljevali smo z implementacijo NEO prijave, s katero aplikacija pridobi sejo za dostop do zalednih sistemov. Sledil je najtežji del razvoja, to je priprava modula za povezavo s platformo OBLO in napravami pametnega doma. Zadnji del je bila povezava statičnega uporabniškega vmesnika s samo logiko aplikacije. Aplikacija je bila med razvojem temeljito testirana, napisanih je bilo tudi nekaj avtomatskih testov. Z uporabo CI/CD platforme Github Actions smo postavili delovni tok, ki skrbi za avtomatsko testiranje in grajenje aplikacije. Izvorna koda je bila na koncu formatirana in statično analizirana.

5.1 Funkcionalnosti aplikacije

Na sliki 4 so prikazani primeri uporabe aplikacije. Za pregled naprav pametnega doma je potrebna prijava z NEO uporabniškim računom. Če uporabnik računa nima, se lahko registrira ali zaprosi za pridobitev promocijske kode. Prijava je sestavljena iz več korakov. Uporabnik mora po vpisu uporabniškega imena in gesla izbrati uporabniško razmerje, profil in NEO Smartbox, na katerega se povezujejo naprave pametnega doma. Po uspešni prijavi je uporabniku omogočen pregled naprav pametnega doma, profila in odjava iz aplikacije.



Slika 4: Diagram funkcionalnosti aplikacije.
Vir: [9].

