Mobilna Android aplikacija za praćenje poljoprivrednih usjeva

Horvat, Toni

Undergraduate thesis / Završni rad

2017

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://urn.nsk.hr/urn:nbn:hr:200:056529

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-06**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Sveučilišni preddiplomski studij računarstva

MOBILNA ANDROID APLIKACIJA ZA PRAĆENJE POLJOPRIVREDNIH USJEVA

Završni rad

Toni Horvat

Osijek, 2017.

ETFOS
ELEKTROTEHNIČKI FAKULTET OSIJEK
Sveučilište Josipa Jurja Strossmayera u Osijeku

Obrazac Z1 - Obrazac za ocjenu završnog rada

Osijek, 22.09.2017.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada

Ime i prezime studenta:	Toni Horvat
Smjer:	Računarstvo
Mat. br. studenta, godina upisa:	R3579, 2013.
Mentor:	Prof. dr. sc. Goran Martinović
Sumentor:	
Naslov završnog rada:	**MOBILNA ANDROID APLIKACIJA ZA PRAĆENJE POLJOPRIVREDNIH USJEVA**
Primarna znanstvena grana rada:	**Računarstvo**
Sekundarna znanstvena grana (ili polje) rada:	**Računarstvo**
Predložena ocjena završnog rada:	
Kratko obrazloženje predložene ocjene:	

Potpis sumentora:	Potpis mentora:

Dostaviti:

1. Studentska služba

Korekcija ocjene Odbora za završne i diplomske ispite

Odbor za završne i diplomske ispite Elektrotehničkog fakulteta Osijek, temeljem članka 11. Pravilnika o završnim ispitima na preddiplomskim i stručnim studijima Elektrotehničkog fakulteta Osijek donio je odluku o korekciji konačne ocjene završnog rada na ____ sjednici Odbora održanoj dana _____ godine.

Korigirana konačna ocjena završnog rada:	_____

Potpis predsjednika Odbora:

_____

Dostaviti:

1. Studentska služba

Sveučilište Josipa Jurja Strossmayera u Osijeku

ELEKTROTEHNIČKI FAKULTET OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 22.09.2017.

Ime i prezime studenta:	TONI HORVAT
Studij :	RAČUNARSTVO
Mat. br. studenta, godina upisa:	R3579, 2013.

Ovom izjavom izjavljujem da je rad pod nazivom**:**

MOBILNA ANDROID APLIKACIJA ZA PRAĆENJE POLJOPRIVREDNIH USJEVA

izrađen pod vodstvom mentora

Prof. dr. sc. Goran Martinović

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada,osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 5
1.1. Zadatak završnog rada 5
2. POSTOJEĆA RJEŠENJA I RATARSKA KULTURA 6
2.1. Postojeća rješenja 6
2.2. Idejno rješenje aplikacije 8
2.3. Ratarska kultura 9
2.3.1. Kukuruz 9
2.3.2. Pšenica 11
3. TEHNOLOGIJE POTREBNE ZA IZRADU APLIKACIJE 13
3.1. Android Studio 13
3.1.1. Izgled početnog zaslona 13
3.1.2. Opisna datoteka (.xml) 14
3.1.3. Java datoteka (.java) 15
3.1.4. AVD upravitelj 16
3.2. Programski jezik Java 17
3.3. Baza podataka(SQL Server) 18
4. PROGRAMSKI KOD APLIKACIJE 19
4.1. Opisni programski kod aplikacije (.xml) 19
4.2. Izvorni programski kod aplikacije (.java) 25
5. NAČIN KORIŠTENJA , TESTIRANJE I ANALIZA RADA APLIKACIJE 30
5.1. Korištenje aplikacije 30
5.2. Testiranje i analiza rada aplikacije 36
6. ZAKLJUČAK 40
LITERATURA 41
SAŽETAK 42
ABSTRACT 42
ŽIVOTOPIS 43

1. UVOD

Završni rad temelji se na izradi mobilne aplikacije koja će pomagati poljoprivrednicima kako bi lakše pratili svoj usjev. Također, ukoliko se na usjevu pojavi nekakva štetočina ili bolest, aplikacija će pomoći korisnicima predložiti sredstva te optimalnu količinu primjene. Nadalje, aplikacija će služiti i kao sredstvo prikupljanja određenih parametara na širem području i širem krugu korisnika kako bi se kroz nekakvo izvjesno vrijeme mogli napraviti dijagrami najučinkovitijih sredstava i gnojiva korištenih u poljoprivredi. Određivanje parametara aplikacije i svi potrebni podaci potrebni za zaštitu sjemena i usjeva te gnojiva prikupljeni su uz pomoć literature na internetu te društva agronoma u Čepinu.

Cilj aplikacije je omogućiti korisniku unos parametara svojeg usjeva i spremati ga u udaljenu bazu podataka. Isto tako prilikom unosa određenog nametnika ili bolesti korisnik će dobiti popis sredstava koja može koristiti i preporuku u kojoj količini. Nakon rada aplikacije kroz nekoliko godina dobiti će se optimalna rješenja tretiranja usjeva u svrhu povećanja prinosa i kvalitete usjeva.

U drugom poglavlju opisana su postojeća rješenja aplikacija za navedenu tematiku. Nadalje, bit će predloženo idejno rješenje aplikacije sa popisom glavnih parametara koji se koriste te opis kultura na kojima se aplikacija bazira. Naredno poglavlje 3 bavi se tematikom tehnologija koje se koriste za ostvarenje aplikacije te opisom pojedine tehnologije. Četvrto poglavlje sadrži programsko rješenje s opisom svih bitnih funkcija i parametara potrebnih za uspješan rad aplikacije. Peto poglavlje sadrži rezultate analize uspješnosti rada aplikacije i njezinu stvarnu primjenu.

1.1. Zadatak završnog rada

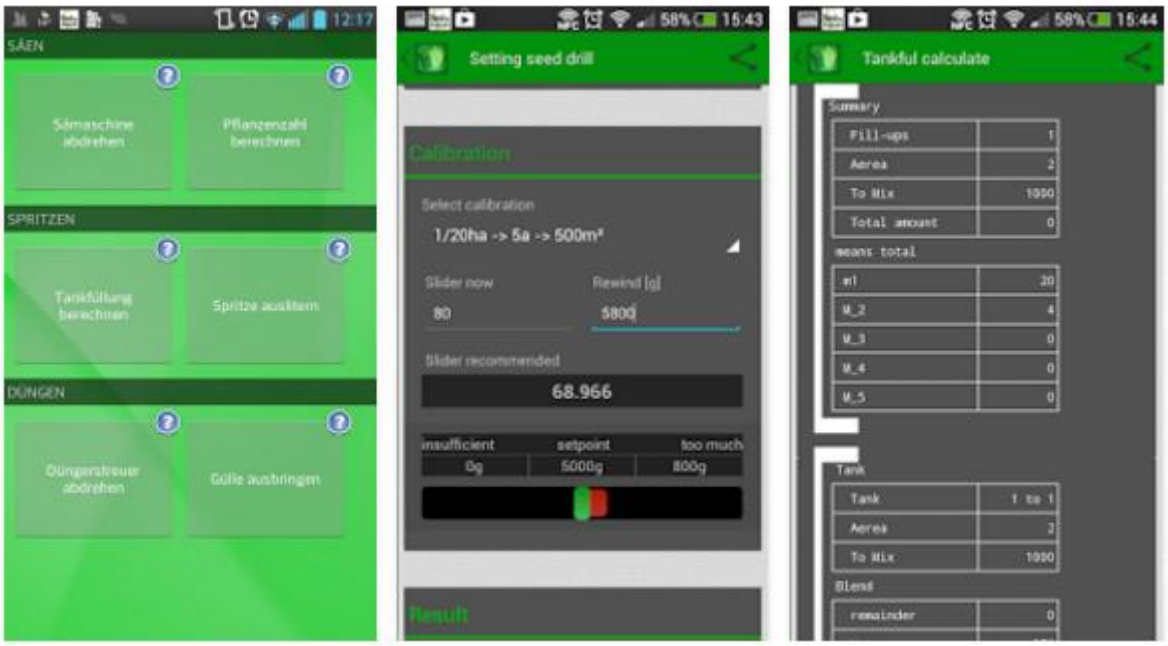
U radu je potrebno opisati i izraditi model praćenja poljoprivrednih usjeva s ciljem poboljšanja prinosa i zaštite od bolesti. Programski je potrebno ostvariti mobilnu aplikaciju za Android platformu s odgovarajućom bazom podataka koja je u mogućnosti prikupljati agrometeorološke i podatke o stanju usjeva, te generirati informacije o prinosima, zaštiti i ostale preporuke struke. Mobilnu aplikaciju potrebno je prikladno ispitati s gledišta funkcionalnosti i točnosti.

2. POSTOJEĆA RJEŠENJA I RATARSKA KULTURA

Ovo poglavlje sadrži popis već postojećih aplikacijskih rješenja za tematiku ovoga rada. Isto tako biti će navedeno idejno rješenje aplikacije te opis što će biti teško a što ne prilikom njezine daljnje realizacije. Nadalje, tema se dotiče poljoprivrede, u obliku usjeva i zaštite, stoga su potrebna predznanja i parametri dobiveni u suradnji s društvom agronoma na području Osijeka i Čepina te iz literature na internet stranicama specijaliziranim za ratarske usjeve.

