Operativni sustavi pametnih satova

Maglica, David

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:800524

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-05**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli
Fakultet Informatike u Puli

DAVID MAGLICA

OPERATIVNI SUSTAVI PAMETNIH SATOVA

ZAVRŠNI RAD

Pula, rujan, 2022.godine

Sveučilište Jurja Dobrile u Puli
Fakultet Informatike u Puli

DAVID MAGLICA

OPERATIVNI SUSTAVI PAMETNIH SATOVA

ZAVRŠNI RAD

JMBAG: 0066306616, redoviti student
Studijski smjer: Informatika
Kolegij: Informacijska tehnologija i društvo

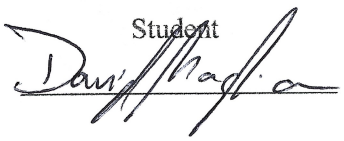
Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijske i komunikacijske znanosti
Znanstvena grana: Informacijski sustavi
Mentor: doc. dr. sc. Snježana Babić

Pula, rujan, 2022.godine

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani DAVID MAGLICA, kandidat za prvostupnika INFORMATIKE ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljeni način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student



U Puli 21.09.2022.

IZJAVA O KORIŠTENJU AUTORSKOG DJELA

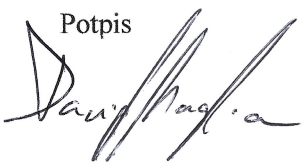
Ja, DAVID MAGLICA dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj Završni rad pod nazivom OPERATIVNI SUSTAVI PAMETNIH SATOVA

koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, 21.09.2022.

Potpis



Sadržaj

1. Uvod 1
2. Operativni sustavi pametnih satova 2
   2.1. Opće karakteristike 3
3. WatchOs 4
   3.1. Verzije watchOS-a 4
      3.1.1. watchOS 1 4
      3.1.2. watchOS 2 5
      3.1.3. watchOS 3 6
      3.1.4. watchOS 4 6
      3.1.5. watchOS 5 7
      3.1.6. watchOS 6 7
      3.1.7. watchOS 7 8
      3.1.8. watchOS 8 8
   3.2. Xamarin.iOS development 9
      3.2.1. Struktura projekta 9
      3.2.2. Životni ciklus aplikacije 11
      3.2.3. Izvođenje koda na main threadu 12
      3.2.4. Interakcije 14
      3.2.5. Kontrole korisničkog sučelja watchOS u Xamarinu 15
      3.2.6. Veličina zaslona 19
      3.2.7. Ograničenja watchOS-a 20
4. Wear OS 21
   4.1. Wear OS development. 22
      4.1.1. Korisničko sučelje Wear OS-a 22
      4.1.2. Obavijesti na Wear OS-u 25
      4.1.3. User Input 27
5. Usporedba watchOS-a i Wear OS-a obzirom na funkcionalnosti 32
6. Budući trendovi pametnih satova 36
7. Rezultati provedenog kratkog istraživanja mišjenja građana o operativnim sustavima pametnih satova 38
   7.1. Metodologija istraživanja 38
   7.2. Sudionici istraživanja i njihove karakteristike 38
   7.3. Rezultati istraživanja 39
      7.3.1. Rezultati ispitanika o posjedovanju pametnog sata 39

7.3.2. Rezultati nekorisnika pametnih satova 40
7.3.3. Rezultati korisnika pametnih satova 41
8. Zaključak 47

1. Uvod

Pametni satovi digitalni su satovi koji imaju puno veću korisnost nego samo praćenje vremena i po funkcionalnostima bi ih mogli usporediti s pametnim telefonima. Rani modeli mogli su samo izvoditi neke matematičke račune i igrati igre, ali danas su oni „nosivo računalo“. Dolaze u svim veličinama i oblicima te su se kroz godine mnogo mijenjali (CoverMagazin, 2021).

Cilj ovog rada jest upoznati se s pojmom operativnog sustava pametnih satova, funkcionalnostima tih operativnih sustava i usporediti dva operativna sustava: watchOS i Wear OS. Rad se sastoji od teorijskog dijela i kratkog istraživanja mišljenja građana o operativnim sustavima pametnih satova. U teorijskom dijelu objašnjen je pojam pametnog sata i operativnog sustava pametnog sata, provedena je analiza operativnih sustava WatchOS i Wear OS, prikazani su prednosti i nedostatci ta dva operativna sustava, odrađena je usporedba WatchOS-a i Wear OS-a obzirom na funkcionalnosti te je odrađen kratki pregled budućih trendova pametnih satova.

U provedenom istraživanju sudjelovalo je ukupno 102 ispitanika korisnika i nekorisnika pametnih satova. Rezultati su pokazali da su korisnici većinski zadovoljni funkcionalnostima operativnih sistema pametnih satova i da im je u potpunosti jednostavno nadograditi operativni sistem novim značajkama. Većina korisnika voljela bi da se dodaju mogućnost PDF readera kako bi mogli čitati PDF dokumente na pametnom satu i mogućnost uvida u bankarske aplikacije, a ne samo NFC plaćanje. Rezultati ankete također pokazuju da korisnici najviše koriste svoje pametne satove za praćenje i zabilježavanje treninga i drugih aktivnosti. Rezultati ankete mogu pomoći nekorisnicima pametnih satova pri odabiru sata uvidom u zadovoljstvo trenutnih korisnika, a za proizvođače pametnih satova, anketa može služiti kao pokazatelj za dodavanje novih značajki koje korisnici smatraju korisnima kao npr. čitač PDF dokumenata i uvid u bankarske aplikacije.

2. Operativni sustavi pametnih satova

Operativni sustav je zapravo skup osnovnih programski sastavnica koje služe kao potpora raznovrsnim primjenskim programima, te omogućuje provedbu radnih operacija. Također automatski obavlja mnoge operacije koje su korisniku nevidljive (Hrvatska Enciklopedija, 2021).

Uz mnoštvo pametnih uređaja i raznolikost njihovih fizičkih veličina, postoje dva glavna načina rada: neki nosivi uređaji rade u samostalnom načinu rada, dok drugi koriste bežičnu vezu s uređajem spojenim na Internet, obično pametnim telefonom. Većina pametnih satova, na primjer, povezuje se s pametnim telefonom korisnika kako bi omogućila njihovo web sučelje. U tom slučaju operativni sustav je u biti prilagođen da omogući satu da služi kao korisničko sučelje, a sam sat je zapravo bežični *Bluetooth* adapter koji proširuje mogućnosti pametnog telefona na sat, dopuštajući korisniku da odgovara i započne telefonske pozive, čita i diktira e-poštu i poruke, dobiva vremensku prognozu ili sportske rezultate, sluša glazbu ili pita digitalnog asistenta pitanje. Samostalni pametni sat, s druge strane, obično je posvećen određenoj namjeni, kao što je *GPS* navigacija, praćenje, snimanje i prijenos podataka vezanih uz zdravstvo (TechTarget, 2015).

Neki od poznatijih operativnih sustava za pametne satove su: Googleov Wear OS, Samsungov Tizen, Appleov WatchOS, Garmin OS, Fitbit OS i Huawei Lite OS (Watchranker, n.d.).

2.1. Opće karakteristike

Najvažnija komponenta operativnih sustava pametnih satova je *kernel*. Nalazi se između softvera i hardvera i određuje kako hardver rješava različite softverske zadatke, upravljanje napajanjem i još mnogo toga.

Svaki *kernel* radi s određenim rasponom hardvera. Na primjer, Googleov Wear OS radi s istim hardverom kao Linux – to je zato što Wear OS „krade“ njegovu hardversku kompatibilnost iz Linuxovog *kernela*. Postoji mnogo hardverskih komponenti u pametnim satovima, a neki od najčešćih su (Yamada, 2014):

- ***MIPS***

MIPS (Microprocessors without Interlocked Pipeline Stages) su: vlastiti skup instrukcija i serija procesora. *Imagination Technologies* je kupio prava na *MIPS* i od tada ga je razvio u *Warrior 64-bitni processor.* U sustavima koji koriste Javu, kao što je Android, *MIPS* procesori bi trebali pokretati većinu aplikacija. Izvorno kodiranje aplikacije neće raditi s *MIPS* čipsetovima. *MIPS* bi zapravo bio vrlo učinkovita platforma za pametne satove pošto sadrži vrlo moćnu (barem za pametne satove) grafiku, 64-bitnu podršku i „anoreksičnu“ potrošnju energije.

- ***ARM Cortex M***

ARM Cortex M je procesor koji se bazira na *RISC*-u (*Reduced Instruction Set Computer*. Cortex M serija nudi čipset ultra niske potrošnje za jeftine prijenosne uređaje. Nedavni napredak ARM Holdingsa rezultirao je serijom M7, koja nudi i dobre performanse i vrlo nisku potrošnju energije.

- ***ARM Cortex A***

Cortex A serija se više fokusira na grafiku i CPU snagu u usporedbi s Cortex M serijom. Kao nosiva platforma, Cortex A je sklon preopterećenju, unatoč očitoj neusklađenosti, proizvođači nastavljaju proizvoditi pametne satove s niskim vijekom trajanja baterije opremljene Cortexom A.

- ***x86***

Intelova x86 platforma, za razliku od ostalih procesora baziranih na RISC-u ovisi o onome što se zove „x86“ skup instrukcija. Iako se ne koristi u mnogim prijenosnim uređajima, ubrzo bi trebao postati novi standard (Yamada, 2014).

3. WatchOs

WatchOS je operativni sustav kojeg je Apple stvorio kako bi ga koristio za svoj Apple Watch. Predstavljen je 2015. uz prvi Apple Watch i zamišljen je da bude kao iPhone na manjem ekranu pošto dijeli mnogo sličnih značajki iOS-u za iPhone i iPad, kao što su na primjer Siri, Apple Maps, Wallet i sl. (Kansal, 2022).

Sučelje je bio novitet za Apple s početnim zaslonom koji sadrži okrugle ikone za aplikacije, koje AppleWatch može pokrenuti i novim gumbom *Digital Crown,* a on se može rotirati i pritiskati, kao tradicionalna kruna sata. Operativni sustav dopušta korisniku da se pomoću krune pomiče kroz liste stavki i da uveća ili smanji početni zaslon (LeFebvre, 2021).

3.1. Verzije watchOS-a

3.1.1. watchOS 1

WatchOS 1 sadržavao je početni zaslon, prateće aplikacije i *Glances* pogled. *Glances* omogućuje korisniku da vidi podatke iz aplikacija koje on sam odredi i bio je poput malog skupa *widgeta* kojima bi korisnik mogao povući prstom prema gore s dna Apple Watcha da bi im pristupio.

Pritiskom na *Digital Crown* otvorio bi se izbornik „Prijatelji“ gdje bi korisnik vidio osobe koje je označio kao „prijatelje“ Apple Watcha.

I *Siri* i *Force Touch* su bili dostupni od prvoga dana te se i dalje koriste dan danas. Kako bi se pozvala Siri bilo je potrebno pritisnuti i držati *Digital Crown* par sekundi ili (ako aktivirano) pozvati ju sa „Hey Siri“. WatchOS 1 također je pružao mogućnost korisniku da provjeri kalendar, uključi način rada u zrakoplovu, koristi Apple Pay, telefonira, iako za to je potrebno imati vlastiti iPhone u blizini te koristiti aplikacije poput *Activity*, koja prati kretanje, treninge i razdoblja stajanja tijekom dana (LeFebvre, 2021).

Slika 1 – početni zaslon Apple Watcha sa okruglim ikonama, preuzeto sa: https://www.mac-tech.com/2015/05/19/watch-os-1-0-1-released-for-the-apple-watch/

3.1.2. watchOS 2

Druga iteracija watchOS-a (watchOS 2.0) uključivala je podršku za takozvane „third-party" aplikacije koje su se mogle koristiti na Apple Watchu bez potrebe da je uređaj uparen s iPhoneom korisnika. Tako se konačno mogao koristiti Facebook Messenger za slanje tekstualnih poruka, audio datoteka i čak za dijeljenje lokacije izravno s Apple Watcha.

WatchOS 2.0 donio je i nove mogućnosti poput značajke putovanja kroz vrijeme koja korisniku omogućuje pregled do 72 sata unaprijed i unazad u vrijeme za aplikacije poput *Weather* i *News*. Također, novim OS-om lakše je odgovoriti na e-poštu i poboljšana je aplikacija *Music*. Uz sve to, uveden je novi popularni način rada „*nightstand mode"* (slika 2), koji dopušta korisniku da postave uređaj na bok kako bi prikazali na minimalistički način vrijeme i alarm te se kruna mogla koristiti kao gumb za odgodu alarma (LeFebvre, 2021).

Slika *2* - *nightstand mode*, preuzeto sa: https://9to5mac.com/2018/05/23/apple-watch-how-to-disable-night-stand-mode/

3.1.3. watchOS 3

WatchOS 3 je donio sa sobom poboljšane performanse, nova lica brojčanika i još više originalnih aplikacija te je uveden kontrolni centar koji bi se aktivirao povlačenjem prema gore po zaslonu. Uz to promijenili su se i neki elementi sučelja poput funkcije krune. Sada kruna više nije otvarala listu prijatelja već takozvani *Dock* kroz kojeg se može kretati rotirajući krunu (slika 3) (LeFebvre, 2021).

Slika *3* - Dock, preuzeto sa: https://support.apple.com/en-us/HT206992

Pojavila se također nova aplikacija *Breathe*, koja je dizajnirana za pomoć pri meditaciji i smanjenju stresa (Hattersley, 2016.). Apple je uveo i značajku *SOS* (koja može obavijestiti odabrane kontakte i nazvati hitnu), *Reminders*, *Home*, *Find My Friends*, sustav za otkrivanje otkucaja srca i *Scribble* s kojim se mogu pisati poruke slovo po slovo (LeFebvre, 2021).

3.1.4. watchOS 4

U novoj iteraciji watchOS-a *Apple Music* je dobio veliki poticaj s lakšim načinom sinkronizacije glazbe s iPhonea i opcijom streaminga. Još jedna korisna značajka bila je mogućnost svjetiljke, koja postavlja svijetlu boju na brojčaniku sata kako bi korisnik lakše vidio u mraku. Dodana je i nova opcija popisa za početni zaslon koja omogućuje prikaz svih aplikacija u sučelju popisa i opcija favorita za *Dock*, tako da se može odabrati što će se prikazati kada se pritisne kruna (prije je *Dock* prikazivao samo najnovije aplikacije) (LeFebvre, 2021).