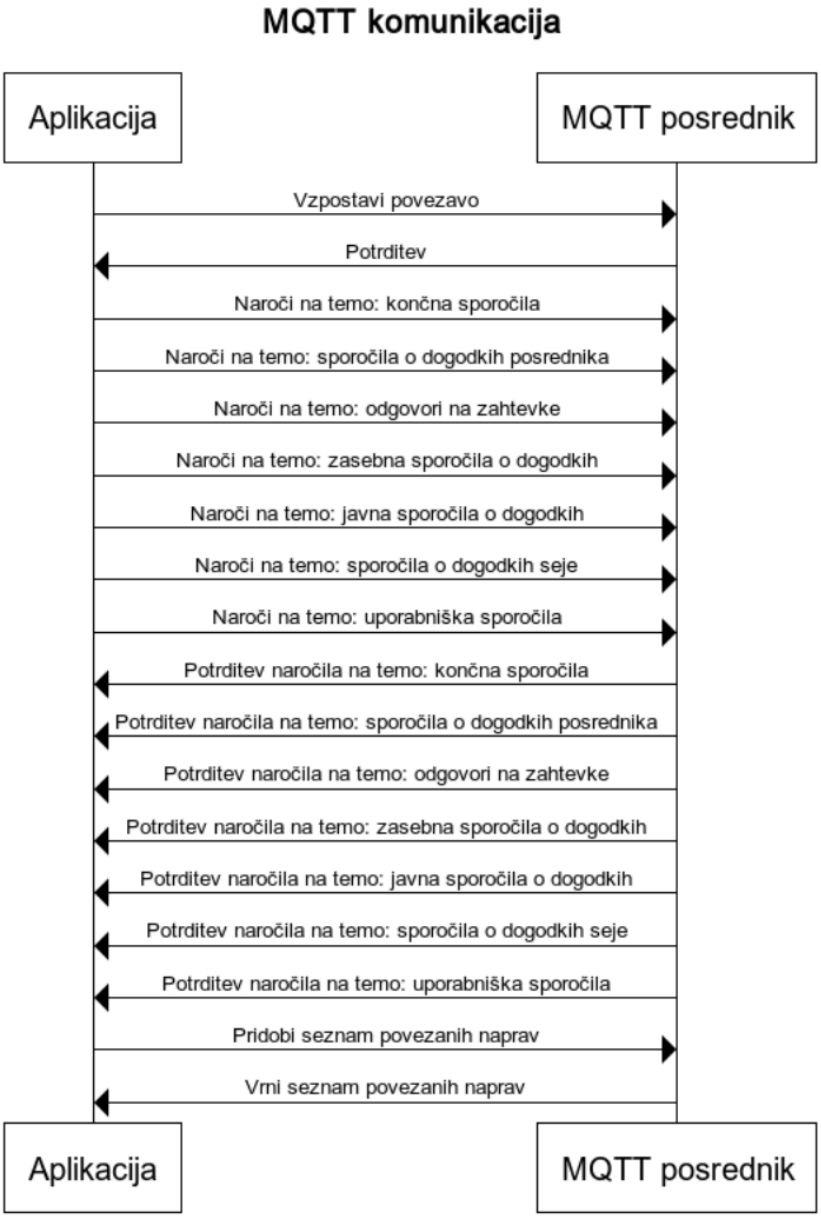
5.2 Integracije zalednih sistemov

Aplikacija uporablja obstoječi zaledni sistem Telekoma Slovenije, d. d. Za overjanje in avtorizacijo uporabnika se uporablja strežnik Stargate API Gateway. To je aplikacijski vmesnik, ki se uporablja tudi za komunikacijo z ostalimi zalednimi sistemi. Po uspešni prijavi uporabniku dodeli sejo za uporabo storitev pametnega doma.

Za delovanje pametnega doma se uporablja oblak OBLO, ki nudi vmesnike za pametni dom. Aplikacija se poveže na obstoječi posrednik MQTT, preko katerega pošilja in prejema sporočila. Naprave pametnega doma se z uporabo protokola Zigbee povežejo na digitalni sprejemnik NEO Smartbox, ki za komuniciranje z aplikacijo uporablja protokol MQTT.

5.3 Komunikacija MQTT

Komuniciranje z napravami pametnega doma poteka z uporabo protokola MQTT. Podatki se prenašajo v obliki JSON. Aplikacija vzpostavi povezavo s posrednikom MQTT in posreduje sejo, ki jo dobi ob prijavi. Način vzpostavitve povezave je odvisen od vrste naprave. Posebnost spletne različice je, da za komunikacijo uporablja spletne vtičnice[3]. Po vzpostavljeni povezavi se aplikacija naroči na vse potrebne teme in počaka na potrditev. Po prejeti potrditvi aplikacija zahteva seznam naprav pametnega doma, ki so trenutno povezane na NEO Smartbox. Posrednik aplikaciji pošlje seznam in naprave se prikažejo na uporabniškem vmesniku. Celoten postopek komunikacije prikazuje slika 5.



Slika 5: Diagram zaporedja poteka komunikacije MQTT.
Vir: [9].

[3] spletna vtičnica – izvorno "websocket"

5.4 Testiranje aplikacije

Pričakovano delovanje aplikacije smo potrdili z ročnim in avtomatskim testiranjem. Napisali smo avtomatske teste enot, gradnikov ter integracijske teste. Zaradi modularne zasnove aplikacije in odvisnosti od zunanjih storitev je bilo potrebno izdelati maketo[4] uporabljenih funkcij. Napisali smo 137 avtomatskih testov in dosegli 49,5-odstotno pokritost programske kode. Nismo dosegli visoke pokritosti programske kode, smo pa pokrili vse kritične dele aplikacije. Doseganje visoke pokritosti nam je tudi oteževalo problematično izdelovanje maket modulov ogrodja GetX [8] in knjižnice odjemalca MQTT. Delovanje aplikacije smo preverili tudi na različnih fizičnih napravah.

5.5 Delovni tok Github Actions CI/CD

Z uporabo Github Actions smo postavili delovni tok, ki se sproži ob vsaki objavi kode na Git veji main. Na začetku se požene virtualni računalnik Ubuntu 20.04 LTS in nastavi vse potrebne okoljske spremenljivke. Nanj se namestita Java in Flutter, ki ju bomo potrebovali za grajenje aplikacije. Iz Flutter repozitorija se naložijo vse knjižnice, ki jih aplikacija potrebuje. Ko je priprava okolja končana, se začne testiranje aplikacije. Na začetku se preveri format izvornih datotek in izvede statična analiza programske kode. Nadaljujemo z izvajanjem testov enot. Po prestanih testih se iz izvorne kode zgradita Android in spletna aplikacija. Namestitvene datoteke za Android aplikacijo se odložijo na Github, kjer so na voljo za prenos. Spletna aplikacija se postavi na Githubov spletni strežnik.

6 Ovrednotenje rezultatov

Razvili smo večplatformno aplikacijo za upravljanje NEO pametnega doma, ki omogoča pregled povezanih naprav. Mogoče jo je uporabljati na Android in iOS mobilnih platformah ter z uporabo spletnih brskalnikov.

