

# Mobilna aplikacija za praćenje poljoprivrednih radova

---

**Kovačević, Ivana**

**Undergraduate thesis / Završni rad**

**2018**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:276622>

*Rights / Prava:* [In copyright](#) / [Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-01-27**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIK, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni preddiplomski studij računarstva**

**MOBILNA APLIKACIJA ZA PRAĆENJE  
POLJOPRIVREDNIH RADOVA**

**Završni rad**

**Ivana Kovačević**

**Osijek, 2018.**

# SADRŽAJ

<b>1. UVOD</b> .....	1
<b>1.1 Zadatak završnog rada</b> .....	1
<b>2. KORIŠTENJE INFORMACIJSKIH TEHNOLOGIJA U POLJOPRIPVREDI</b> .....	2
<b>2.1. Pregled web, mobilnih i ostalih rješenja za praćenje poljoprivrednih radova</b> .....	2
<b>2.2. Pregled poljoprivrednih radova</b> .....	3
<b>2.2.1. Kukuruz</b> .....	4
<b>2.2.2. Kupus</b> .....	6
<b>2.3. Idejno rješenje mobilne aplikacije</b> .....	8
<b>3. PROGRAMSKO RJEŠENJE</b> .....	10
<b>3.1. Korištene programske tehnologije, jezici i okoline</b> .....	10
<b>3.1.1. Android Studio</b> .....	10
<b>3.1.2. Radna okolina</b> .....	10
<b>3.1.3. Opisna datoteka (.xml)</b> .....	11
<b>3.1.4. Java datoteka (.java)</b> .....	12
<b>3.1.5. AVD</b> .....	12
<b>3.1.6. Programski jezik Java</b> .....	12
<b>3.1.7. Parse Server</b> .....	12
<b>3.2. Programsko rješenje korisničkog sučelja</b> .....	13
<b>3.3. Programsko rješenje baze podataka te analize i praćenja aktivnosti i troškova</b> .....	16
<b>4. NAČIN KORIŠTENJA, ISPITIVANJE I ANALIZA RADA APLIKACIJE</b> .....	20
<b>4.1. Način korištenja</b> .....	20
<b>4.2. Prikaz rada aplikacije na primjerima</b> .....	23
<b>4.3. Analiza rada aplikacije</b> .....	26
<b>5. ZAKLJUČAK</b> .....	27
<b>LITERATURA</b> .....	28
<b>SAŽETAK</b> .....	29

<b>ABSTRACT</b> .....	30
<b>ŽIVOTOPIS</b> .....	31
<b>PRILOZI (na CD-u)</b> .....	32

# 1. UVOD

U današnje vrijeme sve se više susrećemo sa temom korištenja novih tehnologija na području poljoprivrede i ostalih djelatnosti koje su se do danas obavljale prema ustaljenim i dobro poznatim tehnikama. No, pojava informacijskih tehnologija uvelike je olakšala dosadašnje kako administrativne, tako i one tehničke poljoprivredne aktivnosti. Informacijske i komunikacijske tehnologije svojim procesima na tržište dovode različita rješenja kojima sama poljoprivreda postaje usmjerenija i pristupačnija pojedincu. Završni rad obrađuje temu mobilne aplikacije za praćenje poljoprivrednih radova. Svaki korisnik sam upisuje svaku svoju aktivnost koju obavlja, površinu koju obrađuje te količinu goriva koja se troši prilikom obavljanja pojedine djelatnosti. Korisnik u svakom trenutku ima uvid u prošle aktivnosti.

Cilj završnog rada, odnosno aplikacije je omogućiti korisniku unos ključnih parametara pri obradi određenog poljoprivrednog zemljišta te pohranu istih na udaljenu bazu podataka, kao i naknadni prikaz svih korisnikovih aktivnosti. U svakom trenutku korisnik može vidjeti svoje prošle djelatnosti. Sama aplikacija olakšava korisniku dosadašnje praćenje svih aktivnosti te omogućuje dostupnost bitnih informacija u svakom trenutku.

U poglavlju 2 su opisana postojeća rješenja sličnih problema, kao i idejno rješenje aplikacije s opisom kultura na kojima se aplikacija može primijeniti. Poglavlje 3 opisuje programsko rješenje te tehnologije, jezik i okolinu korištene prilikom izrade aplikacije. Četvrto poglavlje obuhvaća ispitivanje i analizu rada same aplikacije i prikaz rada aplikacije na raznim primjerima.

## 1.1 Zadatak završnog rada

U radu je potrebno opisati slijed radova i razraditi model praćenje radova u poljoprivredi vezanih za određenu poljoprivrednu površinu i kulturu. Nadalje, potrebno je opisati potrebne programske okoline i tehnologije, te ostvariti korisničko sučelje i bazu podataka mobilne aplikacije koja omogućuje unos aktivnosti, korištenih alata i postupaka, a prikazuje sve dosadašnje aktivnosti

## 2. KORIŠTENJE INFORMACIJSKIH TEHNOLOGIJA U POLJOPRIVREDI

Korištenje informatičkih i komunikacijskih tehnologija u poljoprivredi podrazumijeva razvoj i primjenu inovativnih rješenja informacijskih i komunikacijskih tehnologija u ruralnim područjima s naglaskom na poljoprivredu. Razvoj različitih informacijskih i komunikacijskih tehnologija poljoprivredi omogućuje unaprjeđenje dosadašnjih poljoprivrednih i ruralnih radova. Sve većom popularizacijom informacijskih i komunikacijskih tehnologija, praćenje poljoprivrednih radova, kao i sama poljoprivreda postaje pristupačnija pojedincu jer se na tržištu pojavljuju razna web, mobilna i ostala rješenja koja olakšavaju samo praćenje i obavljanje raznih poljoprivrednih radova.

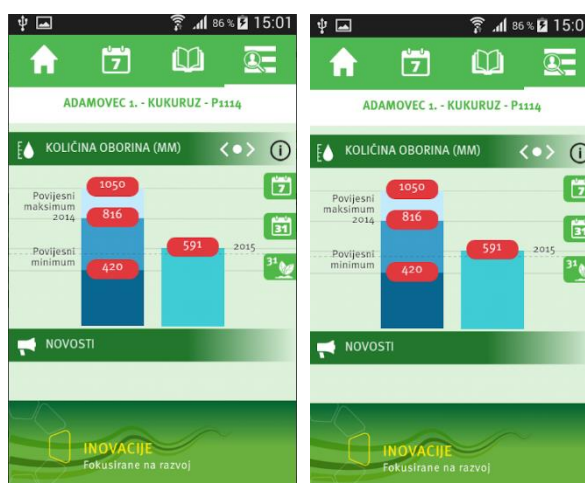
### 2.1. Pregled web, mobilnih i ostalih rješenja za praćenje poljoprivrednih radova

Danas na tržištu postoje brojna web, mobilna te ostala rješenja koja pojedincu nastoje olakšati samo bavljenje poljoprivredom. Većina tih rješenja je dostupna na engleskom jeziku kako bi bilo omogućeno korištenje što većem broju korisnika. Neka od postojećih rješenja za mobilne uređaje dostupna na *Google Play* alatu za preuzimanje aplikacija su *Agrivi*, *Agro-Assist*, *Farm Calculators*, *AgriApp*, *FarmLogs*. Od navedenih aplikacija *AgriApp* [6] aplikacija ima najveći broj preuzimanja. Aplikacija daje uvid u sve informacije o zaštiti i proizvodnji usjeva, kao i neke osnovne informacije o vezane uz traženu kulturu, što možemo vidjeti i na slici 2.1 Omogućuje poljoprivrednicima razmjenu informacija i videozapisa sa stručnjacima koji analiziraju njihove probleme te preporučuju odgovarajuća rješenja.