3.1.5. watchOS 5

Dolaskom *watchOS-a* 5 dodane su nove značajke i aplikacije, *Sirijeve* sposobnosti su nadograđene i obavijesti su poboljšane. Aplikacija *Workouts* dobila je novi *Compete with Friends* sustav, auto detekciju za treninge i joga i planinarenje su dodane na popis dostupnih praćenja. Uvedene su i aplikacije *Podcasts* za *watchOS* i *Walkie-Talkie* (slika 4) (LeFebvre, 2021).

Slika *4* - Walkie-Talkie aplikacija za Apple Watch, preuzeto sa: https://support.apple.com/en-us/HT208917

Kao već prije navedeno, *Siri* je poboljšana, sada omogućuje podršku za *third-party* aplikacije i može se postaviti da samo dizanje zapešća aktivira *Siri*. Što se obavijesti tiče, one su sada grupirane po aplikacijama i watchOS 5 dopušta još više radnji za upravljanje njima. Ako je korisnik student, watchOS 5 nudi podršku za beskontaktne studentske iskaznice, omogućujući pristup domu i drugim dijelovima kampusa ukoliko je potrebna iskaznica za ulaz (LeFebvre, 2021).

3.1.6. watchOS 6

Šesta iteracija *watchOS*-a donijela je *AppStore* namijenjen isključivo za watchOS, nove aplikacije poput kalkulatora i *Voice Memo*, značajke *Activity trends* i još značajke usmjerenih na zdravlje kao što su praćenje menstrualnog ciklusa i aplikacija za zdravlje sluha. *Siri* je dobila novo poboljšanje i sada može reći koja pjesma svira u blizini, aplikacija *Reminders* je redizajnirana i audio knjige se mogu automatski sinkronizirati (LeFebvre, 2021).

3.1.7. watchOS 7

WatchOS 7 je isporučen s nekoliko novih značajki, uključujući *Family Setup*, praćenje spavanja, automatsko detektiranje pranja ruku, novu aplikaciju *Sleep*, nove vježbe i poboljšanja za *Maps* i *Siri*.

Family Setup omogućuje korištenje *Apple Watcha* osobama koje nemaju *iPhone*, naravno pod uvjetom da netko u obitelji posjeduje *iPhone*. *Apple* ovom značajkom zapravo cilja na djecu i starije. Moguće je putem *Family Setupa* i uključiti roditeljski nadzor i aktivnosti osmišljene za djecu. Konačno, nastupa i Apple Cash Family koji omogućuje slanje novca članovima obitelji i to je moguće i putem *Apple Watcha*.

Aplikacija *Sleep* prati i prikazuje korisnikove obrasce spavanja i nudi značajke za odmaranje i buđenje. Sada *Apple Watch* automatski prelazi u način *Do Not Disturb* ako je konfiguriran.

Najbitnije ažuriranje za watchOS 7 je bila verzija 7.0.2 koja je predstavljena sredinom listopada 2020. s kojom se riješio problem pretjeranog trošenja baterije koji se nalazio u originalnoj inačici operativnog sustava (LeFebvre, 2021).

3.1.8. watchOS 8

Osmo izdanje watchOS-a uključuje ažuriranje svojih algoritama za detekciju pada i preobrazbu svoje *wellness* aplikacije. Prethodno zvana *Breathe*, nova aplikacija *Mindfulness* uključuje novu vrstu sesije i vođene meditacije. Unaprijeđene su i funkcije spavanja, uvedeno je mjerenje brzine disanja tako da se mjeri broji udisaja u minuti. Informacije o snu dostupne su u aplikaciji *Health* na *iPhoneu*.

Aplikacija *Wallet* je također dobila nekoliko impresivnih ažuriranja, podržava ultra širokopojasnu vezu za digitalne ključeve automobila, kućne ključeve, korporativne značke pa čak i hotelske ključeve.

Konačno, watchOS 8 nudi potpuno redizajniranu aplikaciju *Home*, novi *Scribble* alat, diktiranje i Emojije za poruke i značajku filtriranja obavijesti pod nazivom *Focus,* koja je dizajnirana da smanji ometanja (LeFebvre, 2021).

3.2. Xamarin.iOS development

Xamarin.iOS omogućuje programerima da kreiraju izvorne iOS aplikacije koristeći iste kontrole korisničkog sučelja koje su dostupne u *Objective-C* i *Xcode*, ali uz dodanu fleksibilnost i elegantnost modernog jezika (C#), snagom .NET Base Class Library (BCL) te dva prvoklasna IDE-a, Visual Studio za Mac i Visual Studio (Ortinau i ostali, 2021).

Kako bi mogli objasniti kako je postavljen projekt u Visual Studiju za Xamarin.iOS moramo prvo objasniti što je projekt, a što je rješenje u Visual Studiju.

Kada se izrađuje aplikacija ili web aplikacija u Visual Studiju, počinje se projektom. Projekt sadrži sve datoteke koje su sastavljene u izvršnu datoteku, biblioteku ili web stranicu. Te datoteke mogu uključivati izvorni kod, ikone, slike, podatkovne datoteke i tako dalje. Projekt također sadrži i postavke prevoditelja i druge konfiguracijske datoteke koje bi mogli biti potrebne raznim uslugama ili komponentama s kojima program komunicira.

Rješenje, unatoč imenu, nije „odgovor“ na ikakvo pitanje, već samo spremnik za jedan ili više povezanih projekata, zajedno s informacijama o izgradnji i svim raznim datotekama koje nisu povezane s određenim projektom (Lee, 2022).

3.2.1. Struktura projekta

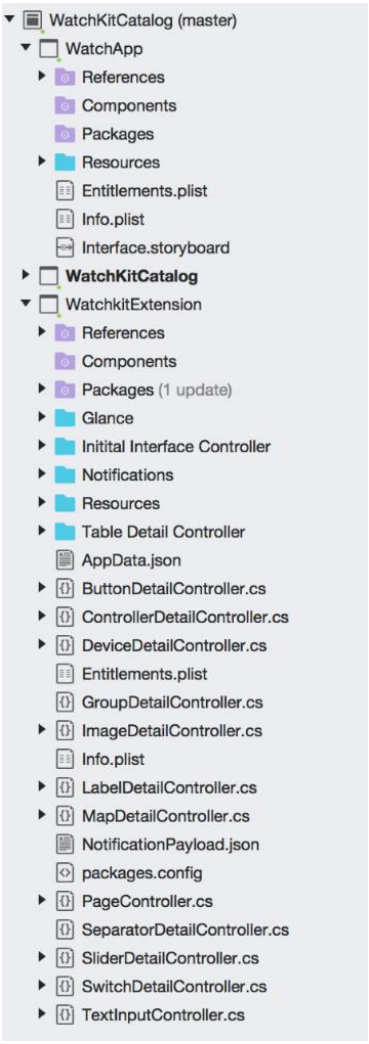
Svako rješenje za watchOS aplikaciju sadrži tri projekta:

- *Watch Extension* – projekt koji sadrži kod za aplikaciju za sat.
- *Watch App* – sadrži resurse i *storyboard* korisničkog sučelja.
- *iOS Parent App* – ovo je obična aplikacija za iPhone. Aplikacija za sat i proširenje u paketu su u iPhone aplikaciji za isporuku na korisnikov sat.

U aplikacijama za watchOS 1, kod u proširenju radi na iPhoneu, Apple Watch je zapravo vanjski zaslon. Od watchOS-a 2 nadalje, aplikacije rade u potpunosti na Apple Watchu (slika 5) (Ortinau i ostali, 2021).

Slika *5* - razlika u radu aplikacija u watchOS 1 i 2, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/get-started/intro-to-watchos-images/arch.png#lightbox

Bez obzira na to koja je verzija watchOS-a ciljana, u Visual Studiju, cjelovito će rješenje izgledati otprilike ovako:



Slika *6* - struktura rješenja za watchOS aplikaciju, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/get-started/intro-to-watchos-images/projectstructure.png#lightbox

3.2.2. Životni ciklus aplikacije

U *Watch Extension*, za svaku scenu *Storyboarda* kreira se podklasa klase WKInterfaceController. Klase WKInterfaceController slične su objektima UIViewController u iOS programiranju, ali nemaju istu razinu pristupa. Na primjer, ne može se dinamički dodavati kontrole ili restrukturirati korisničko sučelje, ali mogu se sakriti i otkriti kontrole, a nekim kontrolama se može i promijeniti veličinu, transparentnost i opcije izgleda.

Životni ciklus WKInterfaceController objekta uključuje sljedeće pozive:

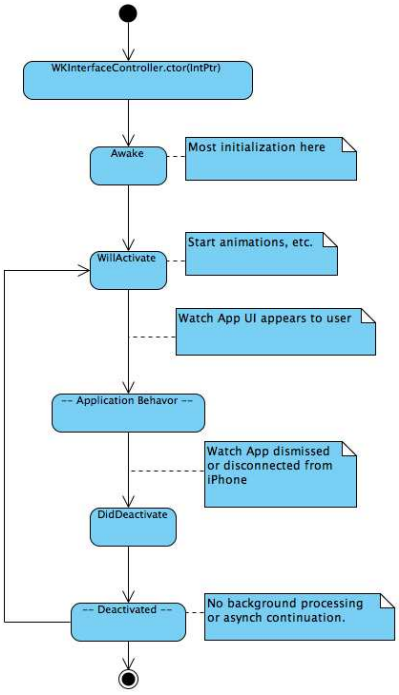
- *Awake* : većina inicijalizacije bi trebalo izvesti ovom metodom.
- *WillActivate* : poziva se neposredno prije nego što se *Watch App* pojavi korisniku. Ova metoda se koristi za izvođenje inicijalizacije u posljednjem trenutku, pokretanje animacija, itd.

Nakon *WillActivate*, *Watch Extension* je pokrenut i počinje reagirati na korisnički unos i ažurirati prikaz aplikacije prema napisanoj logici.

- *DidDeactivate* : Nakon što korisnik prestane koristiti *Watch Extension* poziva se ova metoda. Nakon *returna* ove metode, kontrole korisničkog sučelja ne mogu se mijenjati dok se ponovo ne pozove *WillActivate*.

  Ova metoda će se također pozvati ako je veza s iPhoneom prekinuta.

Nakon što je *Watch Extension* deaktiviran, nije više dostupan programu. Asinkrone funkcije na čekanju neće biti pozvane i *Watch Kit Extensions* neće moći koristiti pozadinske načine obrade. Ako korisnik ponovo aktivira program, ali OS nije prekinuo aplikaciju, prva pozvana metoda bit će *WillActivate* (Ortinau i ostali, 2021).



Slika 7 - životni ciklus WKInterfaceController klase, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/get-started/intro-to-watchos-images/wkinterfacecontrollerlifecycle.png

3.2.3. Izvođenje koda na main threadu

Većina operativnih sustava, uključujući iOS, Android i Windows, koristi jednodretveni (*single-threading*) model za kod koji uključuje korisničko sučelje. Taj model je neophodan za ispravnu serijalizaciju događaja korisničkog sučelja, uključujući pritiske gumbova ili tipki i unosa dodirom. Ta se dretva često naziva i *main thread* ili *UI thread*. Nedostatak ovog modela je što se sav kod koji pristupa elementima korisničkog sučelja mora izvoditi *na main threadu* aplikacije.

Klasa MainThread omogućuje aplikacijama pokretanje koda na *main threadu* izvršenja i također utvrđuje ako se trenutno određeni blok koda izvodi na *main threadu.*

Aplikacije ponekad moraju koristiti evente koji pozivaju *event handlera* na sekundarnoj dretvi izvođenja (Xamarin.Essentials klase Accelerometer, Compass, Gyroscope, Magnetometer i OrientationSensors mogu vratiti informacije o sekundarnoj dretvi kada se koriste s većim brzinama). Ako *event handler* treba pristupiti elementima korisničkoga sučelja, mora pokrenuti taj kod na *main threadu* tako da koristi klasu MainThread.

Kako bi kod pokrenuo na *main threadu* prvo treba dodati referencu na Xamarin.Essentials u klasu.

```
using Xamarin.Essentials;
```

Slika 8 - referenca na Xamarin.Essentials, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/essentials/main-thread?context=xamarin%2Fwatchos

Sljedeće mora se pozvati statička metoda MainThread.BeginInvokeOnMainThread. Argument koji se prosljeđuje je *Action* objekt, koji je zapravo metoda bez argumenata i bez povratne vrijednosti.

```
MainThread.BeginInvokeOnMainThread(() =>
{
    // Code to run on the main thread
});
```

Slika 9 - pozivanje metode MainThread.BeginInvokeOnMainThread, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/essentials/main-thread?context=xamarin%2Fwatchos

Također, moguće je i definirati zasebnu metodu za kod koji se mora izvoditi na *main threadu*.

```
void MyMainThreadCode()
{
    // Code to run on the main thread
}
```

Slika 10 - zasebna metoda za izvođenje koda na *main threadu*, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/essentials/main-thread?context=xamarin%2Fwatchos

Zatim se može pokrenuti metoda na *main threadu* tako da se referencira u BeginInvokeOnMainThread.

Klasa MainThread također omogućuje aplikaciji da odredi izvodi li se određeni blok koda na *main threadu*. Svojstvo IsMainThread vraća True ako se kod koji poziva svojstvo izvodi na *main threadu.* Program može koristiti IsMainThread za pokretanje različitog koda za glavnu ili sekundarnu dretvu.

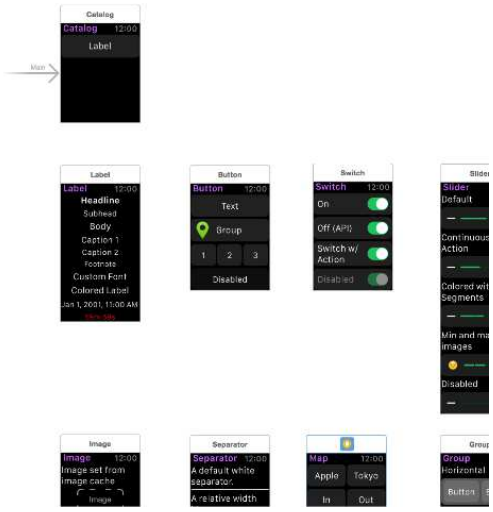
```
if (MainThread.IsMainThread)
{
    // Code to run if this is the main thread
}
else
{
    // Code to run if this is a secondary thread
}
```

Slika 11 - primjer korištenja IsMainThread, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/essentials/main-thread?context=xamarin%2Fwatchos

Implementacije od BeginInvokeOnMainThread same provjeravaju je li poziv napravljen na *main threadu* tako da postoji vrlo mala performansna kazna ako se pozove kada to nije potrebno (Montemagno i ostali, 2021).