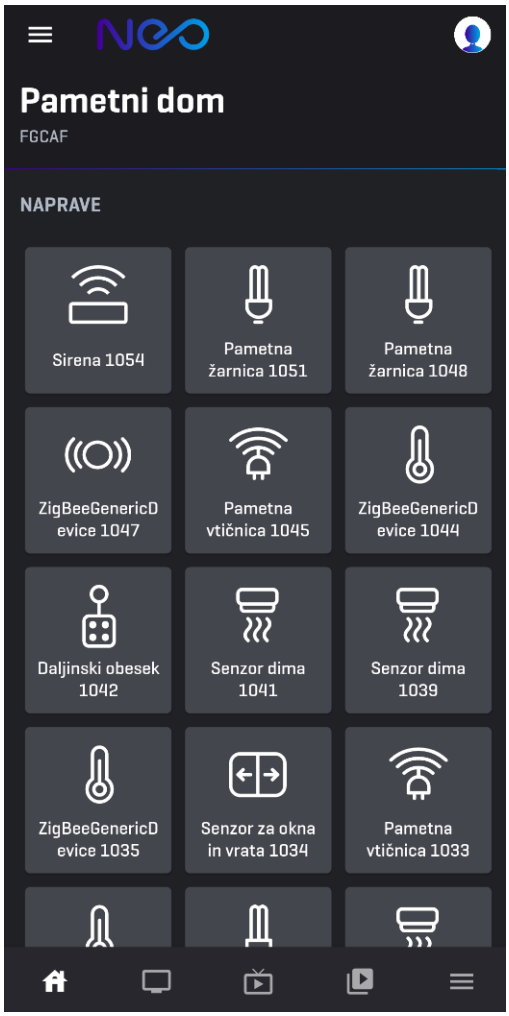
Ob zagonu aplikacije se prikaže začetni zaslon, ki je prikazan, dokler se v ozadju ne naloži vse potrebno. Po končanem nalaganju se prikaže zaslon za prijavo z obrazcem za uporabniške poverilnice. Prijava je sestavljena iz več obveznih zaporednih korakov. Prvi korak od uporabnika zahteva vpis uporabniškega imena in gesla. Če uporabnik nima uporabniškega računa, ga lahko ustvari s klikom na povezavo za registracijo. Uporabnik ima tudi možnost pridobiti tridesetdnevno promocijsko kodo, ki jo lahko uporabi za prijavo. Če uporabnik vpiše napačno uporabniško ime ali geslo, se prikaže rdeče obvestilo. Med čakanjem na odgovor zalednega sistema se prikaže okrogel kazalnik nalaganja.

Po opravljenem prvem koraku prijave uporabnik nadaljuje na naslednji korak. Drugi korak je izbira uporabniškega podračuna. Na zaslonu se prikaže seznam podračunov, ki so na voljo uporabniku. Večina uporabnikov ima samo en podračun. V tem primeru se izbira izvede samodejno, brez akcije uporabnika. Po izbiri uporabniškega podračuna uporabnik nadaljuje na naslednji korak. Tretji korak je izbira uporabniškega profila. Uporabnik izbere enega izmed ponujenih profilov in nadaljuje na naslednji korak. Zadnji korak prijave je izbira NEO Smartboxa, na katerega se povezujejo naprave pametnega doma. Če na tem koraku uporabnik izbere napačen NEO Smartbox, se naprave pametnega doma ne bodo pokazale.

Po uspešno opravljeni večstopenjski prijavi je uporabnik preusmerjen na glavno stran aplikacije. Prikaže se statična domača stran pametnega doma. Če ima uporabnik priključeno vsaj eno napravo, se statična stran zamenja z dinamičnim prikazom povezanih naprav, kot je vidno na sliki 6. Če je naprav več, kot jih lahko prikažemo na zaslonu, se prikaže drsnik, ki omogoča prikaz ostalih naprav. Postavitev spletne različice aplikacije se malenkost razlikuje, saj se zaradi drugačnega razmerja stranic zaslona lahko prikaže več naprav. Za vsako napravo se kreira

[4] maketa - izvorno "mockup"

nova kartica, na kateri sta ime naprave in ikona, ki se nastavi glede na vrsto naprave. Uporabnik lahko s klikom na sliko profila odpre stran za pregled profila. Tam lahko poleg pregleda izbranega podračuna, profila in NEO Smartboxa opravi tudi odjavo iz aplikacije.