Slika 2.1. Zaslona aplikacije AgriApp

Aplikacija *Farm Calculators* [5] najbolje je ocijenjena od strane korisnika među navedenim aplikacijama. *Farm Calculators* pomaže korisnicima prilikom uštede troškova, izračunavanjem točne količine sredstava za prihranu, pesticida i sjemena potrebnih za održivi razvoj pojedinih usjeva. Zatim slijedi aplikacija *Agrivi* [3] koja iako ima mali broj preuzimanja omogućuje korisnicima brz uvid u sve prošle značajne aktivnosti te lak unos novih. *Agrivi* program za upravljanje pomaže korisnicima pri preuzimanju kontrole nad svojim nasadima, povećaju produktivnost te time i povećaju profitabilnost. Neke od značajnijih značajki *Agrivi* aplikacije su jednostavan i brz način planiranja i praćenja poljoprivrednih aktivnosti i korištenja raznih ulaza (goriva, gnojiva, pesticida), upravljanje zalihama pri čemu se pri niskim zalihama aktivira alarm upozorenja te spremanje svih poljoprivrednih dokumenata na oblak (eng. *cloud*). Aplikacija *FarmLogs* [7] pomaže korisnicima prilikom planiranja profitabilnog plana uzgoja, pomoć pri donošenju važnih odluka prilikom planiranja uzgoja korisniku omogućuju određene značajke same aplikacije. Neke od tih značajki su praćenje tržišta određenih kultura, odluke o prodaji ili skladištenju samih proizvoda s obzirom na kretanje cijena na tržištu, praćenje profitabilnosti i proračuna tijekom cijele sezone te mjerenje količina oborina na svakoj lokaciji. Posljednja od izdvojenih aplikacija je *Agro-Assist* [4]. Aplikacija *Agro-Assist* predstavlja digitalni dnevnik poljoprivrednih aktivnosti te sadrži opsežnu bazu podataka o proizvodima te rezultatima ispitivanja. Slika 2.2 prikazuje jednu od mogućnosti aplikacije. Navedena mogućnost korisniku u svakom trenutku daje uvid u količinu oborina i temperaturu tla za svaku lokaciju usjeva.



Slika 2.2. Zaslona aplikacije *Agro-Assist*

## 2.2. Pregled poljoprivrednih radova

Aplikacija će omogućavati korisnicima da sami upravljaju svojim gospodarstvom. Svaki korisnik sam će upisivati koju kulturu će i na koje sve načine obrađivati na određenoj lokaciji.

Osim navedenih aktivnosti korisnik upisuje i pesticide i njihovu primijenjenu koncentraciju prilikom obrade određene poljoprivredne površine. Prilikom određene aktivnosti koju korisnik obavlja, aplikacija mu nudi mogućnost upisivanja potrošnje goriva. Za primjer ćemo uzeti kukuruz kao jednu od najvažnijih kultura kontinentalne Hrvatske te kupus kao jednu od najraširenijih povrtlarskih kultura.

### **2.2.1. Kukuruz**

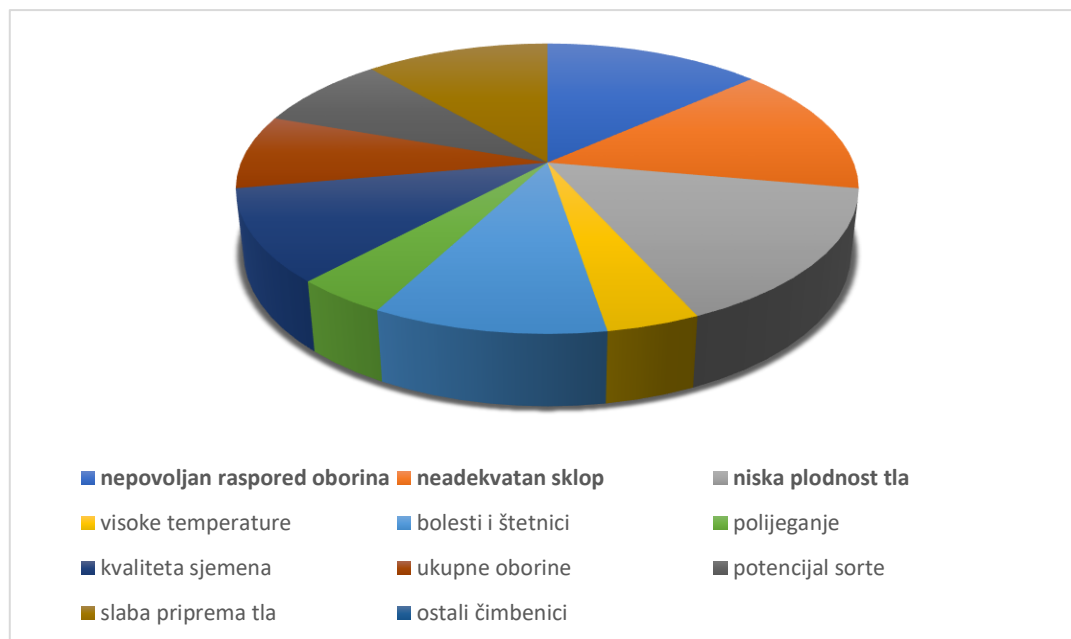
Internetska stranica Agroklub definira kukuruz kao jednogodišnju biljku jarog tipa razvoja, čija vegetacija od sjetve do pune zrelosti ovisi kako o osobinama pojedine sorte, tako i o uvjetima uzgoja. Dužina vegetacije sve hibride kukuruza razvrstava u nekoliko skupina, u one rane, srednje rane i kasne vegetacije. U rane hibride ubrajaju se oni hibridi s remenom vegetacije od 90 do 110 dana, u srednje rane ubrajamo one sa vremenom vegetacije između 120 i 135 dana, dok u kasne hibride ubrajamo one sa vegetacijom 135-145 dana [16].

Kukuruz ima veliki gospodarski značaj koji dolazi iz različitih svojstava same biljke, raznovrsnosti upotrebe te obujma proizvodnje. velika prednost kukuruza je ta što se gotovo svi dijelovi biljke mogu iskoristiti za preradu, što mu osim gospodarskog dodaje i velik ekonomski značaj. Danas se u svijetu proizvodi više od 500 različitih industrijskih prerađevina. Može se koristiti u različitim granama industrijske proizvodnje, od prehrambene i farmaceutske industrije do samih tekstilnih i kemijskih proizvoda i raznih napitaka. Osnovna sirovina za preradu kukuruza je zrno koje se najvećim dijelom koristi za pripremu koncentrirane stočne hrane jer je bogato ugljikohidratima, bjelančevinama, uljima te raznim mineralnim tvarima. Sam prinos kukuruza utječe mnogo čimbenika, a posebno klimatske prilike, plodnost i vlažnost tla te primijenjena agrotehnika. Ostale čimbenike koju utječu na prinos kukuruza prikazuje slika 2.3.

Raznim agrotehničkim mjerama može se utjecati na neke od čimbenika prikazanih na slici 2.3 te tako stvoriti više ili manje povoljne uvijete za sam rast i razvitak kukuruza. Svaki proizvođač kukuruza treba poznavati kako samu biljku kukuruza tako i pojedine faze razvoja biljke i određene aktivnosti u pojedinim fazama razvoja biljke. Što više proizvođač zna o pojedinoj biljci i odgovarajućoj fazi razvoja, to će učinkovitije znati primijeniti pojedine agrotehničke djelatnosti i mjere te samim time i povećati prinose. Uzgoj same kulture podrazumijeva primjenu raznih agrotehničkih mjera. Osnovne agrotehničke mjere koje se primjenjuju pri



uzgoju kukuruza su plodored, obrada tla, gnojidba, sjetva te njega usjeva (kultivacija, zaštita primjenom kemijskih sredstava).



Slika 2.3. Čimbenici koji utječu na uzgoj kukuruza

Kukuruz kao predkultura drugim kulturama može imati pozitivne, ali i negativne posljedice ukoliko se kasno bere jer se tada tlo teško obrađuje. Kukuruz ostavlja veliku vegetativnu masu koja uzrokuje težu obradu tla te tako i smanjuje kakvoću tla. Dobro se odnosi sa svim ostalim ratarskim kulturama te se zato s njim može izmjenjivati na istoj parceli 2 do 3 godine uzastopno, a negdje i znatno dulji niz godina. Osnovna obrada tla u treba se izvršiti u ljetno jesenskom dijelu godine. Tlo se aerira i čini toplijim obradom tla do određene dubine, rahljenjem i miješanjem, razbijanjem nepropusnog sloja. Gnojidba je vrlo važna djelatnost prilikom uzgoja svih kultura. Ako se gnojiva dodaju u osnovnoj obradi, pripremi tla za sjetvu, odmah i prihrani, biljka će u od samog početka imati na raspolaganju sve hranjive tvari potrebne za njen rast i razvoj.