3.2.4. Interakcije

Većina interakcija između *Watch App i Watch Extension* odvijat će se pomoću podklasa u WKInterfaceController, koje se pišu tako da odgovaraju scenama u Interface.storyboard. U sljedećoj slici (slika 10) prikazan je dio Storyboard uzorka Watch Kit Catalog.



Slika 12 - Watch Kit Catalog Storyboard, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/get-started/intro-to-watchos-images/scenes.png

Za svaku prikazanu scenu postoji odgovarajući prilagođeni *WKInterfaceController* (*LabelDetailController, ButtonDetailController, SwitchDetailController,* itd.) u *Watch Extensionu* (Ortinau i ostali, 2021).

3.2.5. Kontrole korisničkog sučelja watchOS u Xamarinu

Programski nazivi svih kontrola imaju prefiks WKInterface kao npr. WKInterfaceLabel, WKInterfaceButton itd.

Postoji petnaest kontrola koje možemo koristiti na svakom Storyboardu: *Label*, Button, *Switch*, *Slider*, *Image*, *Separator*, *Map*, *Movie* & *InlineMove*, *Group*, *Table*, *Device*, *Menu*, *Text Input*, *Digital Crown* i *Gestures*.

U nastavku će nekoliko njih biti opisano.

- **Image**

WatchOS pruža kontrolu zvanu WKInterfaceImage za prikaz slika i jednostavnih animacija. Neke kontrole također mogu imati i pozadinsku sliku.

Najučinkovitiji način za prikaz slika uključiti je ih u *Watch App* projekt i prikazati ih pomoću metode SetImage(imageName). Isti se postupak primjenjuje za metodu SetBackgroundImage(imageName) za klase *Button, Group* i *InterfaceController*. Najbolji učinak se postiže pohranjivanjem slika u sam *Watch App*.

```
myImageControl.SetImage("Whale");
myOtherImageControl.SetImage("Worry");
```

Slika 13 - primjer prikaza slike "Whale" iz *Watch App* projekta, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/image

Osim učitavanja slika koje su pohranjene u samom *Watch Appu*, one se mogu slati i iz *extension budlea* u *Watch App* za naknadni prikaz. Za učitavanje slika iz *Watch Extensiona*, mora se stvoriti instance od *UIImage* nakon čega se poziva *SetImage* s objektom *UIImage.*

```
using (var image = UIImage.FromBundle ("Bumblebee")) {
    myImageControl.SetImage (image);
}
```

Slika 14 – primjer prikaza slike "Bumblebee" iz *WatchExtensiona,* preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/image

Za animiranje skupa slika, sve bi trebale početi s istim prefiksom i imati numerički sufiks, dakle npr. Bus1, Bus2, Bus3, itd.

Za prikaz animacije potrebno je prvo učitati sliku koristeći SetImage s imenom prefiksa, a zatim pozvati StartAnimating. Za zaustavljanje petlje animacije poziva se StopAnimating.

```
animatedImage.SetImage ("Bus");
animatedImage.StartAnimating ();

animatedImage.StopAnimating ();
```

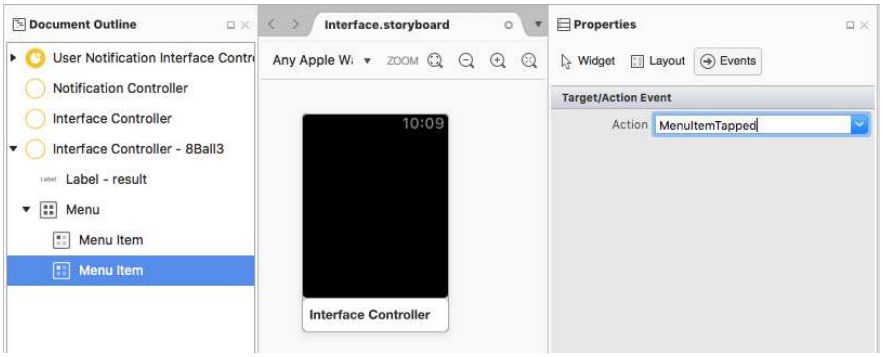
Slika 15- primjer animiranja skupa slika "Bus", preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/image

Pošto je *cache* veličine otprilike 20 MB, potrebno je moći efikasno se koristiti njome. Ona se puni tijekom pokretanja aplikacije, a kada se napuni do kraja, naša je odgovornost da izbrišemo datoteke pomoću jednostavnih metoda RemoveCachedImage ili RemoveAllCachedImages na objektu WKInterfaceDevice.CurrentDevice.

- **Menu**

Ukoliko je za *interface controller* implementiran *Menu*, kada korisnik izvršni takozvani *Force Touch*, izbornik će se prikazati. Kada nijedan izbornik nije implementiran zaslon se nakratko animira i ne događa se druga radnja.

Force Touches nisu povezani ni sa kojim posebnim elementom na zaslonu i samo jedan izbornik može se priključiti na *interface controller* te on će se pojaviti bez obzira na to gdje se na zaslonu dogodi *Force Touch*. *Menu* se treba dodati u *InterfaceController* na *storyboardu* za vrijeme dizajna. Kada se kontrola izbornika povuče na *interface controller*, nema vizualne indikacije na pregledu *storyboarda*, ali se izbornik pojavljuje na *Document Outline padu.*



Slika 16 - dodavanje izbornika u Document Outline padu, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/menu-images/menu-action.png

U izbornik se mogu dodati do četiri stavke koje se onda mogu konfigurirati u *Properties padu* te im se mogu postaviti sljedeći atributi: *Title, Custom Image* ili *system image (Accept, Add, Block, Decline, Info, Maybe, More, Mute, Pause, Play, Repeat, Resume, Share, Shuffle, Speaker, Trash).*

Izbornik se ne može dodati u *interface controller* tijekom izvođenja, iako se zbirka *MenuItems* može izmijeniti metodom *AddMenuItem(),* dok pozivom metode *ClearAllMenuItems* mogu se ukloniti sve programski dodane stavke izbornika.

```
AddMenuItem (WKMenuItemIcon.Accept, "Yes", new ObjCRuntime.Selector ("tapped"));
```

Slika 17 - primjer korištenja metode AddMenuItem(), preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/menu

- **Text Input**

Apple Watch sam po sebi ne nudi tipkovnicu za unos teksta, ali podržava neke alternativne načine unosa kao što su: Odabir unaprijed definiranog popisa tekstualnih opcija, *Siri*, odabir Emojija i *Scribble* (prepoznavanje rukopisa slovo po slovo).

Slika 18 - primjer unosa teksta, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/text-input-images/textinput-sml.png

Da bi se prihvatio unos teksta u *Watch App,* mora se prvo napraviti *array stringova* s unaprijed definiranim opcijama, da bi se mogao onda pozvati *PresentTextInputController* koji će kao argument prihvatiti kreirani *string.* Zatim se pomoću *WKTextInputMode* enumeracijom odlučuje dopušta li se korisniku korištenje Emojija (*Plain, AllowEmoji, AllowAnimatedEmoji*).

```
var suggest = new string[] {"Get groceries", "Buy gas", "Post letter"};

PresentTextInputController (suggest, WatchKit.WKTextInputMode.AllowEmoji, (result) => {
    // action when the "text input" is complete
    if (result != null && result.Count > 0) {
    // this only works if result is a text response (Plain or AllowEmoji)
        enteredText = result.GetItem<NSObject>(0).ToString();
        Console.WriteLine (enteredText);
        // do something, such as myLabel.SetText(enteredText);
    }
});
```

Slika 19 - primjer koda za unos teksta sa tri predefinirane stavke, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/text-input

Kada se odabere *WKTextInputMode.Plain* korisnik može koristiti listu s predefiniranim porukama, *Scribble* ili može diktirati što želi napisati. Rezultat se uvijek vraća kao *NSObject*.

Može se i postaviti obvezno diktiranje poruke bez da se korisniku ponudi mogućnost korištenja predefinirane liste ili *Scribblea*. Za odvođenje korisnika izravno na zaslon za diktiranje, mora se proslijediti prazan *array* za popis prijedloga i postaviti *WKTextInputMode.Plain*.

```
PresentTextInputController (new string[0], WatchKit.WKTextInputMode.Plain, (result) => {
    // action when the "text input" is complete
    if (result != null && result.Count > 0) {
        dictatedText = result.GetItem<NSObject>(0).ToString();
        Console.WriteLine (dictatedText);
        // do something, such as myLabel.SetText(dictatedText);
    }
});
```

Slika 20 - primjer koda za obvezno diktiranje, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/text-input

Kada korisnik govori, zaslon sata prikazuje zaslon na kojem se nalazi razumljeni tekst (npr. „This is a test“) (Ortinau i ostali, 2021).

Slika 21 - zaslon sata pri diktiranju, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/text-input-images/dictation.png

3.2.6. Veličina zaslona

Apple Watch ima dvije veličine zaslona: 38mm i 42mm, obje omjerom zaslona 5:4 i Retina zaslonom. Njihove korisne veličine su:

38mm: 136 x 170 *logical pixels* (272 x 340 *physical pixels*)

42mm: 156 x 195 *logical pixels* (312 x 390 *physical pixels*)

Kako bi se ustanovilo na kojem zaslonu radi aplikacija može se upotrijebiti metoda WKInterfaceDevice.ScreenBounds.

Općenito, lakše je razviti dizajn teksta i izgleda za manji zaslon; od 38mm, a zatim ga povećati jer smanjenje može dovesti do ružnog preklapanja ili skraćivanja teksta (Ortinau i ostali, 2021).

3.2.7. Ograničenja watchOS-a

Postoje neka ograničenja watchOS-a kojih treba biti svjestan kada se razvijaju aplikacije za watchOS (Ortinau i ostali, 2021):

- Apple Watch ima ograničen prostor za pohranu tako da se mora pažljivo baratati dostupnim prostorom prije preuzimanja velikih datoteka (npr. audio ili filmskih datoteka).
- Mnoge watchOS kontrole imaju analoge[1] u UIKit-u, ali su različite klase (WKInterfaceButton umjesto UIButton, WKInterfaceSwitch umjesto UISwitch, itd.) i imaju ograničen skup metoda u usporedbi s njihovim UIKit ekvivalentima. Osim toga, watchOS ima neke kontrole kao što je WKInterfaceDate (za prikaz datuma i vremena) koje UIKit nema.
- Obavijesti se ne mogu usmjeravati samo na sat ili samo na iPhone već se prikazuju ili na oboje ili ni na jednom.
- API-evi koji omogućuju da Apple Watch kontrolira iTunes na povezanom iPhoneu su privatni (Ortinau i ostali, 2021).

[1] analogan. Hrvatska enciklopedija, mrežno izdanje. Leksikografski zavod Miroslav Krleža, 2021. Pristupljeno 16. 5. 2022. <http://www.enciklopedija.hr/Natuknica.aspx?ID=68023>

4. Wear OS

Wear OS (prethodno Android Wear) je operativni sustav koju je razvio Google, iako Google ne proizvodi svoje pametne satove, operativni sustav koristi nekoliko drugih proizvođača uključujući Samsung, Sony, Fossil i Skagen. (Watchranker, n.d)

Wear OS je zapravo „smanjena“ verzija Android sustava i stoga se vrlo dobro integrira s Android telefonima. (Watchranker, n.d) Omogućen je pristup svim Google-ovim uslugama kao što su *Gmail*, *Google Play Store*, *Google Assistant*, *Google Maps* itd. Također, podržava *WiFi*, *Bluetooth*, *NFC* i *LTE*, naravno pod uvjetom da je prisutan odgovarajući hardver. Također, moguće je upućivati i primati telefonske pozive ako sat ima i zvučnik i mikrofon. Ukoliko sat ima LTE vezu, moguće je obavljati pozive i bez povezanog pametnog telefona. Nadalje, Wear OS podržava izravnu sinkronizaciju *Google Fita* koji koristi senzore pametnog sata za spremanje podataka o aktivnostima u stvarnome vremenu (Kansal, 2022).

Iako neki Wear OS satovi imaju vlastita lica i prilagođene *launchere* od proizvođača, Wear OS većini ljudi nudi jedinstveno iskustvo s istim korisnički sučeljem i velikim izvorom aplikacija, što olakšava prebacivanje između različitih marki (Watchranker, n.d.).

Još neke značajke Wear OS-a su: reprodukcija glazbe i videozapisa, pokretanje aplikacija poput Instagrama, Facebooka i WhatsAppa i prepoznavanje glasa (Kansal, 2022).

Što se unosa tiče, prvenstveno se koristi *touchscreen*, ali mnogi proizvođači dodali su i krunski kotačić koji se može koristiti za listanje kroz aplikacije (Wathcranker, n.d.).