Slika 6: Zaslon za prikaz naprav pametnega doma mobilne različice.
Vir: [9].

Razvita aplikacija deluje dobro na vseh ciljnih platformah. Uporabniški vmesnik je konsistenten in uporabniška izkušnja je dobra. Vse naprave NEO pametnega doma so podprte na vseh platformah. Aplikacije se razlikujejo v delovanju protokola MQTT, ker spletna različica za povezovanje uporablja spletne vtičnice. Razlikujejo se tudi v število prikazanih naprav, saj lahko spletna različica zaradi drugačnega razmerja stranic zaslona prikaže več naprav.

7 Zaključek

Med razvojem smo dobro spoznali orodje Flutter in njegove prednosti ter slabosti. Ugotovili smo, da je zelo dobro orodje za razvoj večplatformnih aplikacij, ki ima konstantno in urejeno sintakso ter ne zahteva pisanja odvečne kode. Flutter ponuja nesporne prednosti, kot sta hitrejši razvoj in prihranek stroškov, te prednosti pa lahko pomembno vplivajo na proces razvoja, ko gre za predvideno uporabo sredstev in hitrejše trženje izdelka. Hitrejši razvoj omogoča hitrejšo potrditev ideje, zgodnje testiranje uporabnikov in prihranek denarja. Flutter spreminja način razvoja aplikacij. Pod isto streho združuje mobilne, spletne in namizne aplikacije. Razvijalcem omogoča, da se osredotočajo na sam razvoj funkcionalnosti in ne zapravljajo časa za podpiranje vsake platforme posebej. Je

visoko zmogljivo in visoko produktivno orodje, ki skrajša razvojne cikle in omogoča enkratno pisanje izvorne kode za več platform. Hiter ponovni zagon pohitri razvoj in razhroščevanje aplikacij.

Zastavljene cilje smo uspešno izvedli. Dokazali smo, da je mogoče razviti večplatformno aplikacijo NEO. Razvili smo kakovostno in robustno aplikacijo, ki uporabniku omogoča pregled nad napravami NEO pametnega doma. Podprli smo veliko različnih naprav, saj aplikacija podpira vse naprave, ki so združljive z NEO pametnim domom.

Poseben izziv pri izdelavi aplikacije je bila implementacija komunikacije MQTT, saj platforma OBLO nima podpore za Flutter oziroma Dart. Uradni razvojni paketi obstajajo samo za Android, iOS in spletne aplikacije. Potrebno je bilo implementirati vmesnike MQTT, saj še niso obstajali. Pri tem se nismo mogli zanašati na obstoječo dokumentacijo, ker ta način uporabe ni bil podprt.

Možnosti za izboljšave je veliko. Aplikacija trenutno nima posebne uporabne vrednosti, saj omogoča le pregled naprav. Vnaprej smo sicer pripravili vmesnike tudi za ostala sporočila MQTT, vendar jih nismo uporabili. Na uporabniškem vmesniku pa bi bilo potrebno dodati upravljanje, dodajanje in brisanje naprav.

Literatura

[1] gs.statcounter.com/os-market-share, Operating System Market Share Worldwide - March 2022, obiskano 17. 7. 2022.

[2] neo.io/info, NEO, obiskano 16. 7. 2022.

[3] www.obloliving.com, OBLO, obiskano 16. 7. 2022.

[4] codenameone.com/blog/top-10-best-cross-platform-app-development-frameworks-in-2022.html, Top 10 Best Cross Platform App Development Frameworks in 2022, obiskano 16. 7. 2022.

[5] HUBER Stefan, DEMETZ Lukas Performance Analysis of Mobile Cross-platform Development Approaches based on Typical UI Interactions, julij 2019.

[6] statista.com/statistics/869224/worldwide-software-developer-working-hours/, Cross-platform mobile frameworks used by software developers worldwide from 2019 to 2021, obiskano 16. 7. 2022.

[7] medium.com/flutter/introducing-flutter-3-5eb69151622f, Introducing Flutter 3, obiskano 16. 7. 2022.

[8] pub.dev/packages/get, GetX, obiskano 16. 7. 2022.

[9] MEVLJA, S. 2022. Razvoj večplatformne aplikacije za NEO pametni dom. Univerza v Ljubljani, Fakulteta za računalništvo in informatiko.