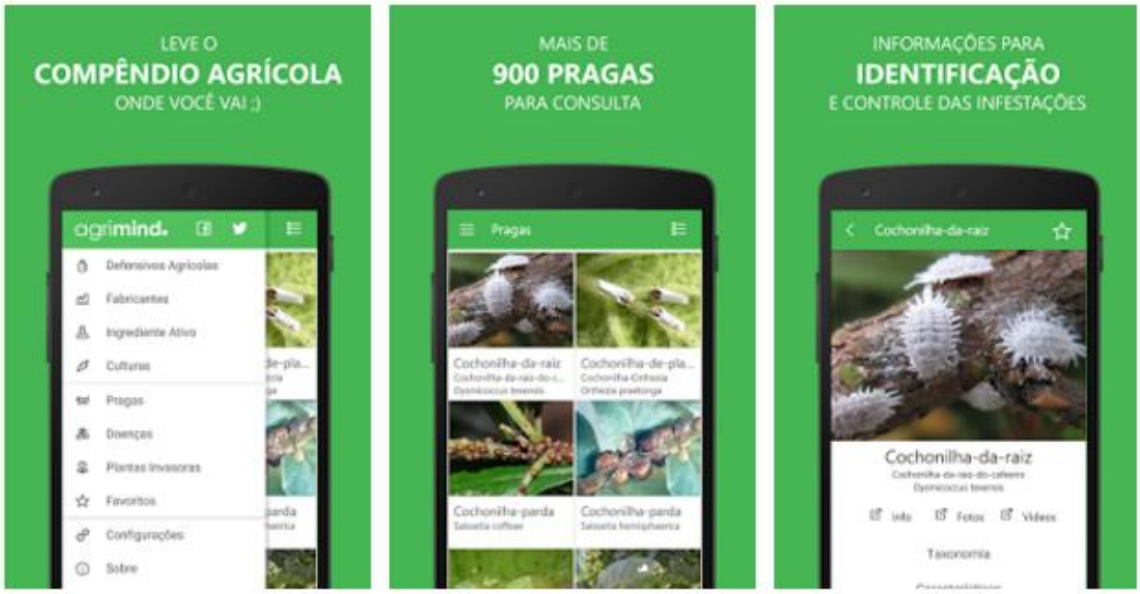
2.1. Postojeća rješenja

Trenutno u svijetu postoji više od tisuću aplikacija vezane uz poljoprivredu i ratarsku tehnologiju. Velika većina tih aplikacija je na engleskom jeziku i dostupna putem *Google Play* alata za preuzimanje aplikacija. Neke od najraširenijih aplikacija po broju preuzimanja su *Agriculture, Farming calculator PRO, Organic gardening, AgriAPP, Agricultuar Compedinum.* Aplikacija *Agriculture*[1] drži vodeće mjesto u preuzimanju aplikacija ovog tipa. Orijentirana je prema edukacijski i sadrži razne literature o pojedinim granama poljoprivrede tako što korisnika upućuje kroz procese obrade zemlje. Nadalje, obrađuje pojmove kao što su traktor, ministarstvo poljoprivrede, perad, navodnjavanje, stočarstvo, uzgoj i slično. Sljedeća aplikacija *Farming calculatorPRO*[2] je aplikacija temeljena na pomoći poljoprivredniku oko računanja primjesa zaštitnih sredstava, računanje omjera gnojiva te za ostale potrebne izračune. Na slici 2.1 prikazano je nekoliko zaslona *Farming calculator PRO* aplikacije sa njezinim mogućnostima.



Slika 2.1 Izgled aplikacije Farming calculator PRO

Zatim slijedi aplikacija *AgriAPP*[3] koja je po najbolja po ocjenama poljoprivrednih proizvođača. Pruža informacije o zaštiti usjeva, skladištenja, gnojidbi, način hranjena stoke, načini izrade stajskog gnojiva te prevencija i zaštita od bolesti stoke. Isto tako aplikacija ima pristup internetu te umrežuje razne poljoprivredne proizvođače diljem svijeta. Na kraju aplikacija *Agricultuar Compedinum*[4] tvrtke *Agrimind* koja za razliku od ranije navedenih stoga ima nešto manji broj preuzimanja. Aplikacija sadrži informacije od preko 900 različitih štetočina i načina kako se obraniti usjev od njih. Izgled zaslona aplikacije prikazan je na slici 2.2 .



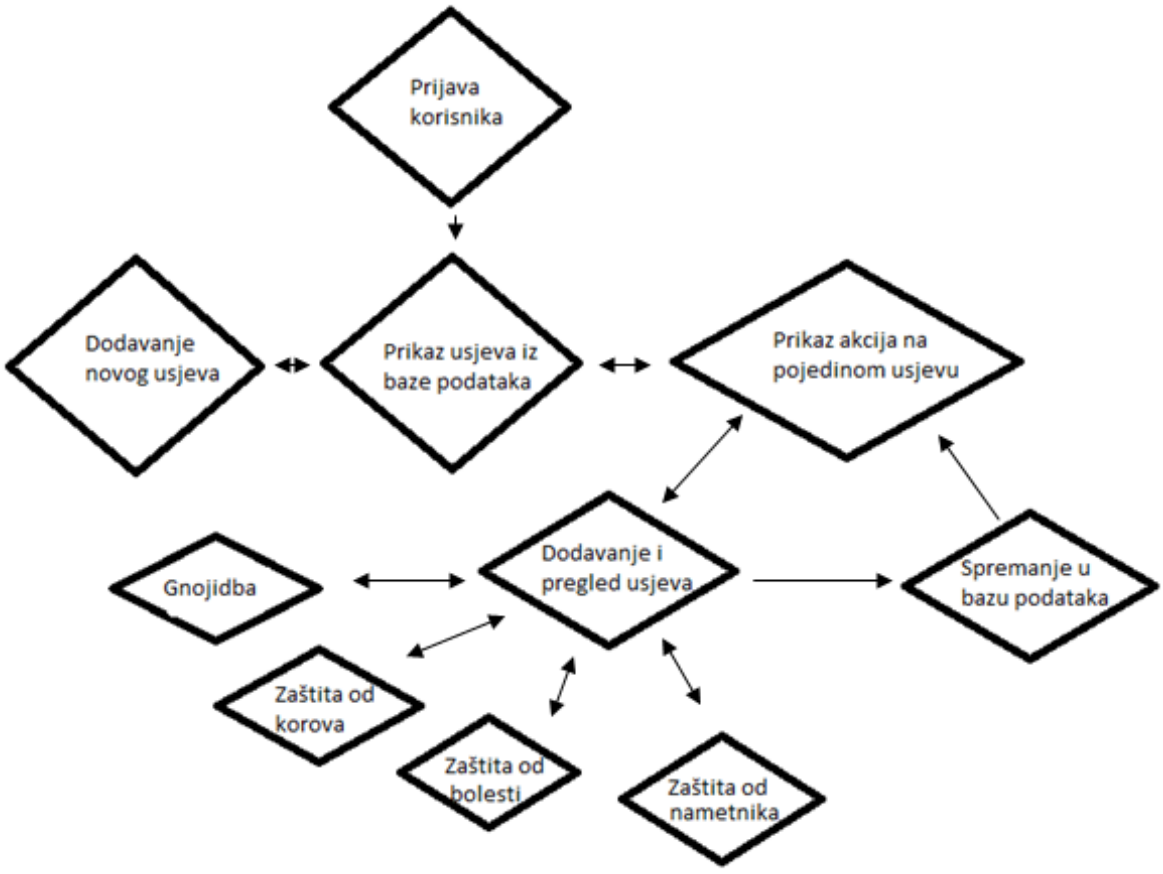
Slika 2.2 Izgled aplikacije Agricultuar Compedinum

Sve aplikacije prethodno navedene su korisne i pomažu korisnicima na njihovim usjevima i vrtovima. Glavni nedostatak korištenja tih aplikacija na našem području je engleski jezik. Većina poljoprivrednika u Republici Hrvatskoj ne poznaje engleski jezik te je vrlo teško razumjeti što aplikacija nudi. Stoga je potrebno napravit korisnu aplikaciju za naše područje primjene što je ostvareno aplikacijom ovog završnog rada koja nudi i neke nove mogućnosti kao što je spremanje usjeva u bazu podataka.

2.2. Idejno rješenje aplikacije

Kako bi se aplikacija ostvarila potrebno je definirati sve parametre koji su potrebni za interakciju između korisnika i onoga sto aplikacija nudi. Ideja je da aplikacija pomogne poljoprivrednicima tako što će im omogućiti da vrlo lako unose određene parametre svih svojih usjeva kao što su vlažnost tla, količina sjemena, klijavost i čistoća što se tiče sijanja te količina prihoda, vlažnost sjemena, čistoća sjemena i kakvoća što se tiče žetve usjeva.