U slučaju da se ranijim gnojidbama nisu unijele dovoljne količine gnojiva te ukoliko se na usjevima počnu prikazivati znakovi nedostatka hranjivih tvari vrši se prihranjivanje. Prihrana kukuruza obavlja se u ranijim fazama razvoja. Prilikom sjetve kukuruza treba odabrati provjerene sorte, odnosno sorte koje preporučuju stručne institucije za određeno područje. Kukuruz se sije sijaćicama gdje razmak između redova treba biti 70 cm. Povećanje ovog razmaka povećava konkurenciju biljaka te se biljke ne mogu ravnomjerno širiti.. Njega usjeva

podrazumijeva dvije agrotehničke djelatnosti, kultivaciju i zaštitu primjenom kemijskih sredstava. Kultivacija je važna mjera kojom se sprječava pojava pokorice, odnosno površinski sloj tla obogaćuje se kisikom, smanjuje se gubitak vode iz tla te se uništavaju korovi. Kukuruz početkom svoje vegetacije vrlo sporo raste što uzrokuje širenje i razvoj korova. Ako preventivne mjere nisu uspjele spriječiti razvoj korova, tada se moraju koristiti kemijsko suzbijanje korova pomoću različitih herbicida. Neki od najučinkovitijih herbicida prikazani su tablicom 2.1.

Tablica 2.1. Zaštita kukuruza

PREPARAT	NAMJENA	FAZA RAZVOJA	DOZA
Merlin	Jednogodišnji travni i širokolisni korovi	Prije sjetve	100-140 g/ha
Banvel 480 S	Jednogodišnji i višegodišnji širokolisni korovi	Nakon nicanja	0.5-0.7 l/ha
Cambio	Jednogodišnji i širokolisni korovi	Nakon nicanja	2.0-2.5 l/ha
Pardner	Jednogodišnji i višegodišnji širokolisni korovi	Nakon nicanja	1.0-1.5 l/ha
Motivel	Višegodišnji uskolisni i jednogodišnji korovi	Nakon nicanja	1.0-1.25 l/ha

### 2.2.2. Kupus

Prema Internetskoj stranici Agro Kluba [15], kupus je dvogodišnja zeljasta biljka koja razvija veliki pup koji nastaje savijanjem lišća. Tijekom prve godine uroda stvara vegetativne organe, a tijekom druge sjeme i cvjetno stablo. Može se upotrebljavati u svježem, konzerviranom ili kuhanom stanju naročito tijekom zime. Kupus je u Republici Hrvatskoj po površinama među prvim mjestima s oko 10 000 ha. Kupus se za jesensku berbu uzgaja u kontinentalnom dijelu, dok se za proljetnu berbu uzgaja u primorskom kraju. Sadrži velike količine ugljikohidrata, bjelančevina te vitamina C, zbog čega ima veliku važnost u prehrani tijekom zimskih mjeseci.

Kupus je kultura kojoj pogoduje vlažno i pro hladno podneblje. Svjetlost ne utječe toliko na rast biljke, ali za razvoj ploda nije pogodno zasjenjivanje. S obzirom na klimatske uvjete današnjice navodnjavanje nasada je obavezno pri čemu treba paziti na održavanje umjerene ali

dovoljne vlažnosti tla. Niža vlažnost dovodi do sporog formiranja glavica te one ostaju meke i sitne. U slučaju kada tlo presuši te se nakon toga naglo natopi, dolazi do pucanja samog ploda koji je tada lošije kvalitete te gubi svoju tržišnu vrijednost. Kupusu najbolje odgovara na duboko, plodno i strukturno tlo. Tlo koje ima veliku vlažnost nikako nije pogodno za uzgoj kupusa jer uslijed slabe prozračnosti tla dolazi do slabog rasta biljke, a samim time se glavice ne oblikuju kako bi trebale. Kao i kod kukuruza, tako i kupusa postoje različite agrotehničke mjere i djelatnosti koje osiguravaju pravila uzgoj i razvitak samog nasada. Neke od osnovnih mjera su plodored, obrada tla, gnojidba, sjetva i sadnja te njega nasada. Za uzgoj kupusa vrlo je važno kupus uzgajati prema plodoredu zbog raznih bolesti i štetnika koje ga napadaju. Kako bi se biljka zaštitila od raznih bolesti i nametnika potrebo je koristiti neka kemijska sredstva koje prikazuje tablica 2.2.

Tablica 2.2. Zaštita kupusa

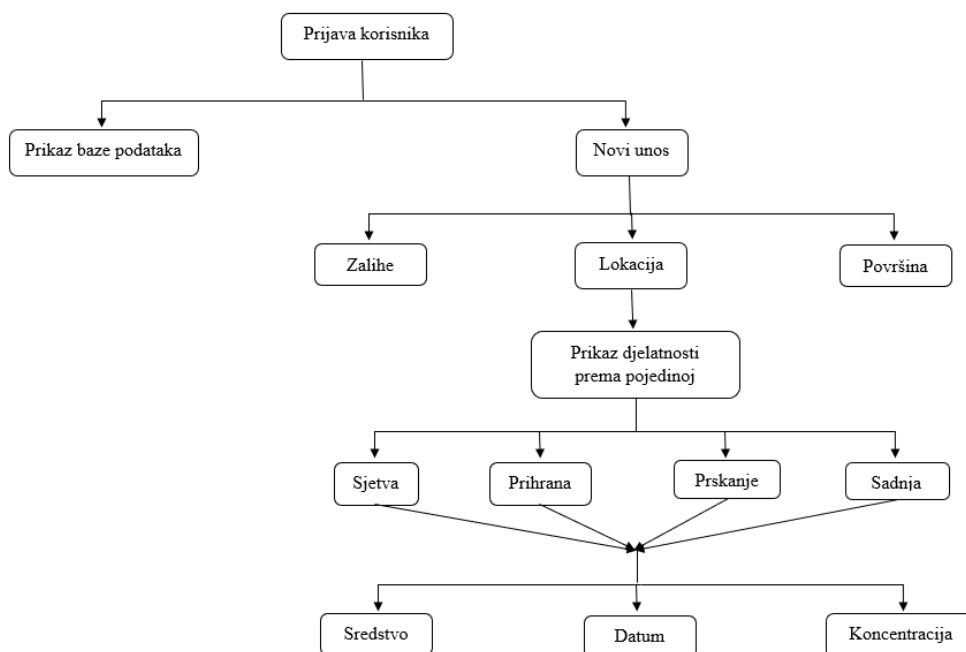
PREPARAT	NAMJENA	FAZA RAZVOJA	DOZA
Efica	Jednogodišnji travni i neki širokolisni korovi	Nakon sjetve	1.5-2.0 l/ha
Kohinor	Kupusne lisne uši	Namakanje presadnica	0.5-0.75 l/ha
Galigan	Jednogodišnji širokolisni i neki višegodišnji korovi	Prije presađivanja	1.5-2.5 l/ha
Pendigan	Jednogodišnji travni i neki višegodišnji korovi	Prije presađivanja	1.5-2.0 l/ha
Chess	Lisne uši	Pojava štetnika	0.4 kg/ha

Kao i kod kukuruza, tako i kupusa postoje različite agrotehničke mjere i djelatnosti koje osiguravaju pravila uzgoj i razvitak samog nasada. Neke od osnovnih mjera su plodored, obrada tla, gnojidba, sjetva i sadnja te njega nasada. Za uzgoj kupusa vrlo je važno kupus uzgajati prema plodoredu zbog raznih bolesti i štetnika koje ga napadaju. Kako bi se osigurao pravilan uzgoj, kupus bi na istu površinu trebao doći svake četvrte godine. Kupus je vrlo dobra predkultura za većinu povrtlarskih kultura jer iza sebe ostavlja rahlo zemljište bez korova. Obrada tla vrlo je važna mjera koja na početku uzgoja osigurava zadovoljavanje svih potrebnih preuvjeta za daljnji rast i razvoj same biljke. Priprema tla za nasad kupusa provodi se kroz tri djelatnosti, kultivacije, drljanja i valjanja. Prilikom uzgoja kupusa obavezna djelatnost je

valjanje tla jer kupus ne podnosi rastresito tlo. Prije same gnojidbe nasada potrebno je stati stanje pojedinih hranjivih tvari u tlu te samu kiselost tla. Kupus iz tla uzima velike količine hranjivih tvari, dok sama biljka zahtjeva velike količine dušika i kalija. Kako bi sam plod postigao dobru tržišnu cijenu potrebno je održati povoljan odnos dušika, fosfora i kalija. Unošenje dušika povećava prinos, no ukoliko se koristi u prevelikim količinama može doći do potpuno negativnih posljedica jer glavica ostaje meka i šuplja te gubi svoju tržišnu vrijednost. S druge strane povećanjem kalija povećava se čvrstoća samog ploda. Sama sjetva kupusa mora biti pažljivo odrađena kako bismo na karaju dobili jednoličan sklop biljaka. Na gustoću same sadnje utječe sorta kupusa te plodnost i vlažnost tla. Najbolji način kako izbjeći probleme raznih bolesti i nametnika je održavati nasad zdravim i zelenim. Glavni nametnici koji napadaju kupus su gusjenice, puževi te lisne uši koji uzrokuju bušotine na listovima.