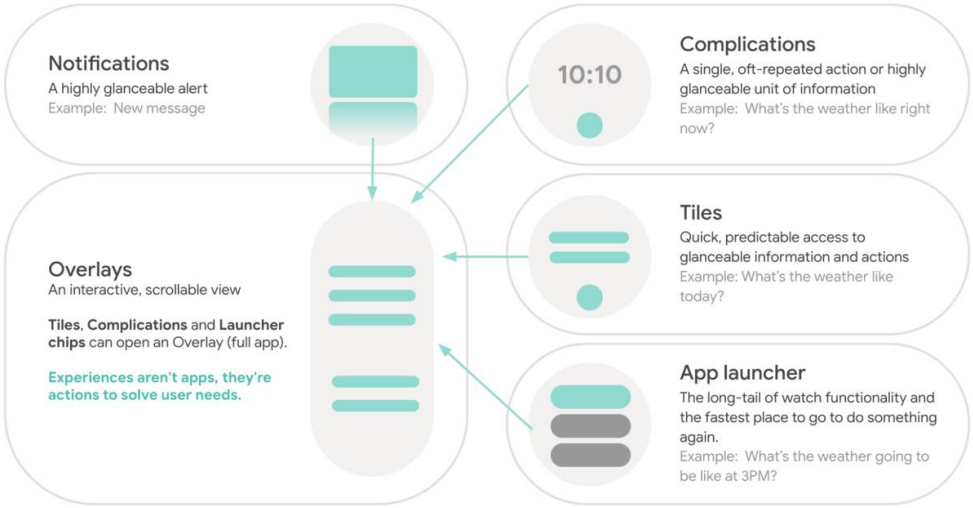
Uz *Google Assistant* može se provjeriti vremenska prognoza, pokrenuti trčanje, postaviti podsjetnik, kontrolirati kompatibilne kućne pametne uređaje i još mnogo toga koristeći glas. On također podržava takozvane *Smart Suggestions,* tako da kad mu postavite pitanje, on će automatski postaviti dodatna pitanja povezana s razgovorom. Odgovori *Google Assistanta* izlaze kroz zvučnik sata (ako ga ima) ili kroz bluetooth slušalice. Ako sat nema zvučnik i nikakve slušalice nisu povezane, odgovor se prikazuje na brojčaniku sata (O'Boyle, 2019).

4.1. Wear OS development.

Wear Os temelji se na Androidu pa se mnogi najbolji primjeri iz prakse za Android odnose i na Wear OS. Međutim, Wear OS je ipak optimiziran za zapešće tako da postoje neke razlike između njih (Android developers, 2021).

4.1.1. Korisničko sučelje Wear OS-a

Wear OS korisnicima olakšava interakciju s aplikacijama optimiziranim za sat. Površine aplikacija na Wear OS-u dizajnirane su imajući na umu „poslove“. Na primjer, ako korisnik više puta dnevno pogleda jednu jedinicu informacija, treba se razmisliti o implementaciji nekog *complicationa*. Ako je sadržaj visoke vrijednosti i vrlo kontekstualan, onda se može implementirati nekakva obavijest.



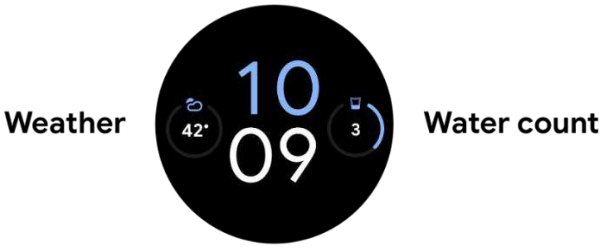
Slika 22 - pregled korisničkog sučelja Wear OS-a, preuzeto sa: https://developer.android.com/wear/images/ui_overview_7.png

Još jedan koristan način za intuitivno dizajniranje sadržaja aplikacije na Wear OS-u je razmatranje prioriteta informacija na svim površinama, podižući najvrjedniji sadržaj na površine Wear OS-a koje se mogu pregledavati. Sadržaj s najvišim prioritetom trebao bi se prikazati u C*omplicationima* i obavijestima, a zatim se može koristiti veći prosto na *Tiles*u i *Overlayu* za prikladan prikaz više sadržaja, a u nastavku bit će objašnjen svaki element prisutan na slici 23.

Slika 23 - primjer pametnog dizajna korisničkog sučelja, preuzeto sa: https://developer.android.com/wear/images/ui_overview_2.png

· *Complication*

Complication ili komplikacija je pojedinačna, često ponovljena radnja ili jedinica informacije koja se vrlo brzo može vidjeti na *watch faceu*. Kao i kod pločica (*tiles*), korisnici mogu dodirnuti komplikaciju kako bi otvorili *overlay* na satu za dobivanje više informacija. Mogu se koristiti za datum, unos popijene vode, korake i trenutno vrijeme.



Slika 24 - primjer *complicationa*, preuzeto sa: https://developer.android.com/wear/images/ui_overview_3.png

· *Notification*

Obavijest pruža pregledne, vremenski osjetljive informacije i radnje za korisnika. Obavijesti na Wear OS-u slične su obavijestima na mobilnim uređajima. Koriste se za prikazati novu poruku ili e-poštu, praćenje treninga nakon što je korisnik napustio *overlay*, i za prikazivanje informacija o trenutnoj pjesmi koja se svira.

Slika 25 - prikaz obavijesti na watch OS-u, preuzeto sa:
https://developer.android.com/wear/images/ui_overview_5.gif

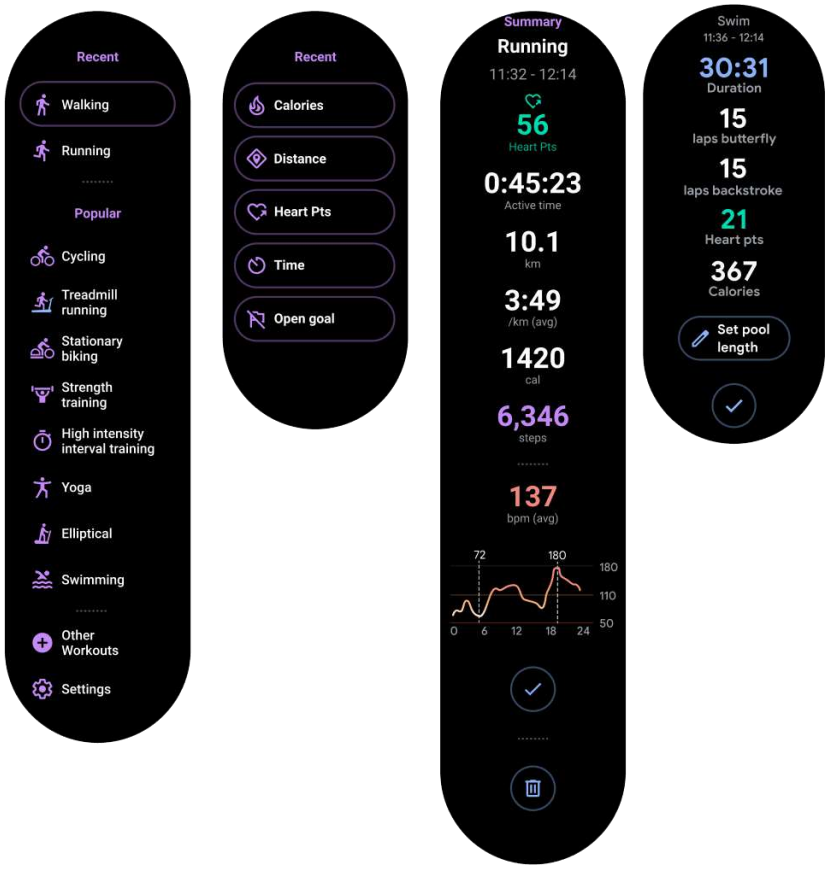
- ***Tile***

Tiles ili pločice pružaju brz i predvidljiv pristup informacijama i radnjama za rješavanje potreba korisnika. Dok *overlays* mogu biti imerzivni, pločice se brzo učitavaju i fokusiraju se na neposredne potrebe korisnika. Ako korisnici žele više informacija, mogu dodirnuti *Tiles* kako bi ih otvorili. Mogu se koristiti za praćenje dnevnih napredaka korisnika, za započeti nedavno sviranu pjesmu, poslati poruku omiljenim kontaktima ili za brzi početak vježbanja.

Slika 26 - *tiles* prikaz, preuzeto sa: https://developer.android.com/wear/images/ui_overview_6.gif

- ***Overlay***

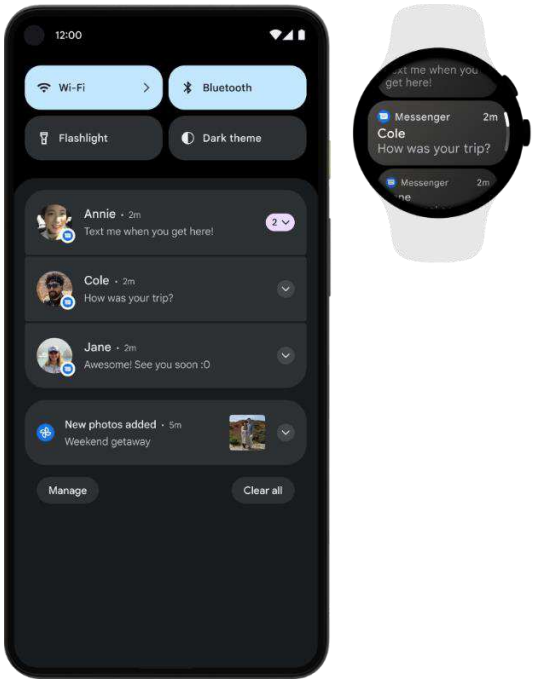
Overlay je fokusirani prikaz koji može poslužiti složenom ili manje uobičajenom zadatku ili skupu zadataka. Imerzivan je i sličan glavnom korisničkom sučelju mobilne aplikacije, iako postoje neke sitne razlike. Ostale površine (*tiles, complications i notifications)* mogu se povezati u *overlay* koji omogućuje izvršavanju složenijih zadataka. Može se koristiti za započeti vježbanje, pregledavanje popisa za reprodukciju, slanje poruka i pregledavanju podataka o vježbanju.



Slika 27 - prikaz *overlaya*, preuzeto sa: https://developer.android.com/wear/images/ui_overview_8.png

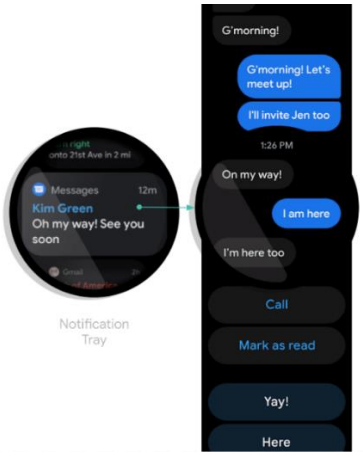
4.1.2. Obavijesti na Wear OS-u

Obavijesti na Wear OS satovima koriste iste API-jeve i imaju istu strukturu kao i obavijesti na telefonima i na satu se mogu pojaviti na dva načina. Prvi je da aplikacija na mobitelu stvara obavijest i sustav automatski premošćuje tu obavijest sa satom, a drugi način je da aplikacija na satu stvara i prikazuje obavijest.



Slika 28 – prikaz iste obavijesti na mobitelu i na Wear OS-u, preuzeto sa: https://developer.android.com/static/images/ui/notifications/wear_2x.png

Za oba scenarija programeri koriste klasu NotificationCompat.Builder za izradu obavijesti. Sažeto stanje prikazuje se u traci s obavijestima za obavijesti koje se mogu prikazuju na brzinu. Ako korisnik dotakne neku obavijest, ona se proširuje, otkrivajući dodatni sadržaj i listu s mogućim radnjama (Slika 29.). Proširiva obavijest se može izraditi na isti način kao i na mobilnom uređaju s bilo kojom od podklasa NotificationCompat.Style.



Slika 29 - primjer standardne notifikacije prikazane pomoću NotificationCompat.MessagingStyle, preuzeto sa: https://developer.android.com/static/images/ui/notifications/expandable_noti.png

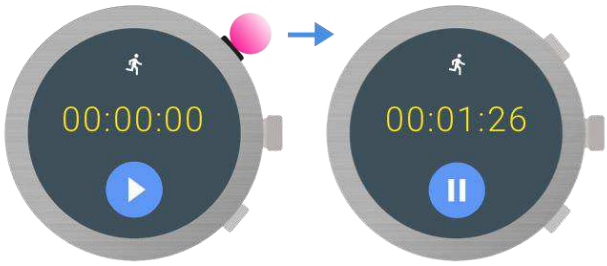
Ukoliko postoji potreba za dodavanjem značajke skrivanja ikona aplikacije iz obavijesti na satu ili dopustiti korisniku da diktira tekstualni odgovor glasovnim unosom, može se upotrijebiti klasa NotificationCompat.WearableExtender da se navedu opcije. Kako bi se koristio taj API, mora se prvo izraditi instanca WearableExtendera, postavljajući opcije za notifikacije koje se prikazuju na satu, a zatim se napravi instanca NotificationCompat.Buildera, postavljajući željena svojstva za obavijesti i pozove extend() prosljeđujući WearableExtender. Na kraju, za izradu obavijesti poziva se build() (Android developers, 2021).

4.1.3. User Input

- ***Fizički gumbi***

Pametni sat koji koristi Wear OS obično sadrži više fizičkih gumba, i uvijek imaju najmanje jedan gumb (*power button*). Osim gumba za napajanje može biti prisutan nula ili više višenamjenskih tipki.

U Svakoj aplikaciji se mogu dodijeliti gumbovi raznim radnjama, na primjer, fitness aplikacija na slici 30. može pokrenuti ili pauzirati trening pomoću gumba.



Slika 30 - primjer rada višenamijenskih gumbova, preuzeto sa: https://developer.android.com/training/wearables/user-input/physical-buttons

Kako bi se dobile dodatne informacije o gumbima na uređaju, koristi se API definiran u biblioteci Wear Input AndroidX. Kako bi se saznalo koliko je gumba dostupno na uređaju, koristi se metoda WearableButtons.getButtonCount(), ali ova metoda

uključuje i *power button* pa ako metoda vrati vrijednost veću od jedan, tada je dostupan broj višenamjenskih gumbova jednak vraćenoj vrijednosti minus 1.

```
int count = WearableButtons.getButtonCount(context);

if (count > 1) {
  // There are multifunction buttons available
}

WearableButtons.ButtonInfo buttonInfo =
  WearableButtons.getButtonInfo(activity, KeyEvent.KEYCODE_STEM_1);

if (buttonInfo == null) {
  // KEYCODE_STEM_1 is unavailable
} else {
  // KEYCODE_STEM_1 is present on the device
}
```

Slika 31 - primjer koda za spremanje gumbova u Javi, preuzeto sa: https://developer.android.com/training/wearables/user-input/physical-buttons#java

Postoji niz mogućih *„button“* kodova s kojima se aplikacija može koristiti a to su recimo: KEYCODE_STEM_1, KEYCODE_STEM_2 itd. Aplikacija može primiti te kodove i pretvoriti ih u određene radnje u aplikaciji. Kako bi prihvatili pritisak gumba kao radnju mora se implementirati metoda onKeyDown().

```
@Override
// Activity
public boolean onKeyDown(int keyCode, KeyEvent event){
  if (event.getRepeatCount() == 0) {
    if (keyCode == KeyEvent.KEYCODE_STEM_1) {
      // Do stuff
      return true;
    } else if (keyCode == KeyEvent.KEYCODE_STEM_2) {
      // Do stuff
      return true;
    } else if (keyCode == KeyEvent.KEYCODE_STEM_3) {
      // Do stuff
      return true;
    }
  }
  return super.onKeyDown(keyCode, event);
}
```

Slika 32 - primjer koda za reakciju na pritisak gumba, preuzeto sa: https://developer.android.com/training/wearables/user-input/physical-buttons#java

- ***Digitalna kruna***

Neki Wear OS uređaji sadrže fizički rotirajući bočni gumb – digitalnu krunu. Kada korisnik okrene krunu, prikaz aplikacije trebao bi se pomicati gore ili dolje.

