

UNIVERZA V MARIBORU  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Klemen Napast

Goran Kirov

Matic Lakota

**Prometna infrastruktura**

PROJEKTNA NALOGA

Mentorji: dr. Uroš Mlakar, red. prof. dr. Aleš Holobar, mag. Sandi  
Majninger

Maribor, 2020

# Namenska programska oprema:

## **Android:**

Android OS z več kot 85-odstotnim deležem celotne industrije po svetu prevladuje v mobilni industriji. Google še naprej osvežuje svoj vodilni operacijski sistem z bolj aktualnimi izdajami, ki vsebujejo pristrčne poudarke in izboljšano varnost. Trenutno ima Google Play več kot 3,3 milijona prenesenih aplikacij in ta številka se nenehno povečuje.

Zakaj Android:

- Enostaven postopek sprejemanja aplikacije
- Neodvisnost strojne opreme
- Java in Kotlin sta programska jezika
- Učni viri

## **Android Studio:**

Android Studio je odličen IDE, ki temelji na prav tako odličnem IntelliJ IDE. Kot že ime pove, je Android Studio IDE zasnovan in razvit posebej za razvoj aplikacij za Android. To je hitro in učinkovito in lahko v nekaj sekundah nastavite nov projekt Android za različne vrste aplikacij Android.

Nekaj področij, ki namigujejo, zakaj je Android Studio boljša možnost:

### **1. Gradle Build System**

Android Studio (AS) uporablja visoko integriran sistem gradnje Gradle. To je odlično orodje, ki ponuja upravljanje odvisnosti.

Izboljša izkušnjo razvijalca, ker je bolj razširljiv.

### **2. Razpoložljivost povleci in spusti**

Android Studio ponuja grafični uporabniški vmesnik. Android Studio ponuja grafični uporabniški vmesnik. Funkcija povleci-in-spusti je zelo izvedljivo za uporabnike, ki so začetniki.

### **3. Java Code Auto-Completion**

Glede na pomembnost dokončanja kode za razvijalca je bil Android Studio zasnovan tako, da ponuja natančne rešitve.

### **4. Stabilnost sistema**

- Android Studio prihaja z manj napakami. Ponuja stabilnejše delovanje.
- Sistemska zahteva za Android Studio je nižja od ostalih IDE
- Za delovanje na zaželen način potrebuje znatno manjši RAM prostor in hitrost CPU-ja.

### **Java:**

Za razvoj mobilnih aplikacij je Java sprejet kot eden najboljših programskih jezikov.

- Objektno usmerjen - Java je objektno usmerjen programski jezik. Vsebuje vse lastnosti OOP, vključno z dedovanjem, polimorfizmom, abstrakcijo in enkapsulacijo.
- Neodvisna od platforme - Javanska koda je sestavljena v bajtno kodo, neodvisno od platforme, ki jo je mogoče izvršiti na kateri koli platformi, na kateri teče JVM.
- Robusten - Java je močan jezik, ki uporablja samodejno upravljanje pomnilnika, samodejno zbiranje smeti in ravnanje z izjemami, da odstrani kode, nagnjene k napakam.
- Visoka zmogljivost - Java omogoča visoko zmogljivost zaradi bajt-koda, ki ga zelo optimizira prevajalnik Just-In-Time (JIT).
- Varen - Java ima vgrajene varnostne funkcije, ki vključujejo statični preizkus tipa in časa izvajanja in prevajanja.
- Več niti - Java podpira več niti, ki omogočajo obdelavo več niti hkrati..

Zakaj je Java dobra možnost?

Java se je enostavno naučiti, jezik ima tekočo angleško skladnjo in manj skrivnostne znake. Java ima bogat API, razčlenitev XML, povezavo z bazo podatkov, mreženje, pripomočke in ponuja skoraj vse, kar potrebuje razvijalec. Je tudi odprtokoden jezik in na voljo brezplačno.