### **2.3. Idejno rješenje mobilne aplikacije**

Sama ideja ovakve mobilne aplikacije osigurava svakom korisniku samostalno upravljanje vlastitim kulturama kao i samom potrošnjom goriva potrebnim za obradu različitih poljoprivrednih površina, a time i samim troškovima proizvodnje. Kako bismo mogli ostvariti ovakvu aplikaciju potrebno je definirati različite parametre koji omogućuju komunikaciju između korisnika i same aplikacije. Aplikacija će korisniku omogućiti unos lokacije i njene površine, odabir djelatnosti te prilikom odabira djelatnosti definiranje različitih sredstava zaštite i njihove koncentracije te datuma obavljanja djelatnosti. Također, korisnik će moći unijeti potrošnju goriva pri obradi određene djelatnosti. Jedna od glavnih funkcija ove aplikacije je da svakom pojedinom korisniku posluži kao svojevrsan dnevnik prilikom obavljanja svih svojih aktivnosti.



Slika 2.3. Dijagram tijeka korištenja aplikacije

Prilikom pokretanja same aplikacije korisniku se prikazuje početni zaslon za prijavu odnosno registraciju korisnika. Početni zaslon za prijavu sadrži polja za unos korisničkog imena i lozinke te gumb za prijavu postojećih odnosno registraciju novih korisnika. Nakon uspješne prijave ili registracije, korisniku se otvara zaslon a prikazom svih dosadašnjih aktivnosti u obliku liste. Svi podaci koji se prikazuje u listi pohranjeni u na udaljenoj bazi te ih na korisnikov zahtjev prikaza liste, aplikacija povlači iz baze. Na istom zaslonu se prikazuje i gumb za stvaranje nove aktivnosti. Zaslon nove aktivnosti sadrži polja za unos svih bitnih parametara, površine, imena parcele, datuma aktivnosti, potrošnje goriva pa i same djelatnosti.. Idejno rješenje ovakve aplikacije može se slikovito opisati dijagramom toka kao na slici 2.3.

## 3. PROGRAMSKO RJEŠENJE

U narednom poglavlju bit će opisane potrebne radne okoline i potrebna znanja za razvoj aplikacije. Za dizajn korisničkog sučelja i smještaj pojedinih elemenata korisničkog sučelja koristimo .xml datoteku. Unutar .java datoteka nalaze se sve metode potrebne za realizaciju određenih funkcionalnosti pojedinih elemenata korisničkog sučelja. Android virtualni uređaj omogućuje simulaciju aplikacije na virtualnom uređaju umjesto na stvarnom.

### 3.1. Korištene programske tehnologije, jezici i okoline

Kako bi razvili aplikaciju potrebna su određena znanja o programskom jeziku Java, razvojnom okruženju Android Studio te određena znanja iz raznih grana poljoprivrede.

#### 3.1.1. Android Studio

Android Studio je službeno integrirano razvojno okruženje (IDE) za razvoj Android aplikacija [9]. Dostupan je za Windows, macOS i Linux operacijske sustave. Prva inačica razvojnog okruženja objavljena je 2013. godine na Google I/O konferenciji, no kasnije će ova verzija biti nadograđena sve do sadašnje 3.1.3. inačice koja je objavljena u lipnju ove godine. Android Studio pruža mnoge alate za izradu aplikacija na svim Android uređajima te omogućuje kodiranje aplikacija i testiranje istih pomoću Android upravitelja virtualnih uređaja (eng. *Android Virtual Device Manager*). Nakon pojave Android Studio razvojnog okruženja, Eclipse Android Development Tools, Googleov postaje sekundarni IDE za razvoj izvornih Android aplikacija, dok Android Studio preuzima vodeću ulogu u razvoju Android aplikacija

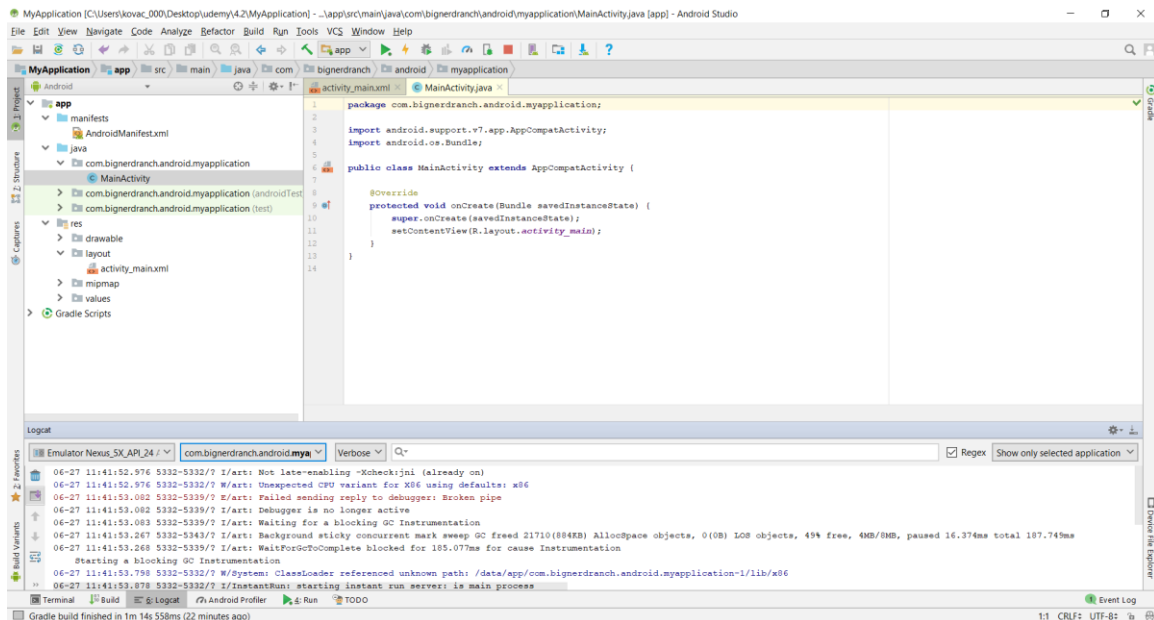
#### 3.1.2. Radna okolina

Prilikom pokretanja razvojnog okruženja Android Studio otvara nam se početni zaslon koji nam nudi brojne mogućnosti.

Prema slici 3.1 možemo vidjeti da se početni zaslon razvojnog okruženja Android Studio sastoji od nekoliko glavnih dijelova. Glavni dijelovi su:

- Alatna traka – omogućuje otvaranje novih i spremanje postojećih projekata kao i pokretanje same aplikacije i primjenu novih izmjena

- Navigacijska traka – pomaže prilikom kretanja kroz projekt, omogućuje otvaranje pojedinih datoteka za uređivanje
- Prozor *Logocat* – prikazuje poruke sustava u stvarnom vremenu te čuva povijest kako bi korisniku omogućio pretraživanje starijih poruka
- Prozor *Editor* – prostor sa stvaranje i izmjenu koda same aplikacije. Ovisno koji dio aplikacije želimo izmijeniti otvaramo onu vrstu datoteke koja nam treba, tako za izmjenu samog korisničkog sučelja aplikacije otvaramo .xml datoteku.