Mnogi *containeri* koji se mogu pomicati, kao što su ScrollView, ListView, HorizontalScrollView i WearableRecyclerView, podržavaju rotiranje krune kao unos bez potrebe za bilo kakvim specifičnim kodom. Da bi mogao odgovoriti na rotaciju krune, container koji se može pomicati mora imati fokus, sljedeće su najbolje prakse u vezi odgovaranja na rotaciju krune (Android developers, 2021):

- Po *defaultu*, da bi se nekom *view* dao fokus, on mora koristiti oznaku <requestFocus /> ili ručno pozvati View.requestFocus().
- *Views* koji se mogu *scrollati* bi trebali biti označeni sa android:focusable="true" i android:focusableInTouchMode="true".
- *Rotary input* događaji šalju se samo u fokusirani *view*. Ovi događaji ne povećavaju hijerarhiju *viewa*. Ako ne postoji fokusirani *view* ili ako fokusirani *view* vrati false iz View.onGenericMotionEvent(), tada se taj događaj šalje u Activity.onGenericMotionEvent().
- Ako je *view* koji se pomiče priložen nakon Activity.onCreate() (na primjer, čekanje da se mrežni zahtjev završi prije izgradnje korisničkog sučelja), pozove se requestFocus() nakon što ga se priloži.
- Ako je *view* koji se može pomicati u početku *INVISIBLE* ili *GONE*, mora se pozvat requestFocus() kada ga se postavi na *VISIBLE*.
- Ako aktivnost sadrži više *viewova* koji se mogu pomicati, mora se odabrati jedan za fokusiranje pomoću oznake <requestFocus />.
- Ako UI sadrži neki drugi *view* koji zauzima fokus kada korisnik stupi u interakciju s njim, npr. ulazni tekst, mora se korisniku pružiti način da vrati fokus na prikaz koji se može okretati. To se postiže praćenjem dodira na *viewu* koji se može okretati i pozivom requestFocus() kao *response* (Android developers, 2021).
-

```
myView.setOnGenericMotionListener(new View.OnGenericMotionListener() {
  @Override
  public boolean onGenericMotion(View v, MotionEvent ev) {
    if (ev.getAction() == MotionEvent.ACTION_SCROLL &&
        ev.isFromSource(InputDeviceCompat.SOURCE_ROTARY_ENCODER)
    ) {
      // Don't forget the negation here
      float delta = -ev.getAxisValue(MotionEventCompat.AXIS_SCROLL) *
          ViewConfigurationCompat.getScaledVerticalScrollFactor(
              ViewConfiguration.get(context), context
          );

      // Swap these axes to scroll horizontally instead
      v.scrollBy(0, Math.round(delta));

      return true;
    }
    return false;
  }
});
```

Slika 33 - primjer korištenja MotionEvent, InputDeviceCompat i ViewConfigurationCompat, preuzeto sa: https://developer.android.com/training/wearables/user-input/rotary-input

- ***Glasovni unos***

Svaki Wear OS uređaj dolazi s mikrofonom, tako da korisnici mogu koristiti svoj glas za interakciju s uređajem. Postoje tri glavne vrste interakcije: snimanje glasa/zvuka, glasovne radnje i govorni unos.

Kako bi se dobio govorni unos od korisnika, mora se pozvati ugrađena aktivnost za prepoznavanje govora. Njime se onda mogu pisati i slati poruke ili pretraživati. To je moguće kad se u aplikaciji pozove startActivityForResult() koristeći ACTION_RECOGNIZE_SPEECH. Time započinje prepoznavanje govora a rezultat se može rukovati pomoću onActivityResult() (Android developers, 2021).

```java
private static final int SPEECH_REQUEST_CODE = 0;

// Create an intent that can start the Speech Recognizer activity
private void displaySpeechRecognizer() {
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
            RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
// This starts the activity and populates the intent with the speech text.
    startActivityForResult(intent, SPEECH_REQUEST_CODE);
}

// This callback is invoked when the Speech Recognizer returns.
// This is where you process the intent and extract the speech text from the intent.
@Override
protected void onActivityResult(int requestCode, int resultCode,
        Intent data) {
    if (requestCode == SPEECH_REQUEST_CODE && resultCode == RESULT_OK) {
        List<String> results = data.getStringArrayListExtra(
                RecognizerIntent.EXTRA_RESULTS);
        String spokenText = results.get(0);
        // Do something with spokenText.
    }
    super.onActivityResult(requestCode, resultCode, data);
}
```

Slika 34 - primjer kako pokrenuti i upravljati prepoznavanjem govora, preuzeto sa: https://developer.android.com/training/wearables/user-input/voice

5. Usporedba watchOS-a i Wear OS-a obzirom na funkcionalnosti

WatchOS i Wear OS dvije su najpopularnije platforme za pametne satove na tržištu i međusobno imaju neke sličnosti i ključne razlike koje ih čine jedinstvenima (Manyinsa, 2021).

1. Kompatibilnost

Glavna razlika između watchOS-a i Wear OS-a je kompatibilnost uređaja. Dok je watch OS kompatibilan i upariv samo s Apple mobilnim uređajima, Wear OS može se upariti i s iOS i s Android uređajem ukoliko je iOS uređaj noviji od iPhonea 5S ili ako je Android verzija 6.0 ili novija. Uz to, starije verzije watchOS-a ostaju kompatibilne sa starijim verzijama iOS-a za novije satove korisnik će morati nadograditi svoj iOS, tako da watchOS 8 zahtjeva zadnju inačicu iOS-a – iOS 15.

2. Sučelje

Što se sučelja tiče, watchOS i Wear OS su prilično slični. Oba operativna sustava koriste takozvani *card-based* dizajn što olakšava korisniku uvid u ono što se događa na prvi pogled. Također, oba operativna sustava koriste jednostavno sučelje koje se temelji na prelasku prstom koje olakšava navigaciju uređajem, međutim postoji nekoliko razlika.

Jedna od tih razlika je način na koji korisnik interagira s uređajem. Wear OS nudi kružne „pločice“ (*tiles*) aplikacija za brzi navigaciju i laki pristup najčešće korištenim aplikacijama. WatchOS s druge strane, ima takozvani *Dock* koji također omogućuje brzi pristup nedavno korištenim aplikacijama, a sve korisnikove aplikacije su prikazane sa šarenom mapom ikona.

Slika 35 - Wear OS i watchOS (s lijeva na desno), preuzeto sa: https://www.androidauthority.com/wp-content/uploads/2021/08/samsung-galaxy-watch-4-review-vs-apple-watch-series-6-all-apps-screen-1200w-675h.jpg.webp

Svaki Wear OS sat dolazi s unaprijed instaliranim skupom *watch faces* ili brojčanika koje korisnik može sam prilagođavati svojim potrebama i željama, a može i skinuti dodatne brojčanike iz Google Playa. Iako watchOS ne podržava *third-party* brojčanike, oni se mogu preuzeti i dijeliti s drugim korisnicima.

3. Aplikacije

I watchOS i Wear OS imaju vlastite ugrađene aplikacije koje se mogu koristiti u svakodnevnom životu. Na primjer, svaki Wear OS po *defaultu* ima unaprijed instaliran Google Play te korisnik onda može skidati koju god aplikaciju želi dostupnu za svoj uređaj. Za watchOS, iPhone aplikacije koje imaju dostupnu watchOS aplikaciju instaliraju se automatski i pojavljuju se na početnom zaslonu.

Također Wear OS satovi po *defaultu* imaju instalirano već pri prvom pokretanju sve dostupne aplikacije instalirane na uparenom mobilnom uređaju, uz dodatak *Google Messages, Google Fit* i *Google Fit.*

WatchOS također ima unaprijed instalirane aplikacije a neke od tih su *iMessages*, *Health*, *Timer*, aplikacije za fitness i mnoge druge.

WatchOS također sadrži aplikacije kao što su *Apple Pay* i *Wallet*, koje se mogu koristiti za sigurna i beskontaktna plaćanja. WatchOS 8 također dodaje nekoliko važnih ažuriranja za *Wallet*, upravljanje pametnim domom i *Photos* aplikaciji. *Wallet* sada može sadržavati sve, od ključa od kuće ili automobila do kreditnih i osobnih kartica.

4. Pomoćnici i glasovno upravljanje

Apple Watch dolazi s ugrađenim pomoćnikom, *Siri. Siri* omogućuje korištenje glasovnih naredbi za otvaranje aplikacija i upravljanje uređajima kao što su pametne žarulje i pametni TV sa zapešća. Za aktivaciju *Siri* dovoljno je podignuti zapešće i reći „Hey Siri!“.

Siri također omogućuje slanje poruka ili pokretanje *iMessage* chata s prijateljima, postavljanje alarma, pokretanje timera ili za provjeru otkucaja srca.

Google Assistant je odgovor Googlea na Appleov *Siri*. Na Wear OS satu dovoljno je reći „Ok Google!“ da bi se *Google Assistant* pokrenuo. Tada mu se mogu postaviti razna pitanja ili mu reći da radi različite stvari.

Trenutačno svim korisnicima Wear OS satova ponuđen pristup *Google Assistantu* osim korisnicima Samsunga, oni imaju pristup Bixbyju, glasovnoj usluzi Samsunga.

Wear OS također podržava opcije glasovnog upravljanja koje funkcioniraju kao prečaci za pametni sat. Neki od zadataka koji se mogu obaviti glasovnim narudžbama su: slanje i odgovaranje na poruke, postavljanje podsjetnika i upravljanje uređajima u pametnom domu.

5. Značajke vezane uz zdravlje

Wear OS omogućuje praćenje osnovnih informacija o zdravlju kao što su broj otkucaja srca, potrošene kalorije i broj napravljenih koraka sve zahvaljujući *Google Fitu*. *Google Fit* je *defaultna* zdravstvena aplikacija za Wear OS i unaprijed je instalirana na pametni sat. Wear OS također pruža vođene aktivnosti disanja putem *Google Fita* kako bi omogućio korisniku da se opusti i bilo koje doba dana. Uz *Google Fit* može se također povezati i bilo koja druga fitness aplikacija kao što su npr. MyFitnessPal, Strava, itd.

Apple Watch ima mnoge zdravstvene funkcije kao Wear OS. Na primjer, i watchOS i Wear OS uključuju GPS senzore i senzore za praćenje otkucaja srca i oba mogu pratiti razinu kisika u krvi putem SpO2. Aplikacija *Workout* Apple Watcha omogućuje korisniku praćenje i bilježenje vježbi poput trčanja, hodanja i vožnje bicikle, ako i mjerenje podataka u stvarnom vremenu tijekom vježbanja. Sat također podržava

programe poput platforme za vježbanje koja se temelji na pretplati Apple Fitness+ i značajke za detekciju pada za starije osobe (Manyinsa, 2021).

6. Budući trendovi pametnih satova

Nosiva tehnologija pametnih satova utječe na razvoj mobilnih aplikacija, prolazeći kroz niz industrija sa svojom optimalnom i fleksibilnom podrškom. Suvremeni tehnološki svijet imao je ključnu ulogu u mobiliziranju tehnologija današnjice, nadmašujući očekivanja svijeta u pogledu onoga što je izvedivo.

Buduća tehnologija u prostoru pametnih satova imat će značajan zamah u smislu inženjerske evolucije s beskrajnim mogućnostima. Prema istraživanjima, predviđa se da će 25% odrasle populacije SAD-a kupiti nosivi uređaj do kraja 2022. godine. Ti uređaji nude podršku za više platformi zajedno s brojnim korisničkim funkcijama. Neki od trendova koji će nadopuniti ovaj zamah su (Singh, 2022):

- *IoT (Internet of Things)*

IoT je jezgra ovih međusobno povezanih uređaja, povezuje uređaje i olakšava razmjenu podataka za izračunavanje logičnih ishoda za pametan život. Povezivost nije samo još jedna grana IoT-a, ona je glavna značajka pametnih satova. Budućnost IoT-a može se vidjeti u popisu slučajeva njegove upotrebe u zdravstvu kao što su: uređaji za praćenje razine glukoze, sportski pomoćni uređaji za izbjegavanje nezgoda i ozljeda, nosivi defibrilatori za srčane bolesni, daljinsko praćenje astme i bluetooth omogućene aplikacije za praćenje kritičnih pacijenata.

Apple kao i Google imaju maloprodajne kupce kao ciljanu publiku za svoje proizvode kao što su Apple Watch, pametne leće itd. Samsung, osim pametnog sata, također ima IoT rješenja u nosivim uređajima. Stoga, budućnost IoT-a snažno je povezana s pametnim satovima i nastavit će utjecati na razvoj aplikacija i operativnih sistema za krajnje korisnike.

- e-plaćanja putem pametnih satova

NFC ili *Near Field Communication Technology* povezuje uređaj sa sredstvima kao što su kreditne i debitne kartice itd. Korisnici mogu upaliti svoju aplikaciju na pametnom satu uz terminal u trgovini i odmah platiti.

- Razvoj pametnih satova

Potrošačka psihologija sklona je brzini. Pametni satovi pružaju iscrpno iskustvo za pametne telefone po daleko nižoj cijeni. Pametni satovi mogu se koristiti kao fitness

trackeri ili kao alternativa pametnom telefonu. U doba vođenim brzim interakcijama, oni mogu odmah primati obavijesti s društvenih mreža, obavijesti vezane uz e-poštu i sl., a neki pametni satovi također dolaze s ugrađenom podrškom za YouTube. Ljudi koji su svjesni zdravlja mogu pratiti svoj san, razinu glukoze, sagorjele kalorije, razinu stresa i još mnogo toga pomoću sata. Tako da bi olakšavanje interakcije sučelja pametnog sata s kućnom elektronikom i *gadgetima* trebala biti sljedeća velika stvar (Singh, 2022).