Početni zaslon služi za prijavu korisnika u sustav udaljene baze podataka odnosno registraciju ukoliko korisnik još ne posjeduje račun. Nakon uspješne prijave aplikacija uzima sve postojeće podatke iz baze podataka navedenog korisnika i prikazuje mu u obliku liste. Također, korisnik može dodati i nove podatke o usjevu s trenutnim parametrima. Prilikom pritiska na određeni usjev otvara se novi zaslon koji sadrži podatke označenog usjeva kao što je gnojidba, zaštita od nametnika, zaštita od bolesti i korova. Isto tako, prilikom pretrage određenog nametnika ili bolesti aplikacija će ponuditi preporuku koja sredstva bi trebalo koristiti te optimalnu dozu. Slikoviti prikaz ideje aplikacije prikazan je dijagramom toka na slici 2.3



Slika 2.3 Dijagram toka zaslona aplikacije

2.3. Ratarska kultura

Aplikacija će biti bazirana na dvije ratarske kulture, kukuruz i pšenicu najraširenije u kontinentalnom dijelu Hrvatske. Informacije potrebne pri izradi aplikacije prikupljene su putem internet stranica te Društva agronoma u Čepinu.

2.3.1. Kukuruz

Prema Internet stranici Agro-klub[5] kukuruz je jednogodišnja biljka jarog tipa razvića, a njegova dužina vegetacije od nicanja do pune zriobe ovisi od osobine sorte, odnosno hibrida, s jedne strane, i uvjeta uzgoja, s druge strane. Po dužini vegetacije sve hibride kukuruza možemo razvrstati u rane, srednje rane i kasne vegetacije. Pod veoma ranim hibridima podrazumijevaju se hibridi s vremenom vegetacije od 90-110 dana, a pod srednje ranim s vegetacijom od 120-135 dana te kasnim od 135-145 dana. Osnovni gospodarski značaj kukuruza proizlazi iz svojstava same biljke, raznovrsnosti upotrebe i obima proizvodnje. Gotovo svi dijelovi biljke kukuruza mogu poslužiti za preradu, pa upravo to daje kukuruzu poseban ekonomski značaj. Danas se proizvodi više od 500 različitih industrijskih prerađevina od kukuruza (prehrambeni i ljekarski proizvodi, farmaceutska i kozmetička sredstva, razni napitci, tekstilni i kemijski proizvodi). Zrno kao osnovna sirovina u pripravljanju koncentrirane stočne hrane ima izuzetno veliku važnost jer sadrži 70-75 % ugljikohidrata, 10 % bjelančevina, oko 5 % ulja, 15 % mineralnih tvari, te 2,5 % celuloze.[6]

Nicanje kukuruza s obzirom na temperaturu traje kao što slijedi:

- 5 do 6 dana ako je temperatura 21 °C;

- 8 do 10 dana ako je temperatura 15,5 do 18 °C;

- 12 dana ako je temperatura 14 °C;

- čak 18 do 20 dana ako je temperatura 10 do 13 °C.

Za dobar prinos potrebno je oko 500 do 600 mm vode u toku vegetacije, a najveća je potreba u fazi svilanja - oplodnji i nalijevanju zrna. Najviše mu pogoduju plodna, duboka, propusna i rastresita tla te tla koja zadržavaju dosta vode, a manje su pogodna lagana pjeskovita i teška glinasta tla. Ne preporučuje se uzgoj na kiselim tlima te na tlima s nagibom većim od 5 %.

Pravilo gnojidbe sastoji se od sljedećih točaka:

1.Oranjem zaorati gnojiva s povišenim sadržajem fosfora (P_2O_5) i kalija (K_2O) (2/3 tih elemenata) (NPK 7:20:30, 6:18:36, 10:20:30 i sl.) te jedan dio dušika (N) kroz UREU

2. Predsjetveno gnojiti startnim gnojivom s izbalansiranim sadržajem svih hranjiva (preostala 1/3 P i K) (NPK 15:15:15 i sl.) te drugim dijelom N gnojiva kroz UREU

3. U ranom proljetnom porastu obaviti prihranu s KAN-om.

Bolesti na kukuruzu koje kod nas prevladavaju jesu: trulež stabljike i korijena, trulež klipa te siva pjegavost lista. Njih se može suzbiti širom plodosmjenom i izmjenom kukuruza s kulturama koje nisu iz porodica trava: lepirnjače, uljana repica, šećerna repa i dr. Također, mjere zaštite mogu biti i uzgoj otpornih hibrida, a sjeme se obvezatno zaštićuje kemijskim sredstvima već pri doradi[6]. U tablici 2.1 prikazana je tablica zaštite kukuruza od korova[7].

Tablica 2.1 Zaštita kukuruza

PREPARAT	NAMJENA	FAZA RAZVOJA	DOZA
Eradikane extra 6E ili Surpas 6, 7 E	jednogodišnji travni i širokolisni korovi	prije sjetve uz inkorporaciju	4-6 l/ha 5-7 l/ha
Primextra Gold 720 SC	jednogodišnji travni i širokolisni korovi	nakon sjetve prije nicanja	3-4 l/ha
Radazin Extra	jednogodišnji travni i širokolisni korovi	nakon sjetve prije nicanja	4-5 l/ha
Merlin	jednogodišnji travni i širokolisni korovi	nakon sjetve prije nicanja	100-140 g/ha
Gesaprim 500 FL ili Radzin T-50 + Dual Gold 960SC ili Proponit ili Bravo ili TROPHY ili Guardian ili Legionar ili Frontier	jednogodišnji širokolisni i neki uskolisni korovi + jednogodišnji travni i neki širokolisni korovi	nakon sjetve prije nicanja do eventualno tri pera kukuruza	2-2,5 l/ha 2-2,5 l/ha + 1-1,3 l/ha 2 l/ha 3 l/ha 2-2,5 l/ha 2-2,5 l/ha 3 l/ha 1,4-1,6 l/ha
Banvel 480 S	jednogodišnji i višegodišnji širokolisni korovi	nakon nicanja faza 5-6 listova	0,5-0,7 l/ha
Deherban A ili Dikocid	jednogodišnji i višegodišnji širokolisni korovi	nakon nicanja faza do 6 listova	2 l/ha
Starane 250 EC	slak, kupina, štir	nakon nicanja	0,8 l/ha
Cambio	jednogodišnji širokolisni korov	nakon nicanja	2-2,5 l/ha

Pardner	jednogodišnji i višegodišnji širokolisni korov	nakon nicanja	1-1,5 l/ha
Motivel + okvašivač	višegodišnji uskolisni i jednogodišnji korovi	nakon nicanja	1-1,25 l/ha ili split aplikacija
Tell WP 75 + okvašivač	divlji sirak, pirika, loboda, štir, ambrozija	nakon nicanja	40 g/ha
Tarot 25 Wg	višegodišnji uskolisni i jednogodišnji korovi	nakon nicanja	30-60 g/ha

2.3.2. Pšenica

Pšenica je jedna od najraširenijih i najvrjednija žitarica u prehrani čovječanstva. Prilikom razvoja vlati pšenice narastu i do 1,5 metar te prilikom dozrijevanja poprime žuti oblik. Ima široku primjenu u prehrambenoj industriji ponajviše u proizvodnji kruha.

Za gnojidbu pšenice tlu treba osigurati najmanje između 140 - 200 kg/ha dušika, 70 - 130 kg/ha fosfora i 80 - 140 kg/ha kalija. Korištenjem tih doza tlo neće izgubiti na hranjivost nego će ostati hranjivije nakon žetve. Gnojidba pšenice događa se u nekoliko faza:
1. Predsjetvena gnojidba u kojoj se priprema tlo za početak rasta pšenice te ujedno i najefektivnija gnojidba
2. Osnovna gnojidba
3. Gnojidba u vegetaciji kada se puni sjeme pšenice[8]

Zaštita od bolesti pšenice odvija se jednom do maksimalno dva puta i to u fazi jednog do dva koljenca te u fazi kada se pšenica klasa pred cvatnju. U tablici 2.2 prikazana je tablica zaštite pšenice od bolesti[9].

Tablica 2.2 Zaštita pšenice

PREPARAT	NAMJENA	FAZA RAZVOJA	DOZA
Flamenco	bolesti lista stabljike i klasa	od početka klasanja do početka cvatnje	1,25 – 1,5 l/ha
Folicur BT	bolesti lista stabljike i klasa	od početka klasanja do početka cvatnje	1 l/ha
Alto Combi	bolesti lista stabljike i klasa	od početka klasanja do početka cvatnje	0,5 l/ha
Duet	bolesti lista stabljike i klasa	od početka klasanja do početka cvatnje	1 l /ha
Impakt C	bolesti lista stabljike i klasa	od početka klasanja do početka cvatnje	1,2 l/ha
Tilt CB	bolesti lista stabljike i klasa	od početka klasanja do početka cvatnje	2 l/ha
Bayleton EC 125	rani napad prepelice i hrđa	od busanja – u početku napada prepelice u proljeće	1 l/ha
Tilt 250	bolesti lista stabljike	u vrijeme pojave bolesti	0,5 l/ha

3. TEHNOLOGIJE POTREBNE ZA IZRADU APLIKACIJE

Za izradu ovog završnog rada potrebno je predznanje o razvojnom okruženju Android Studio, predznanje o programskom jeziku Java te predznanje vezano uz ratarsku kulturu. Kao glavni izvor informacija korišten je Internet, predstavnici struke na našem području, te objavljene *online* knjige, članke i dostupnu dokumentaciju vezanu uz razvojno okruženje.