Slika 3.1. Android Studio radna okolina

Glavni prozor se sakrivanjem i premještanjem alatnih traka i prozora alata može organizirati kako bismo dobili više prostora.

### 3.1.3. Opisna datoteka (.xml)

XML (EXtensible Markup Language) predstavlja jezik za opisivanje podataka. Prilikom stvaranja XML-a, glavni poticaj stvaranju je bio stvoriti jezik koji će i ljudi i računala moći jednostavno čitati. Kako bi se takva ideja mogla realizirati potrebno je odgovarajući sadržaj uokviriti odgovarajućim oznakama koje ga opisuju. XML je danas vrlo raširen te se ne koristi samo za označavanje podataka, nego također i za razmjenu i pohranu podataka te za izradu novih specijaliziranih jezika za označavanje. Standardizacijom XML standardiziranog jezika bavi se World Wide Web Consortium [11].

### **3.1.4. Java datoteka (.java)**

Kako bi svaka aktivnost koju dodamo u .xml datoteku imala svoju funkciju koristimo programski jezik Java. Prema zadanim postavkama projekt sadrži izvornu *MainActivity.java* datoteku koja sadrži klase i aktivnosti koje se pokreću prilikom pokretanja same aplikacije. *MainActivity.java* je stvarna datoteka aplikacije koja se u konačnici pretvara u izvršnu datoteku i pokreće samu aplikaciju. U *MainActivity.java* datoteci se učitava svaka pojedina aktivnost iz opisnog koda te se tako i ostvaruje komunikacija između korisnika i same aplikacije. Unutar ove datoteke moguće je učitati gotove funkcije koje uvelike skraćuju vrijeme potrebno za razvoj aplikacije te time i olakšavaju posao.

### **3.1.5. AVD**

*Android Virtual Device* (AVD) je alat koji razvojnim programerima omogućuje testiranje aplikacije simulacijom stvarnih mogućnosti uređaja. Korištenje AVD-a znatno se štedi vrijeme jer se pojedine mogućnosti aplikacije se ne moraju testirati na stvarnim uređajima. Za korištenje AVD upravitelja potrebno je stvoriti AVD i pokrenuti aplikaciju na računalu.

### **3.1.6. Programski jezik Java**

Java predstavlja objektno orijentirani programski jezik koji je razvijen 1991. godine u Sun Microsystems. Programski jezik Java zamišljen je kao jezik opće namjene za razvoj web, interaktivnih i poslovnih aplikacija. Jedna od najvećih prednosti Java programskog jezika je ta što neovisno o različitim arhitekturama, jednom napisan program u Javi može se pokretati na različitim operacijskim sustavima. Navedenu prednost omogućuje JVM (*Java Virtual Machine*) koji predstavlja hipotetsko računalo na kojem se izvršava kod napisan u Javi.

### **3.1.7. Parse Server**

Parse Serve je eng. *open source* inačica Parse backenda koja se može implementirati na bilo koju infrastrukturu koja može pokrenuti Node.js. Parse Server radi u okviru web Express aplikacija. Može se dodati već postojećim aplikacijama ili se može koristiti sam. Izravno koristi MongoDB, ali ne ovisi o bazi podataka pohranjenima u Parseu.



## 3.2. Programsko rješenje korisničkog sučelja

Za upravljanje korisničkim sučeljem koristimo opisni kod koji je sadržan unutar .xml datoteka. Opisni programski kod upravlja aktivnostima (zaslونima) aplikacije koji se prikazuju korisniku, definira izgled i smještaj alata potrebnih prilikom interakcije aplikacije i korisnika. Aplikacija sadrži četiri glavne aktivnosti, *MainActivity* za registraciju i prijavu korisnika, *NewActivity* za dodavanje nove aktivnosti, *ActivityList* za prikaz dosadašnjih aktivnosti

Programski kod 3.1 prikazuje opisni kod polja za unos lozinke koji se nalazi unutar glavne aktivnosti koja se otvara prilikom pokretanja aplikacije. Metoda *layout* nam omogućuje smještanje *Edit Text* metode unutar aktivnosti na mjesto koje mi želimo.

```
<EditText
    android:id="@+id/passwordEditText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="103dp"
    android:ems="10"
    android:hint="Lozinka"
    android:inputType="textPassword"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.503"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.239" />
```

Programski kod 3.1. Opisni kod polja za unos lozinke

Unutar jedne aktivnosti možemo imati više *EditText* metoda. Svaka *EditText* metoda mora imati svoj ID po kojemu možemo razlikovati dvije naizgled jednake metode.

Osim *EditText* metode, početni zaslon sadrži još jednu *EditText* metodu za unos lozinke, *Button* metodu za prijavu ili registraciju korisnika te *TextView* metodu kojom korisnik bira želi li obaviti aktivnost prijave ili registracije. Svaka od navedenih metoda ima svoj vlastiti ID.

*NewActivity* aktivnost omogućuje korisniku detaljan unos nove aktivnosti. *NewActivity* aktivnost sadrži nekoliko *EditText* metoda za unos površine, korištenih sredstava za zaštitu biljke, potrošnje goriva i djelatnosti.

Programski kod 3.3 prikazuje opisni kod gumba Spremi. Ovakav opisni kod omogućuje smještaj *Button* metode unutar aktivnosti, te opisuje izgled i smještaj unutar iste. Gumb Spremi sprema novostvorenu aktivnost u bazu podataka kako bi navedenu aktivnost mogli prikazati na listi dosadašnjih aktivnosti.

```
<Button
    android:id="@+id/saveButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:onClick="createActivity"
    android:text="Spremi"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.852" />
```

Programski kod 3.3. Opisni kod gumba Spremi

Programski kod 3.4 prikazuje opisni kod *ListView* metode koja se nalazi unutar *ActivityList* aktivnosti. Navedena aktivnost korisniku pruža uvid u dosadašnje obavljene aktivnosti. Osim *ListView* metode unutar *ActivityList* aktivnosti nalazi se i *Button* metoda čiji se opisni kod nalazi na već prikazanog programskom kodu 3.3.

```
<ListView
    android:id="@+id/listView"
    android:layout_width="368dp"
    android:layout_height="407dp"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    app:layout_constraintBottom_toTopOf="@+id/novaAktivnostButton"
    app:layout_constraintTop_toTopOf="parent"
    tools:ignore="MissingConstraints"
    tools:layout_editor_absoluteX="8dp" />
```

Programski kod 3.4 Opisni kod *ListView* metode

Programskim kodom 3.5 prikazan je opisni kod gumba za novu aktivnost. Gumb za stvaranje nove aktivnosti ima svoju *onClick* metodu na čiji se poziv obavlja određena funkcionalnost. Klikom na navedeni gumb korisniku se otvara zaslon za dodavanje nove aktivnosti na svoju listu aktivnosti.

```
<Button
    android:id="@+id/novaAktivnostButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:onClick="newActivity"
    android:text="@string/nova_aktivnost"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.165"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.929" />
```

Programski kod 3.5 Opisni kod gumba Nova aktivnost

*Register* Button opisan Programskim kodom 3.6 smješten jer unutar glavne aktivnosti te se pojavljuje prilikom pokretanja aplikacije. Daje korisniku mogućnost registracije novog korisnika. Isto tako programskim kodom 3.6 prikazan je i *Prijavi se* *Button* za prijavu već postojećih korisnika

```

<Button
    android:id="@+id/Register"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="162dp"
    android:text="Button"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.501"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.386"
    tools:text="Prijavi se" />

```

```

<Button
    android:id="@+id/SignUp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="236dp"
    android:text="Sign up"/>

```

Programski kod 3.6 Opisni kod gumba Registriraj se i Prijavi se

### 3.3. Programsko rješenje baze podataka te analize i praćenja aktivnosti i troškova

Prilikom pokretanja aplikacije korisnik se mora prijaviti u njenu bazu podataka. Prilikom prvog pokretanja aplikacije korisnik se treba registrirati kako bi dalje mogao koristiti aplikaciju. Registraciju novih korisnika omogućuje gumbo Registriraj se i TextView Prijavi se. Klikom na TextView Prijavi se, gumb Registriraj se i TextView Prijavi se mijenjaju svoje funkcije, gumb obavlja funkciju prijave korisnika, a TextView registraciju. Programski kod 3.7 prikazuje *onCreate* metodu početnog zaslona aplikacije. *onCreate()* metoda izvršava se prilikom pokretanja aplikacije. Elementi korisničkog sučelja kojima se odrađuje prijava i registracija korisnika su EditText polja za unos korisničkog imena i lozinke, te već navedeni TextView i Button za prijavu odnosno registraciju korisnika. Programski kod prikazuje aktivnosti

prepoznavanja klika na zaslону, prepoznavanje klika na TextView-u radi zamjene prijave i registracije novog korisnika.

```
@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ButterKnife.bind(target: this);
}
```