Prednosti Java:

1. Enostaven jezik za učenje in razumevanje
2. Prilagodljiv - zaženete ga lahko v oknu brskalnika ali navideznem stroju. To je priročno, ko ponovno uporabite kodo in posodobite programsko opremo
3. Android se zanaša na Java - Android SDK vsebuje veliko standardnih Java knjižnic
4. Pospešena montaža v Gradlu - za to smo vsi hvaležni pri sestavljanju velikih projektov
5. Aplikacije Java so bolj kompaktne in jih je lažje optimizirati, kadar so potrebne boljše zmogljivosti. Java aplikacije so v primerjavi s Kotlinom lažje (tudi v polnem analognem), ker je Java bolj pomemben jezik

**Zadolžitve članov:**

Goran je pripravil projekt, raziskal knjižnice, preučil in poskrbel za senzoriko. Implementiral je algoritem za meritve stanja cestišča glede na pospeškometer in določanja stanja ceste. Klemen in Matic sta delala na implementaciji algoritmov in pridobitvi GPS lokacije.

# Razvoj aplikacij za internet

## **Frontend:**

### **Vue.js**

Za frontend del spletne aplikacije smo se odločili za javascript ogrodje Vue. To nam je omogočilo hitro, efektivno in reaktivno izdelavo spletne aplikacije. Vue slovi po svoji kompatibilnosti in relativno preprosti uporabi. Omogoča t.i. deklarativno programiranje oz. definiranje komponent, ki se lahko potem spet znova in znova uporabljajo skozi vso spletno aplikacijo. Definiramo jim lahko lastnosti in parametre, ki so seveda lahko od uporabe do uporabe drugačni. S tem prihranimo veliko časa in ponavljajoče kode. V ozadju Vue uporablja t.i “virtual DOM (document object model)”, kar pomeni, da se stran ob spremembi ne osveži, kot pri tradicionalnih spletnih straneh, temveč se znova “renderira”. S tem dosežemo gladko uporabniško izkušnjo in tisti občutek, ki je približen uporabi dejanske mobilne aplikacije. Vue uporablja “gole” javascript objekte in vsaka sprememba (npr. klic iz API) vpliva samo na točno določeno komponento. Tako se ne osveži oz. znova renderira celotna stran, temveč samo komponenta, ki je spremembo prejela.

### **Ekosistem Vue.js**

Uradno je na voljo “Vue Devtools”, ki deluje kot vtičnik v brskalniku in nam ob kliku nanj prikaže trenutno vključene komponente in njihov “state”. Prikažemo lahko tudi t.i. “drevo komponent”.

Vue lahko vključimo kot “javascript src” in je nato naložen preko CDN (Content-Delivery-Network). To je koristno, kadar želimo ogrodje le delno vključiti v našo spletno stran. Mi smo za naš projekt uporabili vue-cli. Gre za konzolno skripto (command line interface), ki nam zgenerira okostje in pripravi vse potrebno za nadaljni razvoj naše aplikacije. Postavi se tudi strežnik, ki skrbi za “hot reload” oz. avtomatsko osveževanje brskalnika po vsaki spremembi, ki jo naredimo v kodi.

Alternative Vue so ostala ogrodja, npr. Facebookov React, Angular, Ember ali Knockout.js. V osnovi delujejo na istem principu.