3.1. Android Studio

Android Studio je službeno integrirano razvojno okruženje(IDE) za razvoj Android platforme[10]. Objavljen je 16. svibnja 2013. godine na Google I / O konferenciji te je slobodno dostupan pod licencom Apache License 2.0. Prva faza Android Studio okruženja počela je od inačice 0.1 u svibnju 2013. godine, a zatim je ušla u beta fazu, počevši od inačice 0.8 koja je objavljena u lipnju 2014. godine. Prvi stabilna nadogradnja objavljena je u prosincu 2014. godine, počevši od inačice 1.0. Dostupan je za preuzimanje na Windows, Mac OS X i Linux te je zamijenio Eclipse Android Development Tools (ADT), kao Googleov primarni IDE za domaći Android razvoj aplikacija.

3.1.1. Izgled početnog zaslona

Na slici 3.1. prikazan je početni zaslon razvojnog okruženja Android Studio. Sastoji se on nekoliko osnovnih prozora kao što su:

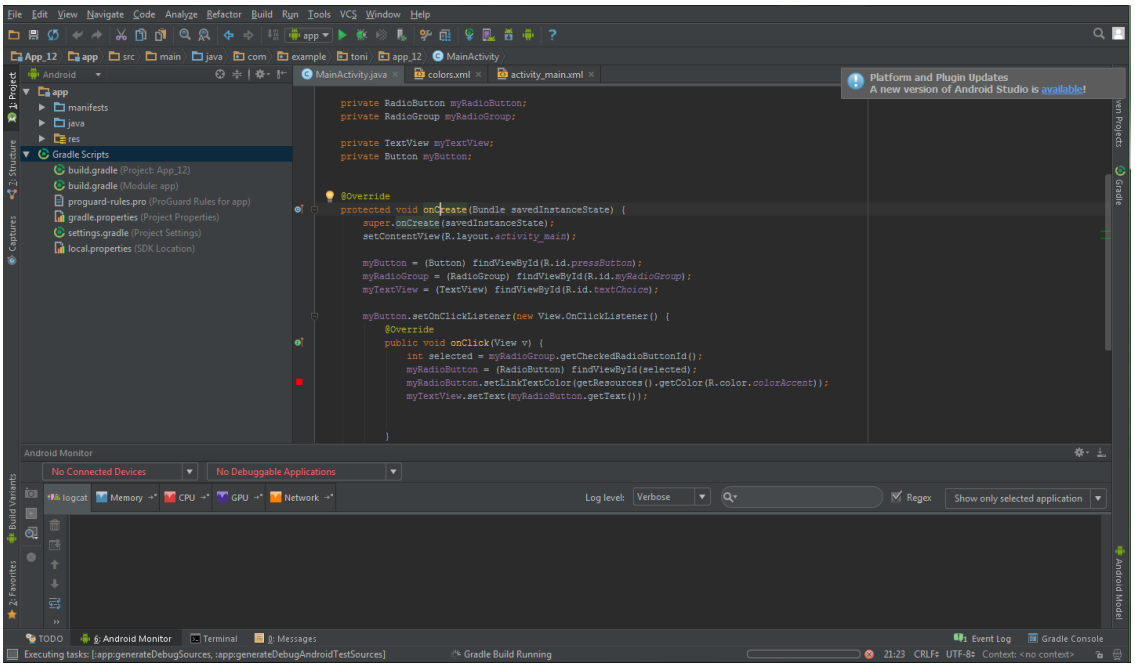
- Alatna traka koja omogućuje obavljanje širokog spektar djelovanja, uključujući i pokretanje aplikacije i pokretanje Android alata.

- Navigacijski traka koji pomaže u pregledu kroz projekt i otvaranju datoteke za uređivanje. On pruža potpuno kompaktan prikaz strukture vidljive u prozoru alata projekta.

- Prozor *editor* gdje se može stvoriti i izmijeniti kod. Ovisno o trenutnoj vrsti datoteke, ovaj prozor se može promijeniti te nudi mogućnost pregleda odgovarajuće XML datoteke.

- Alat za prozore koji daje pristup na određene zadatke kao što su upravljanje projektima, pretraživanje, upravljanje inačicama i još mnogo toga. Također, alati za prozore mogu se proširiti i sažeti.

- Traka statusa koja prikazuje stanje projekta i same IDE, kao i upozorenja ili poruke.

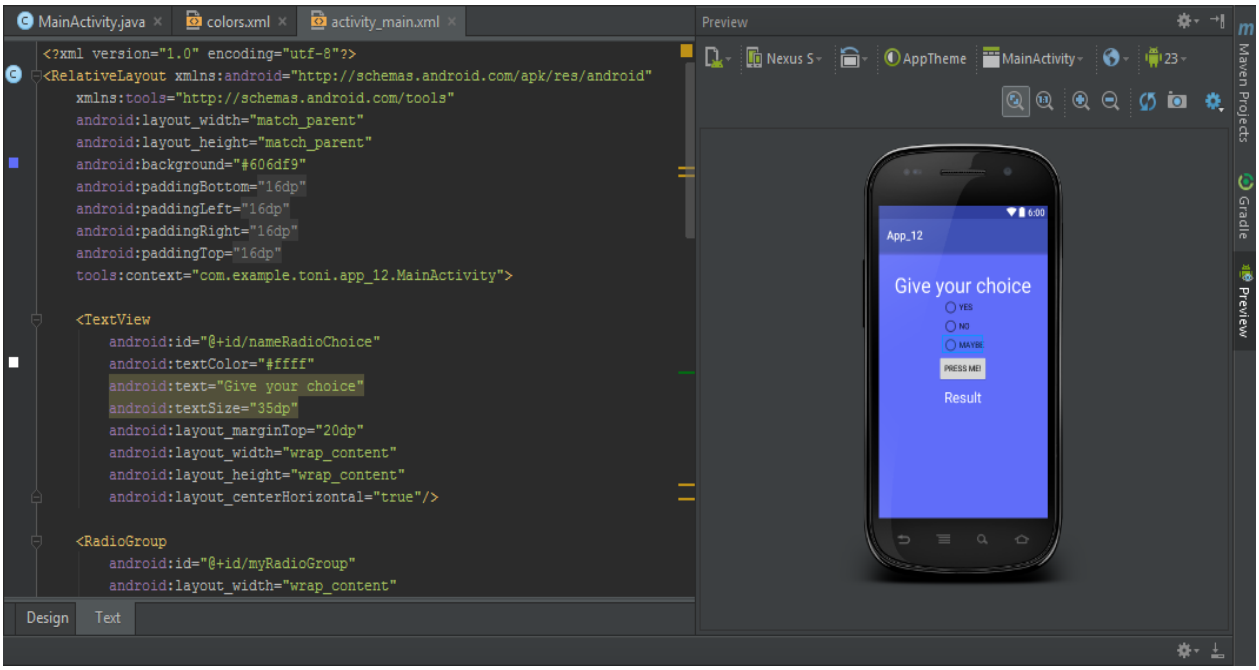


Slika 3.1 Početni zaslon Android Studio

Isto tako, postoji mogućnost i samostalnog organiziranja zaslona ovisno o tome kojim se tipom aplikacije programer bavi te kojim dijelom razvojnog okruženja.

3.1.2. Opisna datoteka (.xml)

XML (EXtensible Markup Language) je, kao što je opisano u knjizi *XML in a Nutshell*[11], opisni jezik za označavanje podataka. Glavna zadaća prilikom stvaranja jezika je bila stvoriti jezik koji će biti jednako razumljiv i računalu i ljudima. Način ostvarenja je vrlo jednostavan; odgovarajući sadržaj treba se uokviriti odgovarajućim oznakama(simbolima) koje ga opisuju i imaju poznato, ili lako shvatljivo značenje. Način označavanja je vrlo sličan HTML jeziku za razvijanje web stranica. Danas je XML jezik vrlo raširen i koristi se za različite namjene kao npr. odvajanje podataka od prezentacije, razmjenu podataka, pohranu podataka, povećavanje dostupnosti podataka i izradu novih specijaliziranih jezika za označavanje. XML spada u skupinu standardiziranih jezika i za njegovu primjenu i nadogradnju brine se World Wide Web Consortium[12].