Programski kod 3.7. Izvorni *onCreate* metode početnog zaslona

Programski kod 3.8 prikazuje izvorni kod *signUp* metode za registraciju korisnika. *signUp* metoda koristi *EditText* elemente korisničkog sučelja za unos korisničkog imena i lozinke, te nakon toga unesene podatke koristi za kreiranje novog korisnika unutar baze podataka. Ukoliko je prijava ili registracija korisnika uspješno obavljena otvara se poziva se metoda *showActivityList()*; koja otvara novi zaslón i prikazuje dosadašnje unesene aktivnosti korisnika.

```
@OnClick(R.id.SignUp)
public void signUp(){

    String userName = userNameInput.getText().toString();
    String password = passwordInput.getText().toString();

    ParseUser user = new ParseUser();
    user.setUsername(userName);
    user.setPassword(password);

    user.signUpInBackground((e) -> {
        if(e == null){
            Log.d(TAG, msg: "Sign up successful.");
        }
        else {
            Log.d(TAG, msg: "Sign up successful. " + e.getMessage());
        }
    });
}
```

Programski kod 3. 8 Izvorni kod *signUp* metode

Prilikom pokretanja aplikacije korisniku se otvara zaslón na kojem se prikazuju dva polja za unos korisničkog imena i lozinke, gumb Registriraj se i gumb Prijavi se. Ukoliko je korisnik jedan od već postojećih korisnika klikom na gumb Prijavi se, korisnik se logira na svoj račun, dok klikom na gumb Sign Up on se registrira kao novi korisnik.

```

@OnClick(R.id.Register)
public void register(){
    String userName = userNameInput.getText().toString();
    String password = passwordInput.getText().toString();

    ParseUser.logInInBackground(userName, password, (user, e) -> {
        if(user != null){
            currentUser = user;
            Toast.makeText(getApplicationContext(), text: "Hello " + currentUser.getUsername(), Toast.LENGTH_SHORT).show();
            goToNextActiviy();
        }
        else {
            Toast.makeText(
                getApplicationContext(),
                text: "Sorry, user not found.",
                Toast.LENGTH_SHORT).show();
        }
    });
}

```

### Programski kod 3. 9 Izvorni kod *register* metode

Programski kod 3.10 prikazuje spremanje nove aktivnosti u bazu podataka. Nakon što je korisnik klikom na određeni gumb otvorio zaslon (aktivnost) za stvaranje nove korisnikove aktivnosti, prikazuju mu se elementi za unos i detaljan opis svoje djelatnosti. Korisniku se nude polja za unos imena i površine parcele koju trenutno obrađuje, polje za unos djelatnosti koje obavlja, polje za unos datuma obavljanja pojedine djelatnosti te polja za unos pesticida i njegove koncentracije te potrošnje goriva prilikom obrade navedene parcele. *Button* Spremi sprema novi aktivnost u bazu podataka kako bi ju kasnije mogli prikazati na listi djelatnosti.

```

final ParseObject aktivnosti = new ParseObject( theClassName: "Aktivnosti");
aktivnosti.put("Parcela", parcela);
aktivnosti.put("Povrsina", površina);
aktivnosti.put("Kultura", kultura);
aktivnosti.put("Datum", datum);
aktivnosti.put("Sredstvo", sredstvo);
aktivnosti.put("Koncentracija", koncentracija);
aktivnosti.put("Potrosnja", potrosnja);
aktivnosti.put("Djelatnost", djelatnost);

aktivnosti.saveInBackground(new SaveCallback() {
    @Override
    public void done(ParseException e) {
        if (e == null) {
            Toast.makeText(getApplicationContext(), text: "Aktivnost kreirana", Toast.LENGTH_SHORT).show();

            goToNextActiviy();
        } else {
            Toast.makeText(getApplicationContext(), text: "Failed", Toast.LENGTH_SHORT).show();
        }
    }
});

```

### Programski kod 3. 10 Izvorni kod kreiranja nove korisnikove aktivnosti

Programski kod 3.11. prikazuje deklaraciju svih elemenata korisničkog sučelja potrebnih za stvaranje nove korisnikove aktivnosti. Elementi koji omogućuju korisniku detaljan opis nove aktivnosti su *EditText* metode u koje korisnik upisuje tražene podatke, gumb za spremanje nove aktivnosti.

```
@BindView(R.id.parcelaEditText)
EditText parcelaInput;
@BindView(R.id.povrsinaEditText)
EditText povrsinaInput;
@BindView(R.id.kulturaEditText)
EditText kuturaInput;
@BindView(R.id.datumEditText)
EditText datumInput;
@BindView(R.id.sredstvoEditText)
EditText sredstvoInput;
@BindView(R.id.koncentracijaEditText)
EditText koncentracijaInput;
@BindView(R.id.potrosnjaEditText)
EditText potrosnjaInput;
@BindView(R.id.djelatnostEditText)
EditText djelatnostInput;
```

Programski kod 3. 11. Deklaracija elemenata korisničkog sučelja

Programski kod 3.12 prikazuje metodu dohvaćanja korisnikovih unosa. Metoda *getAgroActivities()* dohvaća korisnikove aktivnosti iz baze te ih sprema u novu listu.

```
public void getAgroActivities() {
    List<ParseObject> agroActivities = new ArrayList<>();

    ParseQuery<ParseObject> query = ParseQuery.getQuery("Aktivnosti");
    query.whereEqualTo("Dan Stemkoski");
    query.findInBackground(new FindCallback<ParseObject>() {
        public void done(List<ParseObject> agroActivities, ParseException e) {
            if (e == null) {
                Log.d( tag: "Aktivnosti", msg: "Retrieved " + agroActivities.size() + " aktivnosti");
                display(agroActivities);
            } else {
                Log.d( tag: "Aktivnosti", msg: "Error: " + e.getMessage());
            }
        }
    });
}
```

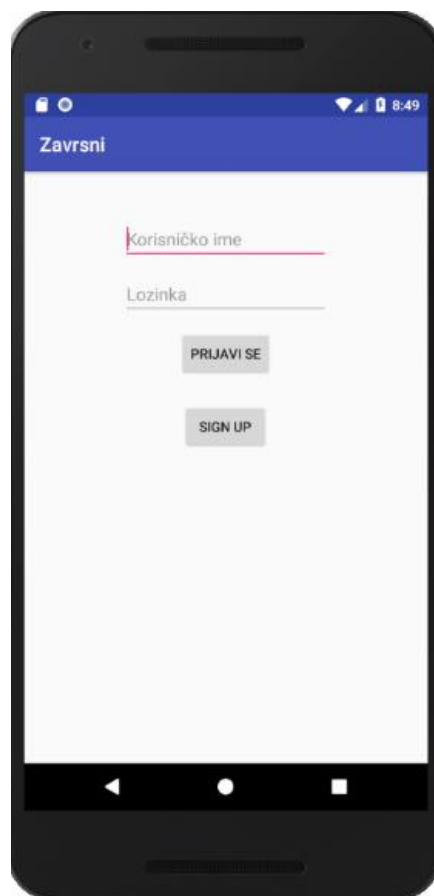
Programski kod 3. 12. Izvorni kod *getAgroActivities* metode

## 4. NAČIN KORIŠTENJA, ISPITIVANJE I ANALIZA RADA APLIKACIJE

U narednom poglavlju bit će objašnjeni načini korištenja aplikacije, prikaz rada aplikacije na nekim primjerima i na samom kraju analiza rada same aplikacije.