7. Rezultati provedenog kratkog istraživanja mišjenja građana o operativnim sustavima pametnih satova

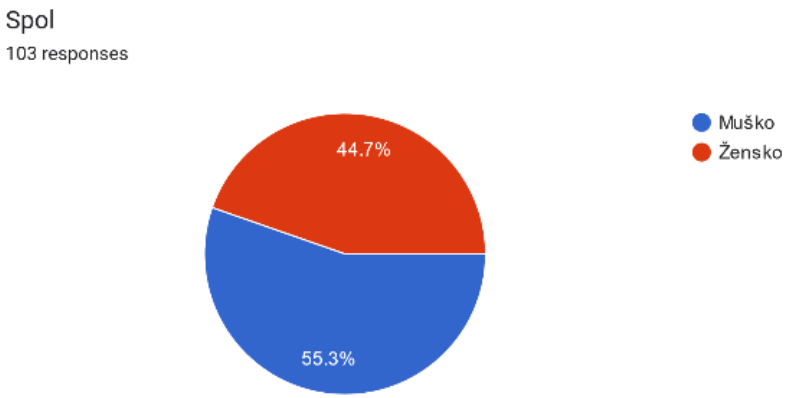
7.1. Metodologija istraživanja

U ovome poglavlju provedena je analiza kratke ankete koja je izrađena za potrebe ovog završnog rada. Cilj ankete bio je dobiti uvid u učestalost posjedovanja pametnih satova te zadovoljstvo korisnika pametnih satova njihovim značajkama.

Odgovori potrebni za ovo istraživanje primali su se od kraja srpnja do kraja kolovoza 2022. godine putem online ankete izrađene putem *Google Formsa*. Podaci su prikupljeni od korisnika i nekorisnika pametnih satova putem društvene mreže *Facebook* pri čemu je naznačeno da je upitnik anoniman. Upitnik sadrži ukupno 10 pitanja, među navedenim pitanjima nalazila su se 2 pitanja vezana uz opće karakteristike ispitanika – dob i spol, zatim pitanje posjedovanja pametnoga sata uz povezano pitanje „…iz kojeg razloga ne koristite pametni sat?“. Nakon toga slijedi pitanje vezano uz operativni sustav kojeg pojedini korisnik koristi. Anketa sadrži i 3 pitanja koja koriste Likertovu ljestvicu u kojima se ispituje zadovoljstvo korisnika operativnim sustavom i lakoća nadogradnje korisnikovog operativnog sustava. Za kraj dodana su 2 pitanja kojima je cilj istražiti koje su najkorištenije funkcionalnosti pametnih satova i koje funkcionalnosti bi pojedini korisnik, kada bi bio u mogućnosti, dodao u operativni sustav svog pametnog sata.

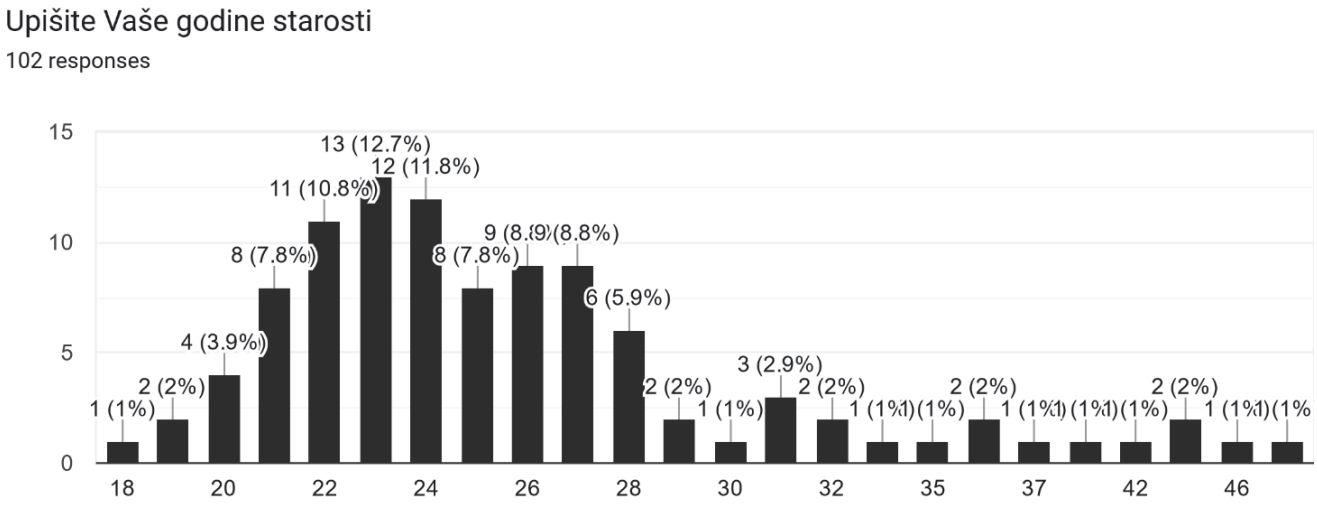
7.2. Sudionici istraživanja i njihove karakteristike

U istraživanju koje je provedeno u ovom završnom radu sudjelovalo je ukupno 102 ispitanika. Dobiveni rezultati pokazuju da je 55.9% odnosno 57 ispitanika muškog, a 44.1% ili 45 ispitanika ženskog roda (Slika 36).



Slika 36 - Postotak odgovora ispitanika o spolu (izvor: obrada autora rada)

Starosna dob ispitanika se razlikuje, a prosječna je dob 26 godina (Slika 37).

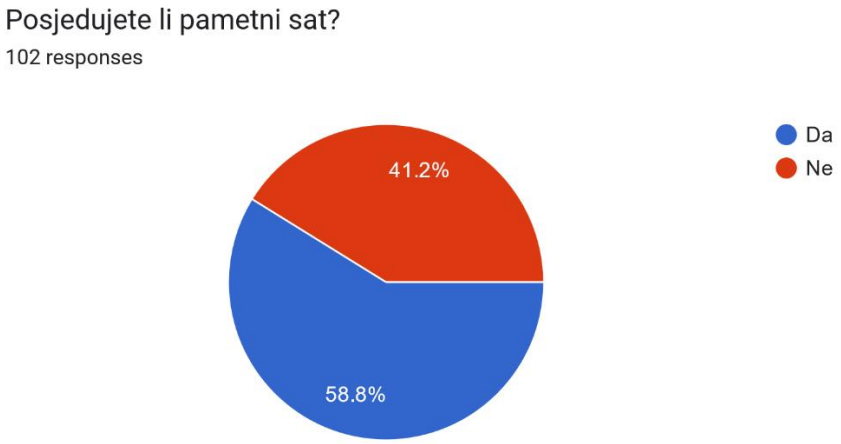


Slika 37 - Postotak odgovora ispitanika obzirom na dob (izvor: obrada autora rada)

7.3. Rezultati istraživanja

7.3.1. Rezultati ispitanika o posjedovanju pametnog sata

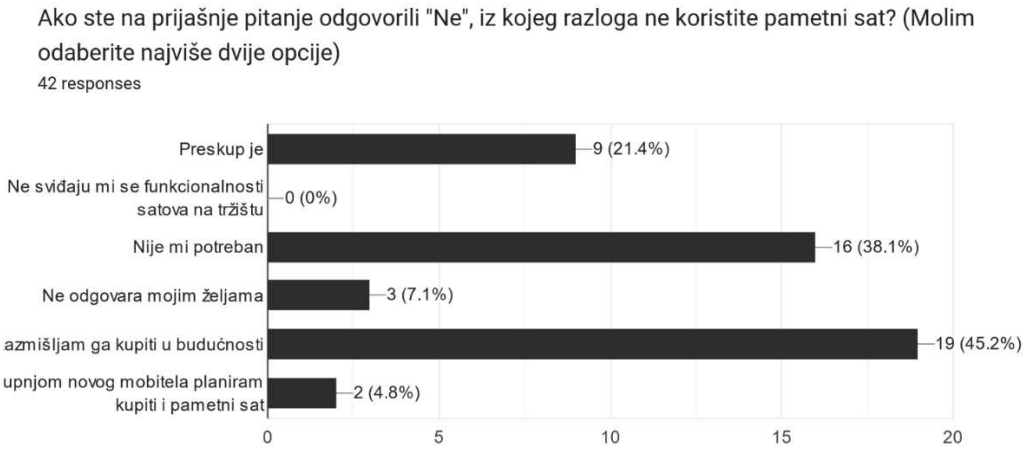
Treće postavljeno pitanje bilo je: „Posjedujete li pametni sat?“ na koje je 42 (41.2%) ispitanika odgovorilo sa „Ne“, dok je ostatak – 60 (58.8%) ispitanika odgovorilo sa „Da“ (Slika 38).



Slika 38 - Postotak odgovora ispitanika obzirom na posjedovanje pametnog sata (izvor: obrada autora rada)

7.3.2. Rezultati nekorisnika pametnih satova

Uz treće pitanje povezano je pitanje „Ako ste na prijašnje pitanje odgovorili "Ne", iz kojeg razloga ne koristite pametni sat? (Molim odaberite najviše dvije opcije)“. Na ovo pitanje odgovorili su samo ispitanici koji su na prijašnje pitanje odgovorili sa „Ne“, dakle 42 njih. Odgovori ispitanika koji ne koriste pametan sat se razlikuju, no veći dio ispitanika, 45.2% njih, odgovorilo je sa „Razmišljam ga kupiti u budućnosti“. Sljedeći najzastupljeniji odgovor jest „Nije mi potreban“ kojeg je odabralo 38.1% ispitanika, dok 21.4% ispitanika misle da je preskup, a 7.1% ispitanika odnosno su odgovorili sa „Ne odgovara mojim željama“. 4.8% ispitanika su odgovorili sa „Kupnjom novog mobitela planiram kupiti i pametni sat“, dok nijedan ispitanik nije odgovorio sa „Ne sviđaju mi se funkcionalnosti satova na tržištu“ (Slika 39.). Odgovori prikazani grafom pokazuju da ispitanici uglavnom razmišljaju o kupnji pametnog sata ili im nije potreban. Preostalim je ispitanicima izvan budžeta ili im je nepotreban.



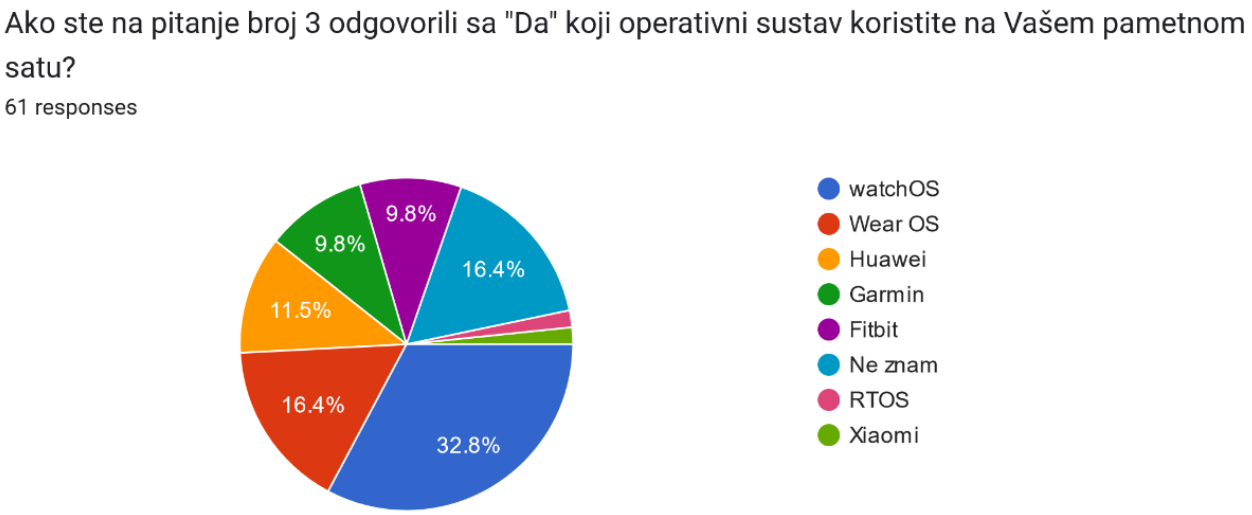
Slika 39 - Postotak odgovora ispitanika na razlog ne korištenja pametnog sata (izvor: obrada autora teksta)

7.3.3. Rezultati korisnika pametnih satova

Na sljedeća pitanja u ovome poglavlju odgovarali su samo ispitanici koji su pitanje broj tri „Posjedujete li pametni sat“ označili sa „Da“, a mišljenja nekorisnika pametnih satova nisu bila uključena, obzirom da ne mogu ponuditi tražene odgovore.

7.3.3.1. Rezultati korisnika pametnih satova o operativnom sustavu kojeg koriste

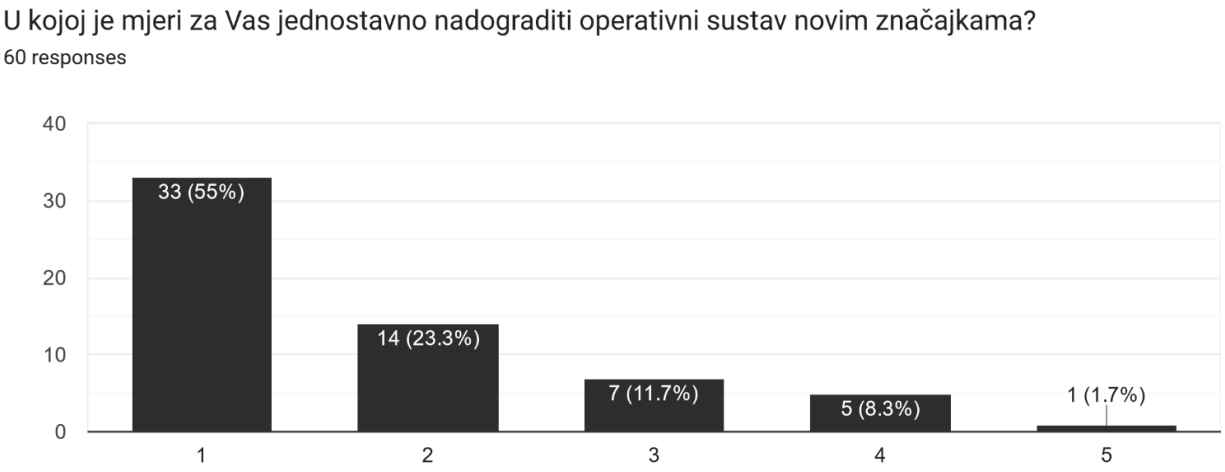
Pitanje broj pet (5) glasilo je „Ako ste na pitanje broj 3 odgovorili sa "Da" koji operativni sustav koristite na Vašem pametnom satu?“. Na slici 40 rezultati pokazuju da 33.3% ispitanika koriste WatchOS. Sljedeći najkorišteniji operativni sustav je Wear OS kojeg koristi 16.7% ispitanika, 15% ispitanika je odgovorilo da ne znaju koji operativni sustav koriste. Nakon toga redom idu: Huawei kojeg koristi 11.7% ispitanika, i Garmin i Fitbit koristi 10% ispitanika, dok i RTOS i Xiaomi koristi 1.7% ispitanika (Slika 40).