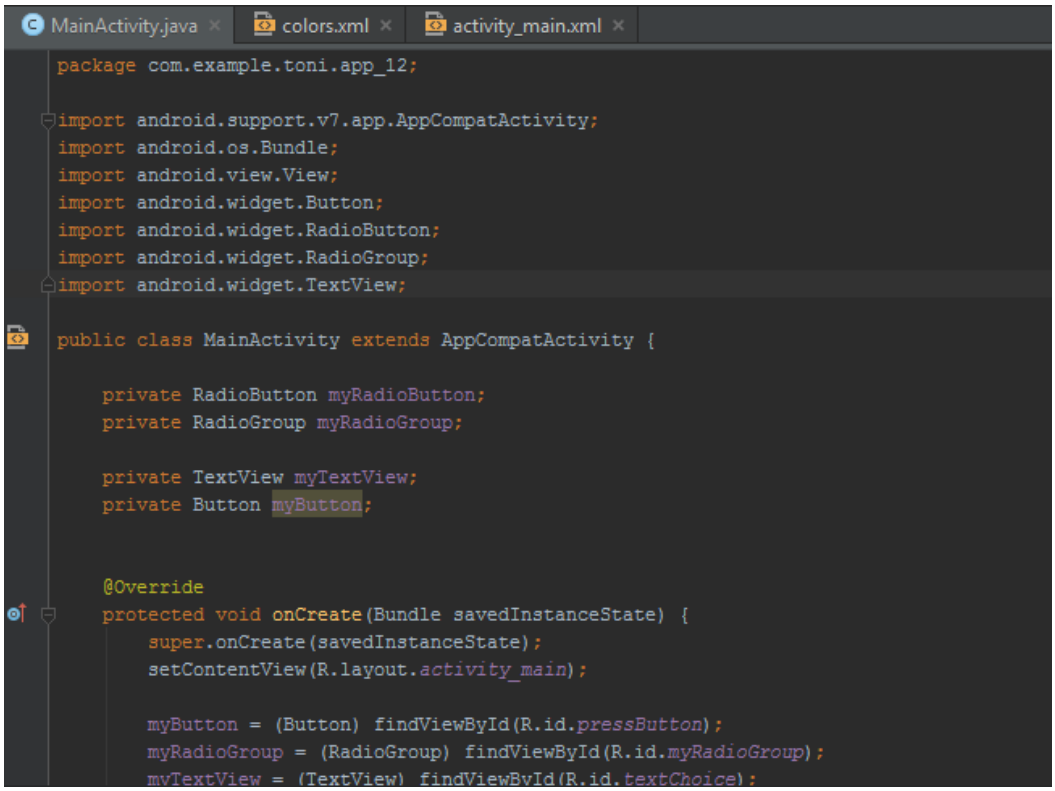


Slika 3.2 Izgled opisne datoteke i njezin prikaz na virtualnom uređaju

U okruženju Android Studio XML jezik se koristi kako bi se opisao položaj pojedine aktivnosti unutar aplikacije, boju pojedinog okvira, tipkala, gumba itd. Isto tako prilikom pisanja opisnog koda u alatnoj traci *Design* može se pratiti promjena izgleda aplikacije. Također, unutar razvojnog okruženja nalaze se razni alati za provjeru sintakse pojedinog jezika kao i alat za automatsko nadopunjavanje dijela riječi ili fraze.

3.1.3. Java datoteka (.java)

Kako bi se povezao opisni jezik(XML) sa potrebnim funkcijama koristi se programski jezik Java. Unutar .java datoteke upisuju se klase i funkcije te kompletan način ponašanja aplikacije. Datoteka uvijek mora sadržavati barem jednu klasu (zadano Main Activity) koja podržava početni zaslon aplikacije. Također, u toj datoteci se učitava pojedina aktivnost iz opisnog koda (npr. *Button*) te se pomoću njega ostvaruje interakcija između korisnika i funkcija. Nadalje, unutar datoteke se mogu učitavati i paketi već gotovih funkcija koja programeru olakšava posao te smanjuje potrebno vrijeme razvijanja aplikacije. Na slici 3.3 može se vidjeti struktura datoteke .java.



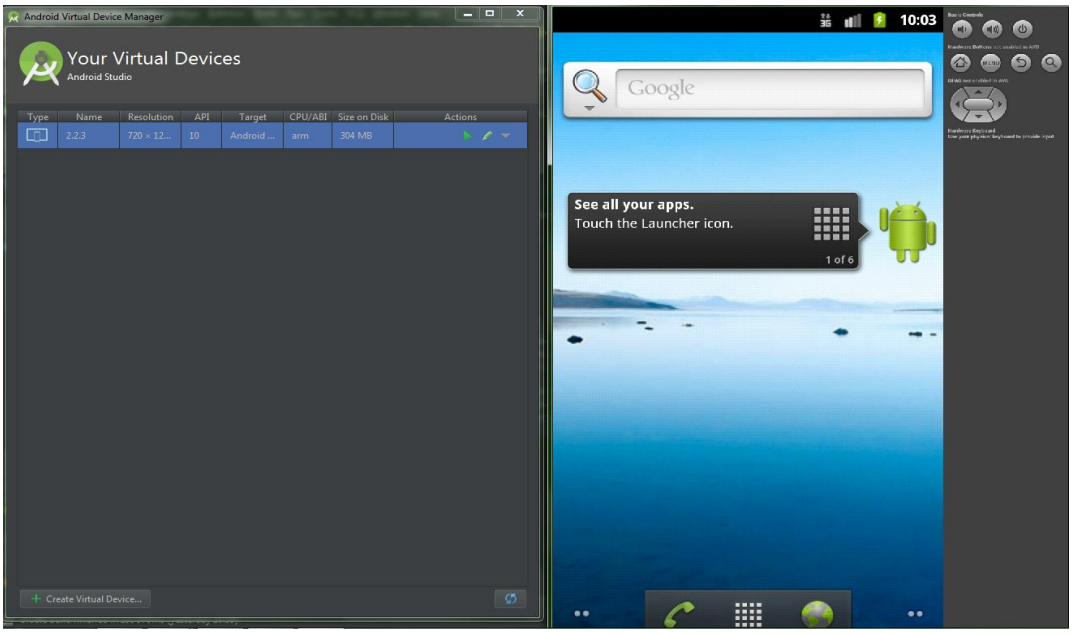
Slika 3.3 Izgled .java datoteke

3.1.4. AVD upravitelj

AVD (hrv. Android virtualni uređaj) je alat koji omogućuje stvaranje virtualnog uređaja. Svrha AVD-a je u tome što podržava testiranje aplikacija virtualno štedeći vrijeme potrebno za testiranje na realnom uređaju. Prilikom testiranja na stvarnom uređaju, potrebno je:

- spremiti aplikaciju u format .apk
- kopirati aplikaciju sa računala na mobilni uređaj i instalirati
- nakon završetka testiranja obrisati aplikaciju

Korištenjem AVD-a potrebno je samo stvoriti virtualni uređaj s odgovarajućom Android inačicom te pokrenuti aplikaciju direktno na računalu. Na slici 3.4 prikazan je izgled AVD upravitelja.

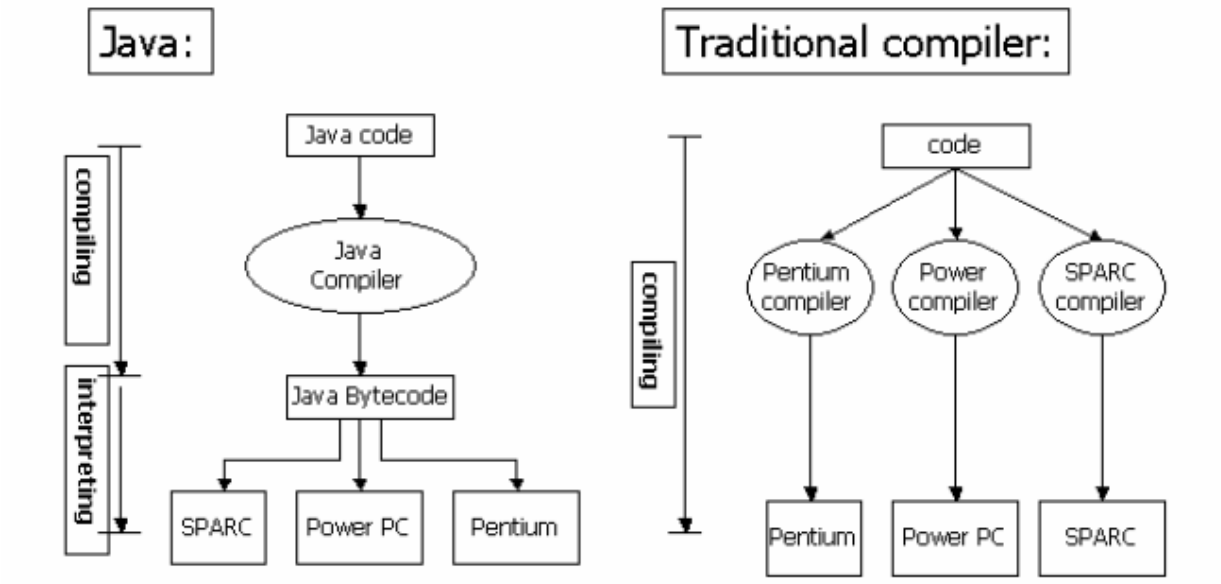


Slika 3.4 Izgled AVD menađera

3.2. Programski jezik Java

Java je objektno orijentirani programski jezik razvijen u timu predvođenim Jamesom Goslingom u kompaniji Sun Microsystems[13] početkom 1990-tih. Ideja je bila da se stvori programski jezik koji bi bio nezavisan od operacijskog sustava, baziran na C++-u, ali s pojednostavljenom sintaksom te stabilnijim *runtime* sustavom. Programski jezik Java jedan je od najpopularnijih jezika među programerima zbog svoje jednostavnosti. Osnovni koncept u Javi je klasa (eng. *class*) unutar kojih je pisan cijeli izvorni kod. Glavna klasa od koje počinje izvođenje programa naziva se *main*. Također, pojedini tipovi podataka, kao što su *boolean, int, float, byte,* predstavljeni su pomoću klasa u odgovarajućem omotu (npr. *java.lang.boolean, java.lang.int, java.lang.float, java.lang.byte*).