### 4.1. Način korištenja

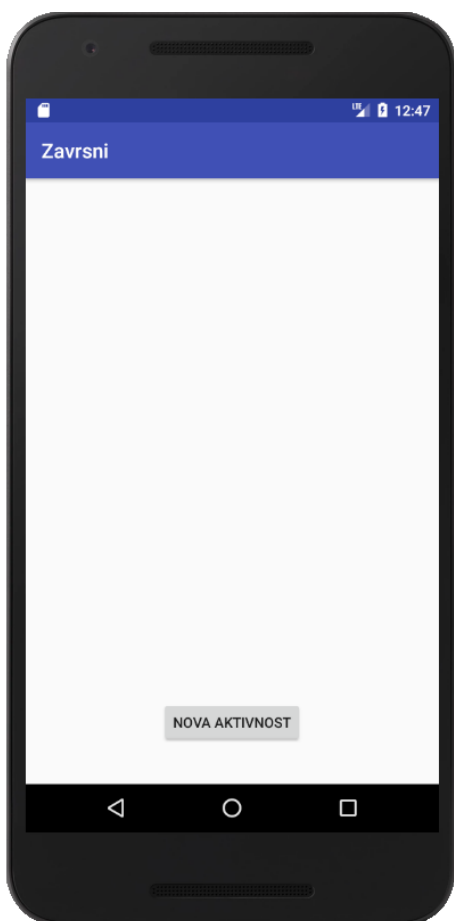
Kako bismo testirali aplikaciju njezin rad ćemo prikazati na njenim glavnim funkcionalnostima, prijava korisnika, dodavanje nove aktivnosti te prikaz liste dosadašnjih aktivnosti. Nakon što korisnik pokrene aplikaciju na zaslonu mu prikazuje aktivnost za prijavu odnosno registraciju korisnika kao što prikazuje slika 4.1. Ukoliko je korisnik nije već postojeći korisnik, ispod gumba za prijavu nalazi se gumb za registraciju novih korisnika.



Slika 4.1. Početni zaslon aplikacije

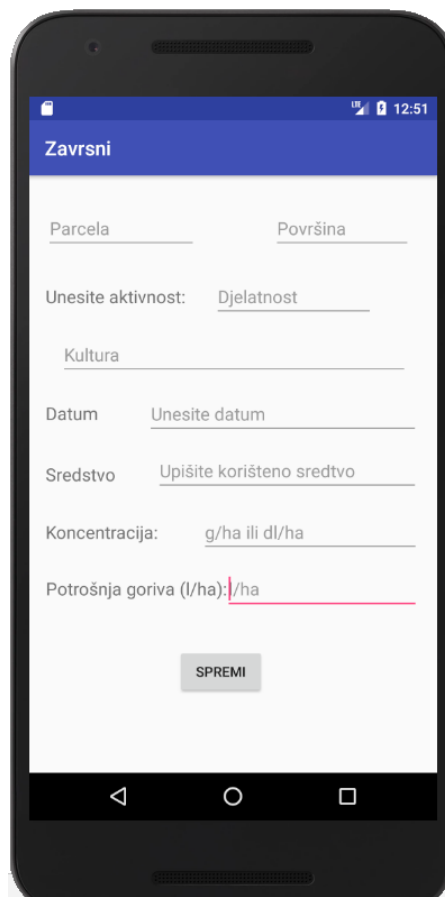


Ukoliko se korisnik uspješno prijavio prikazuje se zaslon sa svim dosadašnjim aktivnostima. Ako je trenutni korisnik novi korisnik prikazuje mu se prazna lista te mu nudi mogućnost kreiranja nove poljoprivredne aktivnosti što i prikazuje slika 4.2.



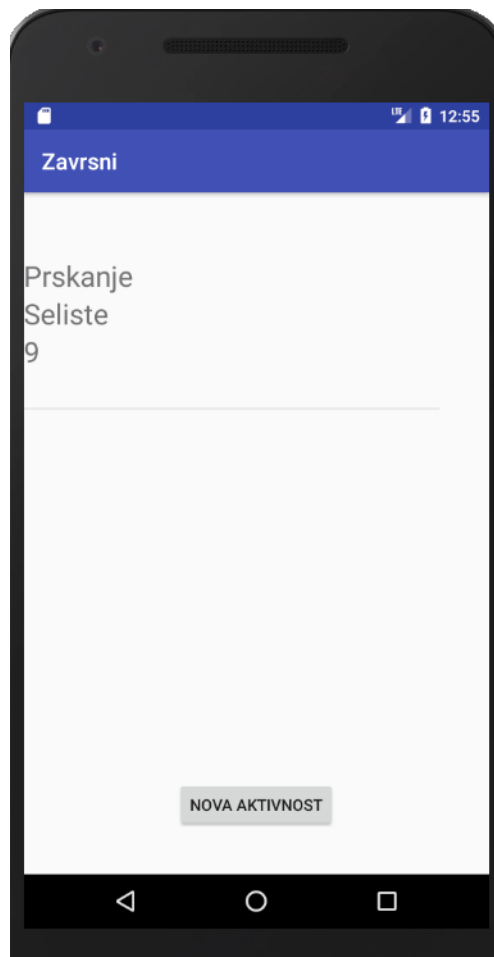
Slika 4. 2. Prikaz prazne liste novog korisnika

Klikom na gumb Nova aktivnost otvara se zaslon koji korisniku omogućuje unos svih potrebnih parametara za uspješno praćenje rada svoga gospodarstva. Zaslon za kreiranje nove aktivnosti prikazan je slikom 4.3. Za kreiranje nove aktivnosti korisnik treba popuniti potrebna polja za unos kako bi mogao što bolje opisati pojedinu djelatnost.



Slika 4. 3. Kreiranje nove aktivnosti

Nakon unosa pojedinih parametara potrebnih za obavljanje pojedine poljoprivredne aktivnosti, korisnik ima mogućnost spremanja nove aktivnosti. Klikom na gumb Spremi korisnik se vraća na zaslon koji mu prikazuje listu dosadašnjih aktivnosti gdje može vidjeti i svoju novostvorenu aktivnost. Povratak na zaslon sa listom djelatnosti korisniku prikazuje listu njegovih dosadašnjih aktivnosti, ali mu i u isto vrijeme daje mogućnost stvaranja nove aktivnosti klikom na gumb Nova aktivnost (slika 4.4).



Slika 4.4. Prikaz liste sa dosadašnjim aktivnostima

## 4.2. Prikaz rada aplikacije na primjerima

Testiranjem aplikacije bit će provjerena ispravnost njenih funkcionalnosti, dodavanje novih korisnika, spremanje novih korisnikovih aktivnosti te prikaz dosadašnjih aktivnosti unutar liste.

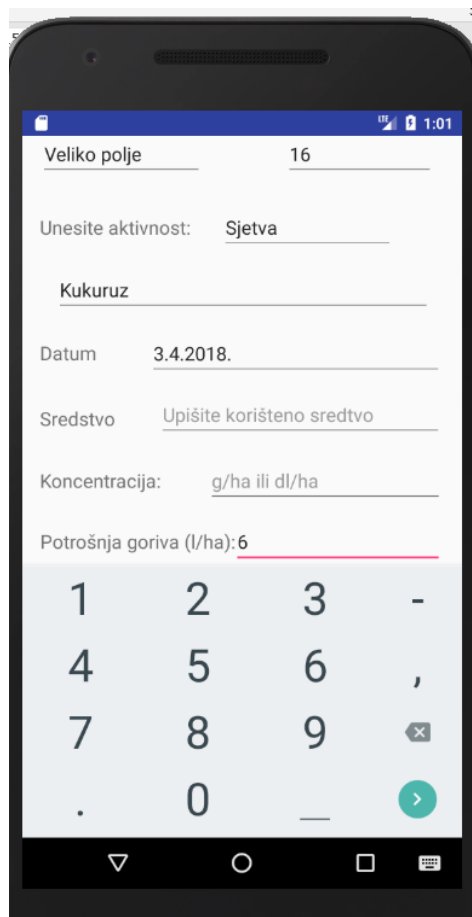
Testiranje započinjemo stvaranjem odnosno registracijom novog korisnika gdje nam se nakon uspješne registracije otvara zaslon sa praznom listom. Zaslon prijave i registracije korisnika prikazan je slikom 4.5.