Slika 40 - Postotak odgovora ispitanika obzirom na korišteni operativni sustav (izvor: obrada autora teksta)

7.3.3.2. Rezultati korisnika pametnih satova o lakoći nadogradnje operativnog sustava novim značajkama

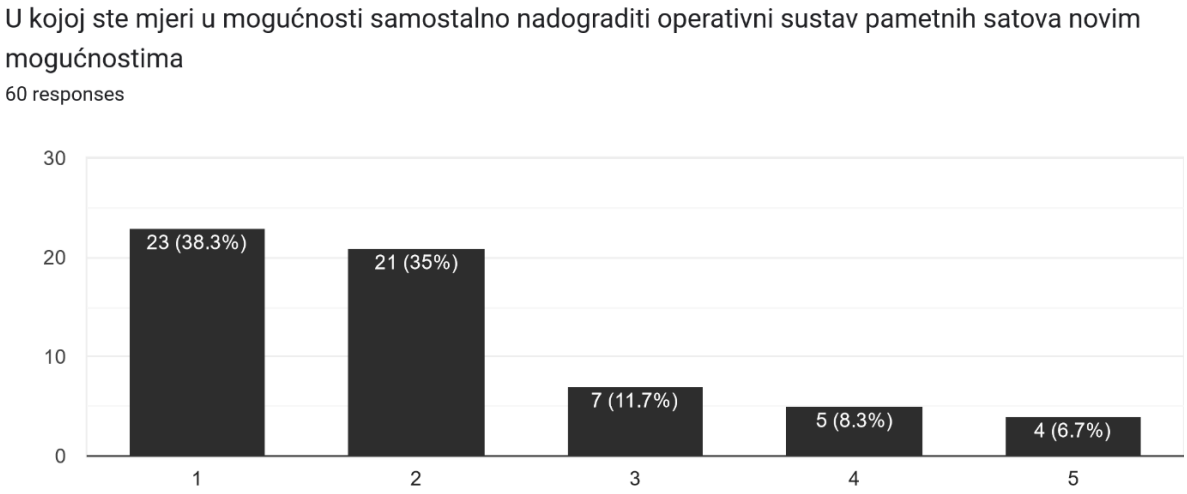
Pitanje broj šest (6) glasi „U kojoj je mjeri za Vas jednostavno nadograditi operativni sustav novim značajkama“ te na njega je odgovorilo 60 ispitanika. U ovom pitanju 1 označuje „Potpuno lako“ a 5 „Potpuno teško“. 55% ispitanika označilo je 1, 23.3% ispitanika je označilo 2, 11.7% ispitanika je odgovorilo s 3, dok je 8.3% ispitanika odgovorilo s 4, te je samo 1.7% ispitanika označilo 5 (Slika 41).



Slika 41 - Postotak odgovora ispitanika obzirom na lakoću ili težinu nadogradnje operativnog sustava novim značajkama (izvor: obrada autora teksta)

7.3.3.3. Rezultati korisnika pametnih satova o mogućnosti samostalne nadogradnje operativnog sustava novim mogućnostima

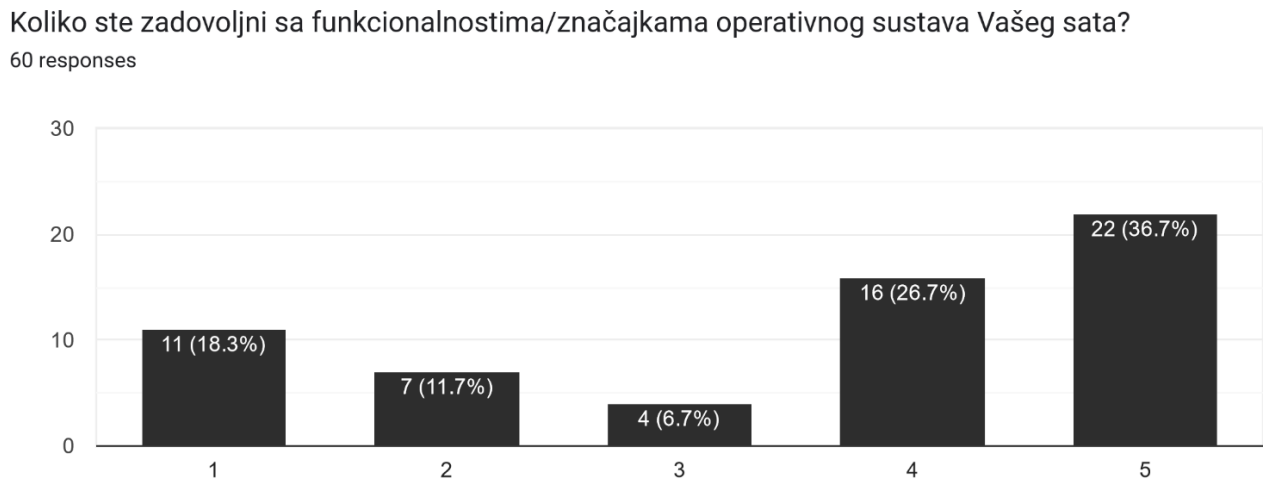
Sedmo pitanje glasi „U kojoj ste mjeri u mogućnosti samostalno nadograditi operativni sustav pametnih satova novim mogućnostima“ te također koristi Likertovu ljestvicu koja ide od 1 „Potpuno lako“ do 5 „Potpuno teško“. Rezultati pokazuju da 38.3% ispitanika smatra da je potpuno lako, 35% ispitanik su označili 2, 11.7% ispitanika je odgovorilo s 3, dok 8.3% ispitanika su označili 4, a 4 ispitanika 6.7% smatraju da je potpuno teško.



Slika 42 - Postotak odgovora ispitanika obzirom na mogućnost samostalnog nadograđivanja operativnog sustava (izvor: obrada autora teksta)

7.3.3.4. Rezultati korisnika pametnih satova o zadovoljstvu funkcionalnostima/značajkama operativnog sustava vlastitog pametnog sata

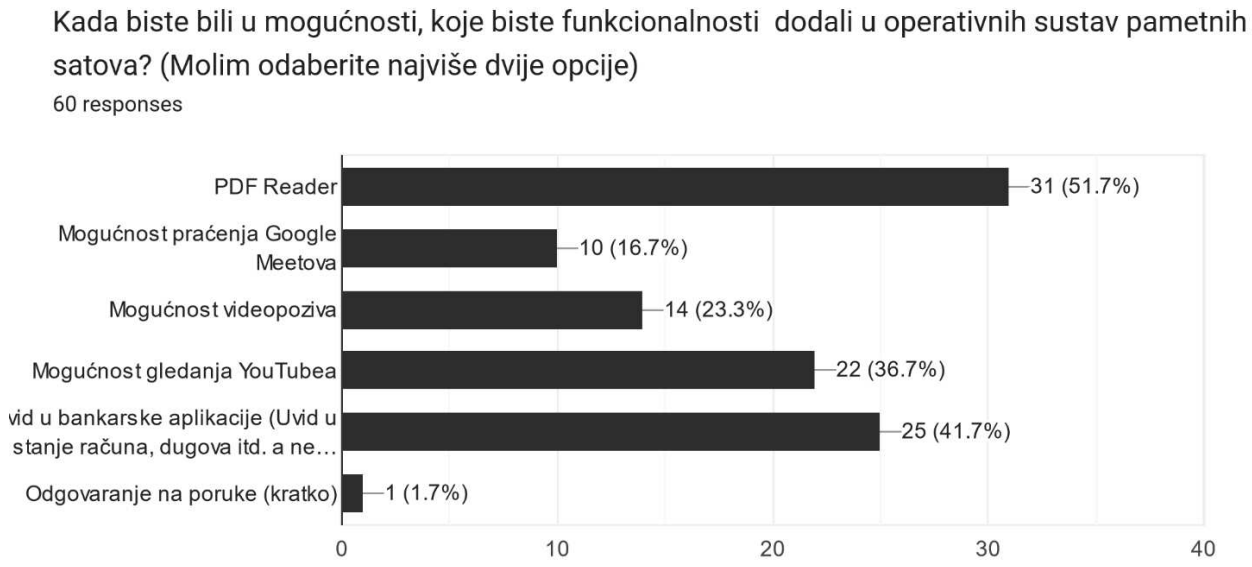
Pitanje broj osam (8), kao i prethodna dva pitanja, koristi Likertovu ljestvicu od 1 „U potpunosti nezadovoljan“ do 5 „U potpunosti zadovoljan“. Rezultati prikazani u slici 43 pokazuju da su 36.7% ispitanika u potpunosti zadovoljni značajkama operativnog sustava vlastitog pametnog sata, 26.7%ispitanika su zadovoljni, 6.7% ispitanika nisu ni zadovoljni ni nezadovoljni, 11.7% ispitanika su nezadovoljni a čak 18.3% ispitanika su u potpunosti nezadovoljni funkcionalnostima operativnog sustava vlastitog pametnog sata.



Slika 43 - Postotak odgovora ispitanika u vezi zadovoljstva funkcionalnostima operativnog sustava vlastitog pametnog sata (izvor: obrada autora teksta)

7.3.3.5. Rezultati korisnika pametnih satova u vezi dodavanja funkcionalnosti u operativni sustav pametnih satova

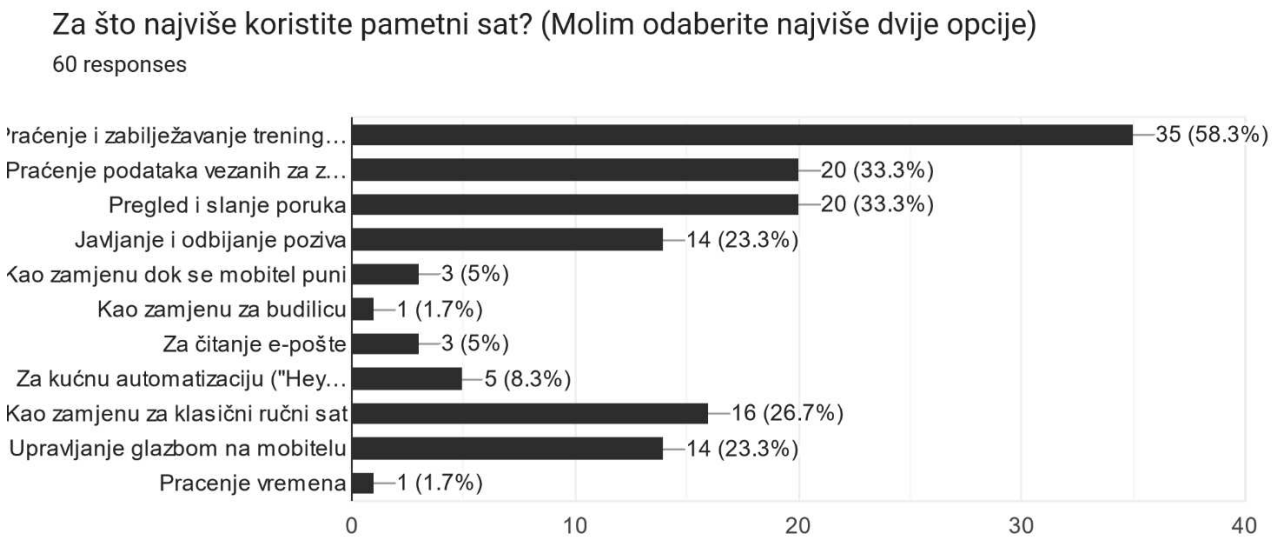
Deveto pitanje glasi „Kada biste bili u mogućnosti, koje biste funkcionalnosti dodali u operativnih sustav pametnih satova? (Molim odaberite najviše dvije opcije)“ i ispitanici su imali mogućnost odabrati ponuđene funkcionalnosti ili su mogli napisati vlastitu. Rezultati prikazani na slici 44 prikazuju da bi većina ispitanika, odnosno 51.7% bi dodalo „PDF Reader“, 41.7% ispitanika bi dodalo mogućnost uvida u bankarske aplikacije, 36.7% ispitanika je označilo da bi dodali mogućnost gledanja YouTubea, 23.3% ispitanika bi dodali mogućnosti videopoziva, 16.7% bi dodalo mogućnost praćenja Google Meetova, a 1.7% bi dodalo mogućnost odgovaranja na poruke.



Slika 44 - Postotak odgovora ispitanika u vezi dodavanja funkcionalnosti u operativne sustave pametnih satova (izvor: obrada autora teksta)

7.3.3.6. Rezultati korisnika pametnih satova u vezi korištenja pametnog sata

Posljednje pitanje ispituje za što ispitanici najviše koriste vlastiti pametni sat. I u ovom pitanju kao i u prethodnom, ispitanici su bili u mogućnosti odabrati ponuđene tvrdnje ili dodati vlastitu tvrdnju. Rezultati prikazani na slici 45 pokazuju da većina ispitanika, 58.3% koristi svoj pametni sat za praćenje i zabilježavanje treninga i drugih aktivnosti, nakon toga podjednak broj ispitanika, 33.3%, koristi sat za praćenje podataka vezanih za zdravlje i za pregled i slanje poruka. Nakon toga, 26.7% ispitanika koristi vlastiti pametni sat kao zamjenu za klasični ručni sat, 23.3% ispitanika koristi sat za javljanje i odbijanje poziva, još 23.3% ispitanika koristi sat za upravljanje glazbom na mobitelu. Manji broj ispitanika, točnije 8.3% koristi sat za kućnu automatizaciju, 5% koristi kao zamjenu dok se mobitel puni, i 5% koriste sat za čitanje e-pošte, a samo 1.7% koriste sat kao zamjenu za budilicu ili za praćenje vremena.