Java programski kod prilikom prevođenja ne pretvara se u strojni kod nego je pretvoren u tzv. *bytecode* koji sam po sebi zahtjeva dodatne alate prilikom izvođenja. Okruženje koje pokreće *bytecode* naziva se Java virtualni stroj (*eng. Java Virtual Machine* ili JVM). JVM podržava gotovo sve operacijske sustave počevši od MAC OS do Windowsa i Linuxa. Jednom preveden kod može se pokretati na bilo kojem uređaju koji ima instaliran JVM. Na slici 3.5 prikazan je način izvršavanja Java koda u usporedbi s tradicionalnim prevoditeljem.[14]



Slika 3.5 Usporedba izvršavanja Java programa s tradicionalnim postupkom prevođenja

3.3. Baza podataka(SQL Server)

Prema Internet stranici *Techtarget*[15], SQL (eng. Structured Query Language) razvila je tvrtka IBM ranih 70-ih godina kao standardni jezik za upravljanje bazama podataka. SQL se sastoji od jezika za definiranje podataka, jezika manipulacije podatcima i jezika kontrole podataka. Opseg SQLa uključuje umetanje podataka, upite, ažuriranje i brisanje, kreiranje i izmjenu sheme i kontrolu pristupa podacima. Iako je SQL često opisan u velikoj mjeri kao deklarativni jezik (4GL), on također uključuje i proceduralne elemente. Sintaksa jezika prikazana je na slici 3.6 gdje se vrši izmjena tablice *drzava* tako što se postavlja broj stanovnika po poznanstvu države.

```
UPDATE drzava SET populacija=populacija+1 WHERE naziv=´Hrvatska´;
UPDATE drzava SET populacija=populacija+2 WHERE naziv=´Mađarska´;
UPDATE drzava SET populacija=populacija+3 WHERE naziv=´Slovenija´;
DROP TABLE drzava;
```

Slika 3.6 Primjer sintakse jezika SQL

4. PROGRAMSKI KOD APLIKACIJE

Kod aplikacije podijeljen je u dvije skupine ovisno o funkciji. Opisni programski kod aplikacije obilježava izgled pojedine aktivnosti (zaslona) od početnog do završnog zaslona koji će prikazati rezultate i preporuke. Izvorni programski kod aplikacije rješava povezanost između pojedinih funkcija te blokova aktivnosti. Također, izvorni programski kod definira tijek aplikacije, kojim redom će se izvršavati naredbe i na koji način pozivati određene aktivnosti.

4.1. Opisni programski kod aplikacije (.xml)

Opisni programski kod definira pojedinu aktivnost(zaslon) aplikacije sa gledišta izgleda i smještaja svih alata za interakciju korisnika i aplikacije. Aplikacija ima četiri glavne aktivnosti a to su aktivnost za prijavu korisnika, dodavanja sijanja odnosno žetve, aktivnost prikaza bolesti i gnojiva te aktivnost popisa svih pesticida, herbicida i gnojiva.
Programski kod 4.1 prikazuje opisni kod polja za unos korisničkog imena i lozinke kao dijela prve glavne aktivnosti. Kod sadrži parametre za smještaj *EditText* metode unutar aktivnosti pomoću metode *layout_margin*.

```
<EditText
    android:id="@+id/logIn_etUsername"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginStart="16dp"
    android:background="@drawable/edit_text"
    android:fontFamily="sans-serif"
    android:hint="Korisničko ime"
    android:inputType="textPersonName"
    android:padding="5dp"
    app:layout_constraintBottom_toTopOf="@+id/logIn_etPassword"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/ivLogo"
    app:layout_constraintVertical_chainStyle="packed" />
```

Programski kod 4.1 Opisni kod polja za unos korisničkog imena

Svaki segment određene aktivnosti prilikom definiranja mora imati svoj identifikacijski *String Id* jer upravo taj *String* se koristi kao predmet razlikovanja dvije jednake *EditText* metode.

Uz *EditText* za interakciju na prvom zaslonu koristi se *Button* te *ImageView* koji se smještaju sa istim metodama unutar aktivnosti i također posjeduju svoj *String Id*.

Sljedeća glavna aktivnost je aktivnost za unos podataka o sjetvi odnosno žetvi. Za razliku od prve aktivnosti na njoj se nalazi *RadioButton* koji omogućuje odabir samo jedne mogućnosti, radi li se o sjetvi, žetvi, kukuruzu ili pšenici. Programski kod 4.2 prikazuje opisni kod *RadioButton*-a sa svim potrebnim parametrima za definiranje izgleda i smještaja unutar aktivnosti.

```
<RadioButton
    android:id="@+id/insertCrop_rbCorn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:buttonTint="@color/colorOrange"
    android:text="Kukuruz"
    android:layout_gravity="center"
    app:layout_constraintBottom_toTopOf="@+id/insertCrop_ivSow"
    app:layout_constraintTop_toBottomOf="@+id/insertCrop_ivCorn"
    android:layout_marginBottom="8dp"
    app:layout_constraintLeft_toLeftOf="@+id/insertCrop_ivCorn"
    app:layout_constraintRight_toRightOf="@+id/insertCrop_ivCorn"
    android:checked="true"/>
```

Programski kod 4.2 Opisni kod RadioButton-a za određivanje usjeva

Za definiranje odabira određenog *RadioButtona* koristi se metoda *checked.* Ukoliko se odabere jedan *Button* njegovo stanje prelazi u "*true*" dok kod svih ostalih stanje postaje "*false*". Isto tako na aktivnosti se nalaze i dva *Button*-a koja služe za promjenu aktivnosti nakon što se završi sa unosom parametara i odabirom određenog unosa. Opisni kod za *Button* Natrag nalazi se na programskom kodu 4.3 a kod za *Button* Dodaj na programskom kodu 4.4 . Kod omogućuje smještaj metode *Button* unutar aktivnosti, te opisuje izgled i smještaj unutar aktivnosti. Smještaj postiže se unošenjem parametara relacijske udaljenosti s obzirom na krajnju točku aktivnosti. Nadalje, izgled definiramo učitavanjem već postojećeg izgleda *Button-a* iz mape *drewable.*

```
<Button
    android:id="@+id/insertCrop_btnBack"
    android:layout_width="wrap_content"
    android:layout_height="40dp"
    android:layout_marginLeft="24dp"
    android:background="@color/colorOrange"
    android:drawableLeft="@android:drawable/ic_menu_revert"
    android:paddingRight="5dp"
    android:text="Natrag"
    android:textColor="@color/colorWhite"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/insertCrop_etPurity" />
```

Programski kod 4.3 Opisni kod gumba Natrag

```
<Button
    android:id="@+id/insertData_btnAdd"
    android:layout_width="wrap_content"
    android:layout_height="40dp"
    android:layout_marginRight="24dp"
    android:layout_marginTop="16dp"
    android:background="@color/colorOrange"
    android:drawableRight="@android:drawable/ic_input_add"
    android:text="Dodaj"
    android:textColor="@color/colorWhite"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/insertCrop_etGermination" />
```

Programski kod 4.4 Opisni kod gumba Dodaj

Treća velika aktivnost je aktivnost za unos svih pesticida, herbicida te gnojiva. Po svojem opisu ova aktivnost je složenija od ostalih. Uz *Button*-e i *EditText* koristi se i *Spinner* (padajući izbornik) koji služi za odabir određene bolesti iz baze podataka i odabira određenog pesticida. Programski kod 4.5 prikazuje opisni kod za oba padajuća izbornika sadržana u toj aktivnosti. Taj kod sadrži popis svih inicijala bolesti, te smještaj unutar aktivnosti s obzirom na njezine krajnje točke.

```
<Spinner
    android:id="@+id/illness_spinnerPhase"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toTopOf="@+id/illness_spinnerUseFor"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintLeft_toRightOf="@+id/illness_spinnerUseFor"
    android:layout_marginRight="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp" />

<Spinner
    android:id="@+id/illness_spinnerUnit"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintLeft_toRightOf="@+id/illness_etAmount"
    android:layout_marginRight="16dp"
    app:layout_constraintTop_toTopOf="@+id/illness_etAmount"
    android:layout_marginTop="8dp"
    app:layout_constraintBottom_toBottomOf="@+id/illness_etAmount"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="16dp" />
```

Programski kod 4.5 Opisni kod padajućeg izbornika Spinner

Nadalje, koristi se dvije dodatne aktivnosti za prikaz usjeva te za prikaz popisa korištenih zaštitnih sredstava i gnojiva iz baze podataka. U aktivnosti za prikaz usjeva korišten je tablični prikaz podataka koji je omogućen metodama *TableLayout* i *TableRow* prikazanima u programskom kodu 4.6.

```
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_toRightOf="@id/listItem_ivImage"
    android:layout_below="@id/listItem_tvSort"
    android:id="@+id/listItem_tlDetails"
    android:stretchColumns="0,1"
    android:gravity="center"
    android:layout_marginEnd="8dp"
    >

    <TableRow>

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_column="0"
            android:layout_margin="2dp"
            android:gravity="center"
            android:text="Težina [kg]"
            android:textColor="@color/colorOrange" />
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="Vlažnost [%]"
            android:gravity="center"
            android:textColor="@color/colorOrange"
            android:layout_margin="2dp"
            android:layout_column="1"
```

Programski kod 4.6 Opisni kod tablice za prikaz usjeva

Svaki redak tablice ima u sebi dodatne TextView i EditText za prikaz i unos teksta u memoriju aplikacije. Upravo pomoću tih tekstualnih alata povezuju se korisnik i aplikacija.