## **Backend:**

Za backend smo izbrali javaskriptno ogrodje Express. Za Express smo se odločili, ker smo se člani ekipe s tem ogrodjem spoznali pri vajah in nam je bil všeč "workflow" oz. sama izdelava strežnika. Prednost te tehnologije je, da lahko le z nekaj vrsticami kode postavimo popolnoma delujoč strežnik. Nato po potrebi dodajamo še ostale module, ki jih morebiti potrebujemo za izdelavo strežnika in dodajanje funkcionalnosti. Podatkovna baza aplikacije je podprta s strani MongoDB. Za NoSQL podatkovno bazo smo se odločili, ker nam je princip izdelave takšnih "JSON dokument" baz bolj všeč in je za to nalogo tudi bolj priročen. Ker imamo dosti opravka s formatom JSON lahko pridobljene („scraped“) podatke neposredno shranimo v NoSQL bazo brez karkršnihkoli sprememb v strukturi podatkov, ali spreminjanju česa podobnega. Velik razlog za izbiro MongoDB je bil tudi da, da izjemno dobro deluje v kombinaciji z Express strežnikom. Izkoristili pa smo še eno prednost podatkovne baze Mongo. Razvijalska ekipa te podatkovne baze ponuja brezplačno storitev MongoDB Atlas. S pomočjo storitve Atlas smo lahko brezplačno gostovali našo podatkovno bazo v oblaku in prihranili čas pri vzpostavitvi baze.

Alternative ogrodju Express (v ekosistemu Javascripta) so npr. Fastify ali Koa. Gre spet za dokaj podobne stvari. Če bi se odločali v popolnoma drugi smeri, bi lahko izbrali npr. LAMP stack (Linux, Apache, MySQL, PHP) ali kakšno drugo zaledje npr. napisano v Pythonu (Django) ali kakšno Javansko.

## **Zadolžitve članov:**

Klemen je poskrbel za „scrape“ oz. pridobitev podatkov iz našega podatkovnega vira (promet.si), Matic je uredil in poskrbel za frontend ter vizualizacijo podatkov, Goran pa je poskrbel za podatkovno bazo in raziskal podatkovne vire.

## Osnove računalniškega vida

Pri tem predmetu smo kot ekipa odločili za uporabo OpenCV. Naša naloga je namreč, da iz slike/video prepoznamo prometne znake in jih identificiramo. Gre za knjižnico funkcij, ki nam olajšajo delo s slikami in videom. Omogoča hitro manipulacijo in ekstrakcijo podatkov iz slike oziroma videa, ter nato nadaljno manipulacijo s pridobljenimi podatki. Knjižnica je odprtokodna in „cross platform“, kar pomeni, da jo lahko uporabljamo na več platformah (vsaka ima seveda svoj specifičen SDK vendar v osnovi gre za isto stvar). OpenCV velja za standard med „computer vision“ knjižnicami in nima dejanskih alternativ, ki bi bile tako dobro spisane in uporabljane. Mogoče bi tukaj kot alternativo navedli dlib c++ ali morda vlfeat.

### **Zadolžitve članov:**

OpenCV spada pod androidno mobilno aplikacijo, saj smo s pomočjo kamere na mobilnem telefonu zajeli video in iz njega prepoznali znake. Vsi trije člani smo delali na algoritmičnem delu prepoznavne znakov in integraciji ogrodja v androidno aplikacijo.

## Dodatna uporabljena orodja

### **Version Control:**

Pri izdelavi projekta smo uporabljali še Github. Gre za podjetje, ki ponuja shranjevanje kode s pomočjo sistema git. Github je najbolj priljubljen vendar obstaja tudi konkurenca. Predvsem Gitlab in Bitbucket. Gitlab v osnovi ponuja SaaS storitev (software as a service), lahko pa se odločimo za self-

hosted verzijo in repozitorije gostujemo sami, neodvisno od ponudnika. To je največja prednost Gitlaba pred Githubom in Bitbucketom. V preteklosti je tudi veljalo, da smo morali za Github privaten repozitorij plačati nekaj denarja, kar je mnogo samostojnih in „hobi“ razvijalcev odvrnilo od platforme. Tukaj sta bila v prednosti Gitlab in Bitbucket. To se je nekaj časa nazaj spremenilo in tudi na Githubu lahko sedaj ustvarimo zastonjski privaten repozitorij.

### **IDE:**

Pri izdelavi aplikacije in algoritmov za njeno delovanje smo uporabljali Android Studio. Za izdelavo spletnega dela smo uporabljali tekstovni urejevalnik Visual Studio Code, saj omogoča dobro podporo različnim jezikom in ogrodjem s pomočjo vtičnikov.