Slika 45 - Postotak odgovora ispitanika u vezi korištenja pametnog sata (izvor: obrada autora teksta)

8. Zaključak

Pametni satovi postaju, iz dana u dan, sve popularniji zbog lakoće korištenja i zbog sve većih mogućnosti samih satova. Korisnik sa svojim satom može uputiti i primati pozive, primati, čitati i slati poruke, upravljati aplikacijama za glazbu, zabilježavati treninge i pratiti podatke vezane za zdravlje kao npr. sagorjele kalorije tijekom treninga, pregledati koliko mu traje trening, pratiti odrađene serije i setove treninga itd. Korisnik uz sve to može i obavljati plaćanja na POS uređajima putem NFC tehnologije koje svaki moderni sat podržava. Također, satovi danas imaju mogućnost upravljanja pametnom kućom, tako da korisnik može paliti i gasiti svjetla u kući, paliti razne uređaje za grijanje ili hlađenje kao što je npr. klima i koristiti sve ostale mogućnosti pametne kuće.

Ispitanici u provedenoj anketi koji još ne posjeduju pametni sat su većinom potvrdili da ga razmišljaju kupiti u budućnosti, dok ostatak smatra da im je pametan sat ili nepotreban ili preskup. U rezultatima ankete vidljivo je i da ispitanici koriste razne satove s raznim operativnim sustavima kao što su watchOS, wear OS, Huawei, Garmin, Fitbit, RTOS i Xiaomi. Većini je ispitanika lako nadograditi sustav novim značajkama i u mogućnosti su samostalno nadograditi operativni sustav novim mogućnostima. Uz to, zadovoljni su značajkama koje im njihovi pametni satovi nude. Kad bi mogli dodati neke značajke, u najvećem postotku ispitanici bi dodali rješenje za čitanje PDF dokumenata, dok su ostali najzastupljeniji odgovori: uvid u bankarske aplikacije (uvid u stanje računa, dugova itd.) i mogućnost gledanja YouTubea. Što se tiče najkorištenijih značajki, rezultati pokazuju da korisnici najviše koriste pametne satove za praćenje i zabilježavanje treninga i ostalih aktivnosti te za praćenje podataka vezanih za zdravlje. Ograničenje ankete jest mali broj sudionika, obzirom da su na anketu odgovorila samo 102 ispitanika, od kojih 42 (41.2%) ne posjeduje pametni sat. Kako bi rezultati bili precizniji, potreban je veći broj ispitanika.

Literatura

1. CoverMagazin (2021). URL: https://www.prisjetimo.covermagazin.com/?p=1288 [Pristup: 23.4.2022.]
2. TechTarget (2015). URL: https://www.techtarget.com/whatis/definition/wearables-OS-wearables-operating-system [Pristup: 25.4.2022.]
3. Yamada, K. (2014.). *What Operating Systems do Wearable Devices Run On?.* URL: https://www.makeuseof.com/tag/what-operating-systems-do-wearable-devices-run-on/ [Pristup: 26.4.2022.]
4. Hrvatska enciklopedija, mrežno izdanje. *Leksikografski zavod Miroslav Krleža* (2021). URL: http://www.enciklopedija.hr/Natuknica.aspx?ID=45234 [Pristup: 26.4.2022]
5. Watchranker (n.d). *Smartwatch Operating Systems (Complete List + Pros & Cons)*. URL: https://watchranker.com/smartwatch-operating-systems/ [Pristup 26.4.2022.]
6. Kansal, H. (2022). *7 Different Smartwatch Operating Systems Explained- Pros & Cons*. URL: https://wearablestouse.com/blog/2022/04/05/smartwatch-os-pros-cons/ [Pristup: 27.4.2022.]
7. LeFebvre, R. (2021). *What Is watchOS*. URL: https://www.lifewire.com/what-is-watchos-4690550 [Pristup: 27.4.2022.]
8. Hattersley, L. (2016). *What is Breathe for Apple Watch*. URL: https://www.macworld.com/article/672199/what-is-breathe-for-apple-watch.html [Pristup: 2.5.2022.]
9. Ortinau D., Gates, W., Britch. D, Dunn, C. (2021.). *Get Started with Xamarin.iOS*. URL: https://learn.microsoft.com/en-us/xamarin/ios/watchos/ [Pristup: 16.5.2022.]
10. Lee, G. T., Hogenson, G., Anand, M., Jones, M., Toliver, K., Martens, J., Parente, J., Jacobs, M., Warren, G., Petersen, T., Quinones, J., Brown, K., Cai, S. (2022.). *What are solutions and projects in Visual Studio?*. URL: https://docs.microsoft.com/en-us/visualstudio/ide/solutions-and-projects-in-visual-studio?view=vs-2022 [Pristup 16.5.2022.]
11. Montemagno J., Buck, A., Hemant, A., Coulter, D., Britch, D., Dunn, C., Petzold, C. (2021.). *Xamarin Essentials: MainThread*. URL:

https://docs.microsoft.com/en-us/xamarin/essentials/main-thread?context=xamarin%2Fwatchos [Pristup: 6.6.2022.]

12. O'Boyle, B. (2019.). *Wear OS: Your complete guide to Google's smartwatch OS.* URL: https://www.pocket-lint.com/smartwatches/news/google/127960-wear-os-your-complete-guide-to-googles-smartwatch-os [Pristup: 7.6.2022.]
13. Android developers (2021.). *Get started with Wear OS.* URL: https://developer.android.com/training/wearables [Pristup: 10.6.2022.]
14. Manyinsa, D. (2021.). *WatchOS vs. Wear OS: Which Is Best?.* URL: https://www.makeuseof.com/watchos-vs-wear-os-which-is-best/ [Pristup: 23.6.2022.]
15. Singh, S. (2022.). *6 Trends Shaping Up the Future for Wearable App Development in the Year 2022.* URL: https://blog.datamatics.com/7-dominant-trends-underpinning-wearable-app-development-in-2021 [Pristup: 24.6.2022.]

Popis slika

Slika 1 – početni zaslon Apple Watcha sa okruglim ikonama, preuzeto sa: https://www.mactech.com/2015/05/19/watch-os-1-0-1-released-for-the-apple-watch/5
Slika 2 - nightstand mode, preuzeto sa: https://9to5mac.com/2018/05/23/apple-watch-how-to-disable-night-stand-mode/ 5
Slika 3 - Dock, preuzeto sa: https://support.apple.com/en-us/HT206992 6
Slika 4 - Walkie-Talkie aplikacija za Apple Watch, preuzeto sa: https://support.apple.com/en-us/HT208917 7
Slika 5 - razlika u radu aplikacija u watchOS 1 i 2, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/get-started/intro-to-watchos-images/arch.png#lightbox 10
Slika 6 - struktura rješenja za watchOS aplikaciju, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/get-started/intro-to-watchos-images/projectstructure.png#lightbox 10
Slika 7 - životni ciklus WKInterfaceController klase, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/get-started/intro-to-watchos-images/wkinterfacecontrollerlifecycle.png 12
Slika 8 - referenca na Xamarin.Essentials, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/essentials/main-thread?context=xamarin%2Fwatchos 13
Slika 9 - pozivanje metode MainThread.BeginInvokeOnMainThread, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/essentials/main-thread?context=xamarin%2Fwatchos 13
Slika 10 - zasebna metoda za izvođenje koda na main threadu, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/essentials/main-thread?context=xamarin%2Fwatchos 13
Slika 11 - primjer korištenja IsMainThread, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/essentials/main-thread?context=xamarin%2Fwatchos 14
Slika 12 - Watch Kit Catalog Storyboard, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/get-started/intro-to-watchos-images/scenes.png..... 14
Slika 13 - primjer prikaza slike "Whale" iz Watch App projekta, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/image..... 15
Slika 14 – primjer prikaza slike "Bumblebee" iz WatchExtensiona, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/image..... 16
Slika 15- primjer animiranja skupa slika "Bus", preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/image..... 16
Slika 16 - dodavanje izbornika u Document Outline padu, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/menu-images/menu-action.png 17
Slika 17 - primjer korištenja metode AddMenuItem(), preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/menu..... 17
Slika 18 - primjer unosa teksta, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/text-input-images/textinput-sml.png..... 18
Slika 19 - primjer koda za unos teksta sa tri predefinirane stavke, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/text-input..... 18

Slika 20 - primjer koda za obvezno diktiranje, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/text-input 19
Slika 21 - zaslon sata pri diktiranju, preuzeto sa: https://docs.microsoft.com/en-us/xamarin/ios/watchos/user-interface/text-input-images/dictation.png 19
Slika 22 - pregled korisničkog sučelja Wear OS-a, preuzeto sa: https://developer.android.com/wear/images/ui_overview_7.png 22
Slika 23 - primjer pametnog dizajna korisničkog sučelja, preuzeto sa: https://developer.android.com/wear/images/ui_overview_2.png 23
Slika 24 - primjer complicationa, preuzeto sa: https://developer.android.com/wear/images/ui_overview_3.png 23
Slika 25 - prikaz obavijesti na watch OS-u, preuzeto sa: https://developer.android.com/wear/images/ui_overview_5.gif 24
Slika 26 - tiles prikaz, preuzeto sa: https://developer.android.com/wear/images/ui_overview_6.gif 24
Slika 27 - prikaz overlaya, preuzeto sa: https://developer.android.com/wear/images/ui_overview_8.png 25
Slika 28 – prikaz iste obavijesti na mobitelu i na Wear OS-u, preuzeto sa: https://developer.android.com/static/images/ui/notifications/wear_2x.png 26
Slika 29 - primjer standardne notifikacije prikazane pomoću NotificationCompat.MessagingStyle, preuzeto sa: https://developer.android.com/static/images/ui/notifications/expandable_noti.png .. 26
Slika 30 - primjer rada višenamijenskih gumbova, preuzeto sa: https://developer.android.com/training/wearables/user-input/physical-buttons 27
Slika 31 - primjer koda za spremanje gumbova u Javi, preuzeto sa: https://developer.android.com/training/wearables/user-input/physical-buttons#java 28
Slika 32 - primjer koda za reakciju na pritisak gumba, preuzeto sa: https://developer.android.com/training/wearables/user-input/physical-buttons#java 28
Slika 33 - primjer korištenja MotionEvent, InputDeviceCompat i ViewConfigurationCompat, preuzeto sa: https://developer.android.com/training/wearables/user-input/rotary-input 30
Slika 34 - primjer kako pokrenuti i upravljati prepoznavanjem govora, preuzeto sa: https://developer.android.com/training/wearables/user-input/voice 31
Slika 35 - Wear OS i watchOS (s lijeva na desno), preuzeto sa: https://www.androidauthority.com/wp-content/uploads/2021/08/samsung-galaxy-watch-4-review-vs-apple-watch-series-6-all-apps-screen-1200w-675h.jpg.webp 33
Slika 36 - Postotak odgovora ispitanika o spolu (izvor: obrada autora rada) 39
Slika 37 - Postotak odgovora ispitanika obzirom na dob (izvor: obrada autora rada) 39
Slika 38 - Postotak odgovora ispitanika obzirom na posjedovanje pametnog sata (izvor: obrada autora rada) 40
Slika 39 - Postotak odgovora ispitanika na razlog ne korištenja pametnog sata (izvor: obrada autora teksta) 41
Slika 40 - Postotak odgovora ispitanika obzirom na korišteni operativni sustav (izvor: obrada autora teksta) 42
Slika 41 - Postotak odgovora ispitanika obzirom na lakoću ili težinu nadogradnje operativnog sustava novim značajkama (izvor: obrada autora teksta) 42

Slika 42 - Postotak odgovora ispitanika obzirom na mogućnost samostalnog nadograđivanja operativnog sustava (izvor: obrada autora teksta) 43
Slika 43 - Postotak odgovora ispitanika u vezi zadovoljstva funkcionalnostima operativnog sustava vlastitog pametnog sata (izvor: obrada autora teksta) 44
Slika 44 - Postotak odgovora ispitanika u vezi dodavanja funkcionalnosti u operativne sustave pametnih satova (izvor: obrada autora teksta) 45
Slika 45 - Postotak odgovora ispitanika u vezi korištenja pametnog sata (izvor: obrada autora teksta) 46

Sažetak

Pametni satovi sve su moderniji i napredniji te svakom korisniku olakšavaju svakodnevni život. Svaki korisnik može pregledati i pratiti podatke vezane uz vlastito zdravlje kao što su otkucaji srca, tlak, razina kisika u krvi itd. Uz to, korisnik može pametni sat koristiti za: pregled poruka i odgovaranje istima, obavljanje poziva, upravljanje glazbom, pokretanje treninga i njegovo praćenje, beskontaktno plaćanje, kućnu automatizaciju i još mnogo toga.

Cilj ovog rada bio je upoznati se s pojmom operativnog sustava pametnih satova i usporediti dva operativna sustava: watchOS i Wear OS. U sklopu ovoga rada provedena je i kratka anketa. Podaci su prikupljeni od ukupno 102 ispitanika. Rezultati omogućuju uvid u percepciju društva u vezi operativnih sustava pametnih satova.

Ključne riječi: operativni sustav, pametni uređaji, pametni sat, IoT, watchOS, Wear OS

Abstract

Smart watches are getting more modern and advanced and they make life easier for every user. Each user can view and monitor dana related to his own health, such as hearth rate, blood pressure, blood oxygen level etc. In addition, the user can use the smart watch to: view and reply to messages, make calls, manage music, monitor training dana, contactless payment, home automation and much more.

The aim of this paper was to get acquainted with the concept of the operating system of smart watches and the comparison of two operating systems: watchOs and Wear OS. As part of this paper, a short survey was conducted. Data was collected from a total of 102 participants. The results provide an insight into society's perception regarding the operating systems of smart watches.

Key words: operating system, smart devices, smart watch, IoT, watchOS, Wear OS