Zaključna veća aktivnost aplikacije je aktivnost prikaza liste unesenih zaštitnih sredstava. Navedena lista je posebna po tome što uz prikaz teksta i slike određenog zaštitnog sredstva ima mogućnost otvaranja nove aktivnost pritiskom na člana liste kao na gumb. Na programskim kodovima 4.7 i 4.8 prikazani su opisni kod liste i njezini parametri inicijalizacije te opis teksta i slike relativnih rasporeda.

```
    <include layout="@layout/activity_crop_details_content"/>

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fertilizer_fabAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|right"
        android:clickable="true"
        app:backgroundTint="@color/colorOrange"
        app:fabSize="mini"
        android:layout_margin="8dp"
        app:srcCompat="@mipmap/add" />
```

Programski kod 4.7 Opisni kod liste zaštitnih sredstava

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/listview_item">


    <ImageView
        android:id="@+id/illnessItem_ivImage"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:scaleType="centerCrop"
        android:layout_margin="8dp"
        android:src="@mipmap/illness"
        />

    <TextView
        android:id="@+id/illnessItem_tvName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="8dp"
        android:layout_toRightOf="@id/illnessItem_ivImage"
        android:text="Ime"
        android:textColor="@color/colorOrange"
        android:textSize="18sp"
        android:textStyle="bold"
        />
```

Programski kod 4.8 Opisni kod relativnog rasporeda liste

4.2. Izvorni programski kod aplikacije (.java)

Prilikom pokretanja aplikacije potrebno se prijaviti u bazu podataka aplikacije. Također, prilikom prvog korištenja aplikacije potrebno je izraditi račun. Registracija je omogućena preopterećenim tekstom *TextView* na kojega je dodana mogućnost klika i otvaranja nove aktivnosti za registraciju. Programski kod 4.9 prikazuje kod prve aktivnosti aplikacije. Poziva se funkcija *onCreate()* koja se izvršava prilikom pokretanja aplikacije. Prijava je ostvarena pomoću funkcija *EditText* i *Button*. Programski kod prikazuje učitavanje teksta i njegovo korištenje u prijavi korisnika u sustav baze podataka. Nakon unosa korisničkih podataka i lozinke prilikom pritiska gumba pokreće funkciju *login()*. Ukoliko je prijava uspješna aplikacija otvara novu aktivnost koja je ujedno i glavna aktivnost aplikacije, suprotno izbacuje tekst neuspješne prijave i vraća aktivnost na početak.

```
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    this.etUsername = (EditText) findViewById(R.id.logIn_etUsername);
    this.etPassword = (EditText) findViewById(R.id.logIn_etPassword);
    this.btnLogIn = (Button) findViewById(R.id.logIn_btnLogIn);
    this.btnLogIn.setOnClickListener((v)  › {
            if(checkFields()) {
                login();
            }
            else {
                Toast.makeText(getApplicationContext(), EMPTY_FIELD_TEXT, Toast.LENGTH_SHORT).show();
            }
    });
    this.tvRegister = (TextView) findViewById(R.id.register_tvLogin);
    this.tvRegister.setOnClickListener((v)  › {
            Intent registerIntent = new Intent(LoginActivity.this, RegisterActivity.class);
            startActivity(registerIntent);
```

Programski kod 4.9 Izvorni kod početnog zaslona aplikacije

```
private boolean checkFields() {
    if(etPassword.getText().toString().equals("")||etUsername.getText().toString().equals("")){
        return false;
    }
    else {
        return true;
    }
}

private void login() {
    String username = etUsername.getText().toString();
    String password = etPassword.getText().toString();
    String type = "login";
    BackgroundWorker backgroundWorker = new BackgroundWorker(LoginActivity.this);
    backgroundWorker.execute(type, username, password);
```

Programski kod 4.10 Izvorni kod funkcije login()

Nakon prijave korisnika, aplikacija počinje sa dohvaćanjem podataka iz udaljenje baze podataka, stoga je potrebno imati potpunu mrežnu dostupnost. Definira se klasa *CropDataActivity* koja će u svojim parametrima prikupljati sve podatke koje pronađe u bazi podataka. Programski kod klase prikazan je na programskom kodu 4.11.

```
public class CropDataActivity extends AppCompatActivity {

    RecyclerView rvCrops;
    String mUsername;
    TextView tvUsername, tvSignOut;
    CropAdapter mCropAdapter;
    RecyclerView.LayoutManager mLayoutManager;
    RecyclerView.ItemDecoration mItemDecoration;
    ArrayList<Crop> crops;
    FloatingActionButton fabAdd;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_crop_data);


        mUsername = getIntent().getExtras().getString("username");
        setUpUI();

        new GetCropData(CropDataActivity.this, mUsername).execute();

    }
```

Programski kod 4.11 Izvorni kod klase za prikupljanje podataka usjeva

Taj kod sadrži parametre kao što su *mUsername* za definiciju imena sorte, *tvUsername* za prikaz sorte usjeva te listu usjeva *crops*. Kada je klasa kreirana počinje proces prikupljanja podataka. Svaki korisnik ima svoje korisničko ime te ne može doći do preklapanja podataka. Funkcija koja prikuplja podatke *doInBackground* prikazana je programskim kodom 4.12.

```
protected String doInBackground(Void... params) {

    try {
        URL url = new URL(json_url);
        HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
        httpURLConnection.setRequestMethod("POST");
        httpURLConnection.setDoInput(true);
        httpURLConnection.setDoOutput(true);
        OutputStream outputStream = httpURLConnection.getOutputStream();
        BufferedWriter bufferedWriter = new BufferedWriter(new OutputStreamWriter(outputStream, "UTF-8"));
        String get_username = URLEncoder.encode("username", "UTF-8") + "=" + URLEncoder.encode(mUsername, "UTF-8");
        bufferedWriter.write(get_username);
        bufferedWriter.flush();
        bufferedWriter.close();
        outputStream.close();

        InputStream inputStream = httpURLConnection.getInputStream();
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));
        StringBuilder stringBuilder = new StringBuilder();
        while ((JSON_STRING = bufferedReader.readLine()) != null) {
            stringBuilder.append(JSON_STRING + "\n");
        }
```

Programski kod 4.12 Izvorni kod spajanja aplikacije na URL

Funkcija *doInBackground* temelji se na poznavanju URL adrese baze podataka aplikacije. Zatim uzima ime korisnika i traži ga u bazi podataka te ukoliko ga pronađe, počinje čitati bazu podataka red po red pomoću *BufferedReader* i sprema ga kao tip podatka *string* operatorom *StringBuilder*. Nakon pregledavanja i učitavanja baze podataka korisnik može dodavati nove podatke ovisno o tome želi li dodati sjetvu ili žetvu. Svaki element aktivnosti , kao što su *Button* , *RadioButton, EditText,* mora se povezati u programski kod aplikacije. Programski kod 4.13 prikazuje povezivanje metode za prikupljanje podataka o sjetvi odnosno žetvi. Ideja povezivanja proteže se i u daljnjem tijeku aplikacije prilikom dodavanja zaštita od bolesti, unos pesticida te gnojidbi.

```
private void setUpUI() {
    this.etGermination = (EditText) findViewById(R.id.insertCrop_etGermination);
    this.etPurity = (EditText) findViewById(R.id.insertCrop_etPurity);
    this.etHumidity = (EditText) findViewById(R.id.insertCrop_etHumidity);
    this.etSort = (EditText) findViewById(R.id.insertCrop_etSort);
    this.etWeight = (EditText) findViewById(R.id.insertCrop_etWeight);
    this.rbCorn = (RadioButton) findViewById(R.id.insertCrop_rbCorn);
    this.rbCorn.setOnCheckedChangeListener(this);
    this.rbHarvest= (RadioButton) findViewById(R.id.insertCrop_rbHarvest);
    this.rbHarvest.setOnCheckedChangeListener(this);
    this.rbSow= (RadioButton) findViewById(R.id.insertCrop_rbSow);
    this.rbSow.setOnCheckedChangeListener(this);
    this.rbWheat= (RadioButton) findViewById(R.id.insertCrop_rbWheat);
    this.rbWheat.setOnCheckedChangeListener(this);
    this.btnBack = (Button) findViewById(R.id.insertCrop_btnBack);
    this.btnBack.setOnClickListener((v)  -> {
            Intent CropDataIntent = new Intent(InsertCropDataActivity.this,CropDataActivity.class);
            CropDataIntent.putExtra("username", username);
            startActivity(CropDataIntent);
    });
    this.btnAdd = (Button) findViewById(R.id.insertData_btnAdd);
    this.btnAdd.setOnClickListener((v)  -> {
            Crop crop = getCropFromForm();
            if(crop!=null){
                new InsertCropData(InsertCropDataActivity.this,username).execute(crop);
            }
    });
```

Programski kod 4.13 Izvorni kod prikupljanja podataka o žetvi odnosno sjetvi

Za odabir određenog unosa koristi se uvjetna metoda *switch-case* . Metoda upravlja radom *RadioButton* koji ima ulogu omogućiti korisniku odabira samo jednog slučaja , radi li se o sjetvi ili žetvi te radi li se o kukuruzu ili pšenici. Programski kod 4.14. omogućuje odabir usjeva i sadrži *boolean* tip podatka koji prati koje je polje označeno, te sva ostala polja isključuje iz odabira.