Slika 4.5. Prijava korisnika

Ukoliko je korisnik uspješno registriran možemo vidjeti njegovu uspješnu registraciju na poslužitelju., dok mu se u samoj aplikaciji otvara zaslon sa listom djelatnost. Uz poruku dobrodošlice sa korisnikovim korisničkim imenom.

Trenutna korisnikova lista je prazna jer nije do sada dodao niti jednu poljoprivrednu aktivnost. Ispod prikaza liste nudi nam se gumb Nova aktivnost. Klikom na gumb nova aktivnost korisniku se otvara novi zaslon *NewActivity*.. Unutar navedenog zaslona nalaze se polja koja korisnik treba popuniti kako bi stvorio novu aktivnost. Ukoliko je korisnik popunio potrebna polja, novu poljoprivrednu aktivnost može spremiti klikom na gumb Spremi.



Slika 4.6. Unos potrebnih parametara

Klik na gumb Spremi sprema novu aktivnost u bazu što je prikazano slikom 4.4., otvara zaslon sa popisom dosadašnjih aktivnosti sprema i prikazuje novu aktivnost u listi te daje korisniku obavijest kako je nova aktivnost kreirana. Kada je lista stvorena, korisnik može kreirati novu listu klikom na gumb Nova aktivnost.



Slika 4. 7. Potvrda kreiranja nove aktivnosti

### 4.3. Analiza rada aplikacije

Testiranjem aplikacije utvrđeno je da aplikacija obavlja sve svoje zadane funkcionalnosti. Nakon uspješnog pokretanja aplikacije, korisniku se otvara zaslon za unos korisničkog imena i lozinke. Ukoliko je korisnik pravilno unio tražene podatke, klikom na gumb Prijavi se prikazuje se novi zaslon sa prikazom dosadašnjih djelatnosti uz poruku dobrodošlice. Uspješnim popunjavanjem potrebnih polja korisnik sprema novu aktivnost klikom na gumb Spremi nakon čega se izbacuje poruka o uspješnom kreiranju nove aktivnosti. Zatim se novostvorena aktivnost prikazuje korisniku na dnu liste dosadašnjih aktivnosti. S obzirom da se u aplikaciji ne prikazuju nikakve greške, cilj aplikacije je ostvaren, ali još uvijek ima prostora za nadogradnju aplikacije kako bi ona mogla olakšati poljoprivrednicima vođenje i brigu, kako o svojim usjevima tako i o svojim djelatnostima.

## 5. ZAKLJUČAK

Osnovni cilj i primjena aplikacije je pomoći poljoprivrednicima prilikom vođenja svakodnevnih djelatnosti kako bi se mogao što učinkovitije brinuti o svojim usjevima. Kroz aplikaciju je omogućena prijava postojećih i registracija novih korisnika *register* i *SignUp* metodama, kreiranje nove aktivnosti i spremanje iste te dohvaćanje i prikaz svih dosadašnjih aktivnosti u obliku liste pomoću metode *getActivities*. Korisnik svojom prijavom i kreiranjem svoje nove aktivnosti sprema svoje podatke na udaljenu bazu podataka koje zatim može dohvatiti i prikazati u obliku liste.

Testiranjem je utvrđeno kako aplikacija uspješno izvršava sve svoje zadane funkcionalnosti. Aplikacija omogućuje kontinuirano praćenje svih korisnikovih aktivnosti od sjetve do skidanja ploda. Nakon testiranja i korištenja aplikacije tijekom nekoliko godina, cilj je izrada modela po kojem bi aplikacija mogla podsjećati svoje korisnike koju djelatnost bi trebali obaviti u koje vrijeme.

# LITERATURA

- [1] L.M. Cespedes, Working in development, How ICT Tools Are Improving Efficiency Of Agricultural Development, The Guardian, 2013., dostupno na: <https://www.theguardian.com/global-development-professionals-network/2013/jan/24/data-collection-evaluation-technology-agriculture> [lipanj 2018.]
- [2] Information and Communications Technology In Agriculture, 2017., dostupno na: [https://en.wikipedia.org/wiki/Information\\_and\\_communications\\_technology\\_in\\_agriculture](https://en.wikipedia.org/wiki/Information_and_communications_technology_in_agriculture)
- [3] Agrivi, Agrivi, 2014., dostupno na: <https://play.google.com/store/apps/details?id=com.agrivi.agrivi> [lipanj 2018.]
- [4] Pioneer, Agro-Assist, Pioneer Hi-Bred Magyarország, 2015., dostupno na: <https://play.google.com/store/apps/details?id=com.pioneer.agroassist>
- [5] FarmLogs, FarmLogs, Michigan, SAD, 2012., dostupno na: <https://play.google.com/store/apps/details?id=io.trigger.forge633ed212c46d11e185cd12313d1adcbe> [lipanj 2018.]
- [6] Meet Android Studio, Developers, 2018., dostupno na: <https://developer.android.com/studio/intro/> [lipanj 2018.]
- [7] 5 Essential Android Development Tools for 2017, Android Community, 2017., dostupno na: <http://androiddeveloper.galileo.edu/2017/02/24/5-essential-android-development-tools-in-2017/>
- [8] PROGRAMIRANJE U JAVI, LV1, FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK, 2017.
- [9] XML, 2017, dostupno na: <https://hr.wikipedia.org/wiki/XML>
- [10] <https://ieeexplore.ieee.org/document/7892470/>
- [11] B. Hardy, B. Phillips, Android Programming: The Big Nerd Ranch Guide, Big Nerd Ranch, Inc, Atlanta, 2013.
- [12] Agro Klub, 2015., dostupno na: <https://www.agroklub.com/sortna-lista/zitarice/kukuruz-115/> [lipanj 2018.]
- [13] Agro Klub, 2015., <https://www.agroklub.com/sortna-lista/povrce/kupus-142/>
- [14] I. Medved, Ljetna sadnja kupusa, Agroportal.hr, 2017., dostupno na: <http://www.agroportal.hr/povrtlarstvo/1816> [lipanj 2018.]



# SAŽETAK

Cilj ovog završnog rada je modeliranje i izrada mobilne aplikacije koja može pomoći poljoprivrednicima prilikom vođenja svojih svakodnevnih djelatnosti, što je i ostvareno. Korisniku je omogućen unos svih parametara koji definiraju pojedinu djelatnost te trajno spremanje istih na udaljenu bazu podataka. Aplikacija je izrađena u razvojnom okruženju Android Studio, uz opisni jezik kojim definiramo smještaj pojedinih elemenata na korisničkom sučelju i izvorni jezik za izvršavanje svih funkcionalnosti aplikacije. Testiranjem aplikacije utvrđeno je da je korisniku omogućeno pokretanje same aplikacije te unos korisničkog imena i lozinke, kao i ostale funkcionalnosti koje su zadane. Aplikacija se u budućnosti može nadgraditi nekim novim aktivnostima koje bi korisniku mogle još više olakšati dosadašnje aktivnosti.

**Ključne riječi:** Android Studio, informacijske tehnologije, Java, mobilna aplikacija,

poljoprivreda

## **ABSTRACT**

The aim of this final work is to model and develop a mobile application that can help farmers in managing their daily activities, which is accomplished. The user is enabled to enter all the parameters that define the individual activity and permanently save them to the remote database. The app is created in the Android Studio development environment, along with a descriptive language that defines the placement of individual elements on the user interface and the native language for performing all application functionality. Testing the application found that the user was enabled to run the application itself and enter the username and password, as well as other functionality that were set. The application can be upgraded in the future with some new activities that could make the user even easier to do so.

**Keywords:** Android Studio, information technology, Java, mobile application, agriculture

# ŽIVOTOPIS

Ivana Kovačević rođena je 4.4.1996. godine u Našicama. 2011. godine završila je Osnovnu školu Ante Starčevića u Viljevu. Nakon završetka osnovne škole upisuje opću gimnaziju u Srednjoj školi Donji Miholjac koju 2014. godine i završava. Iste godine na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija upisuje smjer računarstvo, koji i dana pohađa. Posjeduje vještine rada na računalu (Word, Excel, Power Point, Internet, C, C++) te posjeduje znanje programskog jezika Java kao i rada u razvojnom okruženju Android Studio.

## **PRILOZI (na CD-u)**

Prilog 1. Dokument završnog rada

Prilog 2. Pdf završnog rada

Prilog 3. Programski projekt aplikacije