```
public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
    switch(buttonView.getId()){
        case R.id.insertCrop_rbCorn:
            if(isChecked) rbWheat.setChecked(false);
            else rbWheat.setChecked(true);
            break;
        case  R.id.insertCrop_rbWheat:
            if(isChecked) rbCorn.setChecked(false);
            else rbCorn.setChecked(true);
            break;
        case R.id.insertCrop_rbHarvest:
            if(isChecked) rbSow.setChecked(false);
            else rbSow.setChecked(true);
            break;
        case R.id.insertCrop_rbSow:
            if(isChecked) rbHarvest.setChecked(false);
            else rbHarvest.setChecked(true);
    }
}
```

Programski kod 4.14 Izvorni kod switch-case petlje za označavanje oblika usjeva

Drugi dio aplikacije odnosi se na zaštitu i prevenciju od bolesti te gnojidbu usjeva. Kada korisnik odabere svoj usjev, usjevu može dodati sve što je koristio prilikom prihrane i zaštite određenog usjeva. Klasa *Illness* sadrži parametre i konstruktore za unos zaštite od bolesti te je prikazana na programskom kodu 4.15 . Programsko rješenje korišteno za zaštitu od bolesti podržava i zaštitu od štetočina te korova. U svakoj zaštiti postoje parametri koji prikupljaju opis zaštite, datum primjene koji se automatski generira preopterećenom metodom *parseDate*.

```
public class Illness {
    private String name, use_for, phase, unit, mDate, amountData;
    private float amount;

    public String getAmountData() { return amountData; }

    public void setAmountData(String amountData) { this.amountData = amountData; }

    public Illness(String name, String use_for, String phase, double amount, String unit, String mDate, String amountData) {
        if(mDate!=null) this.mDate = parseDate(mDate);
        else  this.mDate=null;
        this.name = name;
        this.use_for = use_for;
        this.phase = phase;
        this.amount = (float) amount;
        this.unit=unit;
        this.amountData=amountData;
    }
```

Programski kod 4.15 Izvorni kod klase za zaštitu od bolesti

Nakon zaštite slijedi programsko rješenje za unos gnojiva koje sadrži dodatne parametre prikazane na programskom kodu 4.16 . Uneseni tip podatka u bazu je *String* stoga se koristi funkcija *float.valueOf ()* kako bi tekst pretvorila u realan broj s kojim se računa postotak korištenja pojedinog gnojiva.

```
public Fertilizer(String mDate, String mNPKWeight, String mNPKratio, String mExtra, String mComment, String mTime) {

    if(mDate!=null) this.mDate = parseDate(mDate);
    else  this.mDate=null;
    this.mNPKWeight = mNPKWeight;
    String[] weight=mNPKWeight.split(" ");
    this.mWeight = Float.valueOf(weight[0]);
    this.mNPKratio = mNPKratio;
    this.mExtra = mExtra;
    this.mComment = mComment;
    this.mTime = mTime;
    String str = mNPKratio.replaceAll(" ","");
    String[] NPK = str.split(":");
    mN=mWeight*(Float.valueOf(NPK[0])/100);
    mP=mWeight*(Float.valueOf(NPK[1])/100);
    mK=mWeight*(Float.valueOf(NPK[2])/100);
}
```

Programski kod 4.16 Izvorni kod pretvaranja unesenog podatka u realan broj

Kako korisnik ne bi morao unositi posebno ime svake bolesti, pesticida, korova i herbicida aplikacija nudi popis svim sredstava iz baze podataka. To je omogućeno tipom podatka *ArrayList* koji sprema sva zaštitna sredstva u polje *String*-ova. Programsko rješenje , u ovom slučaju pesticida, prikazano je na programskom kodu 4.17 . Izlazak iz aplikacije omogućeno je metodom *Button Odjava* . Nakon odjave sve promjene se spremaju u udaljenu bazu kako bi bile spremne za buduće potrebe.

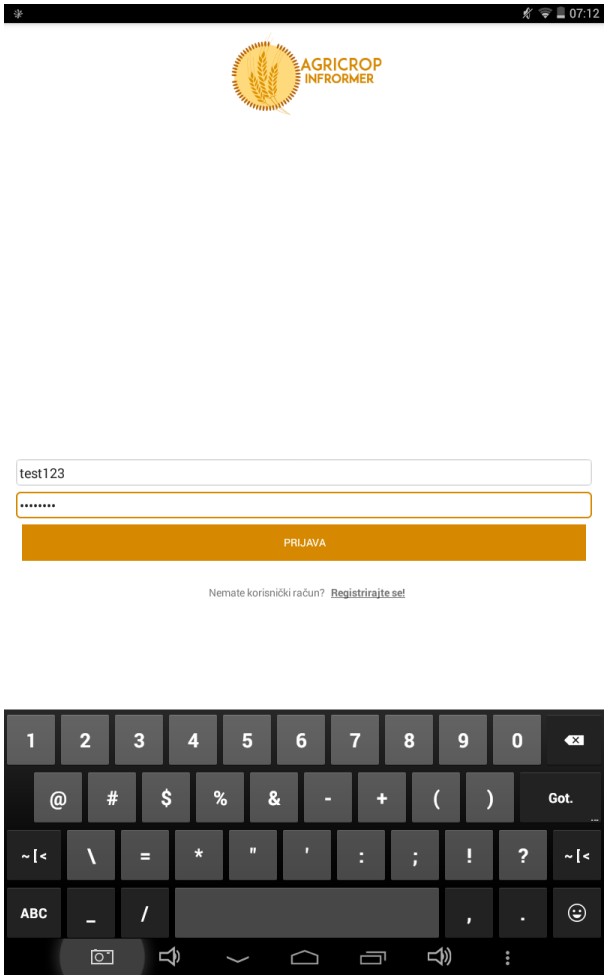
```
ArrayList<String> pestForSpinner = new ArrayList<>();
pestForSpinner.add("");
pestForSpinner.addAll(pests);
ArrayAdapter<String> dataAdapter = new ArrayAdapter<~>(this,
        android.R.layout.simple_spinner_item, pestForSpinner);
dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
this.spinnerPest.setAdapter(dataAdapter);
this.spinnerPest.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        if (!spinnerPest.getSelectedItem().toString().equals("")) {
            ArrayList<CropDetailListViewRow> rows = new ArrayList<~>();
            for (int i = 0; i < firstPesticideDetails.size(); i++) {
                if (firstPesticideDetails.get(i).getmPhase().equals(spinnerPest.getSelectedItem().toString())) {
                    rows.add(firstPesticideDetails.get(i));
                }
            }
            setListView(rows);
        }
        else setListView(firstPesticideDetails);
    }
```

Programski kod 4.17 Izvorni kod polja Stringova za spremanje zaštite

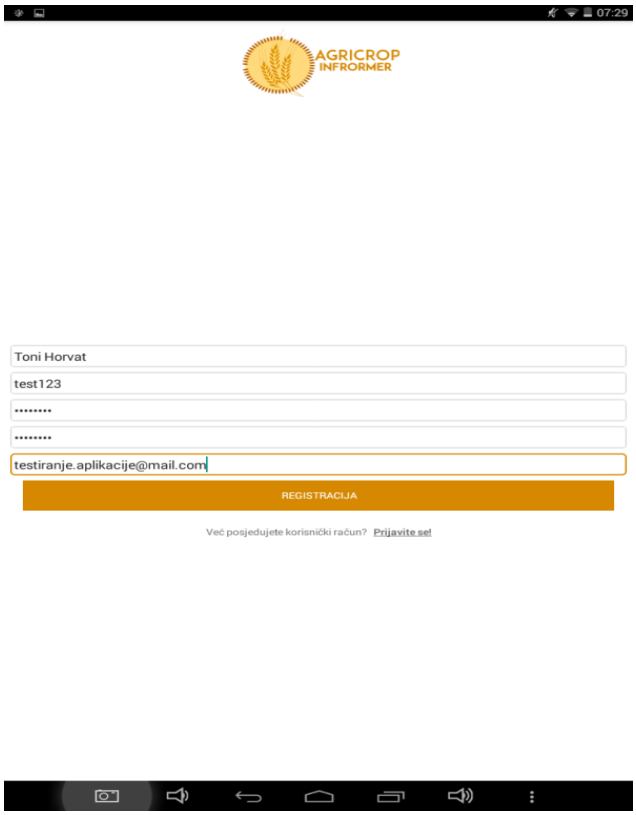
5. NAČIN KORIŠTENJA , TESTIRANJE I ANALIZA RADA APLIKACIJE

5.1. Korištenje aplikacije

Prilikom testiranja aplikacije korišten je osobni tablet VIVAX model NO:TPC-800 sa dvostrukim operacijskim sustavom (Android i Windows). Nakon instaliranja aplikacije pojavljuje se početni zaslon za prijavu prikazano na slici 5.1 . Ukoliko korisnik ne posjeduje račun, ispod gumba za prijavu nalazi se gumb za registraciju koji otvara novu aktivnost za registriranje novih korisnika prikazano na slici 5.2 .



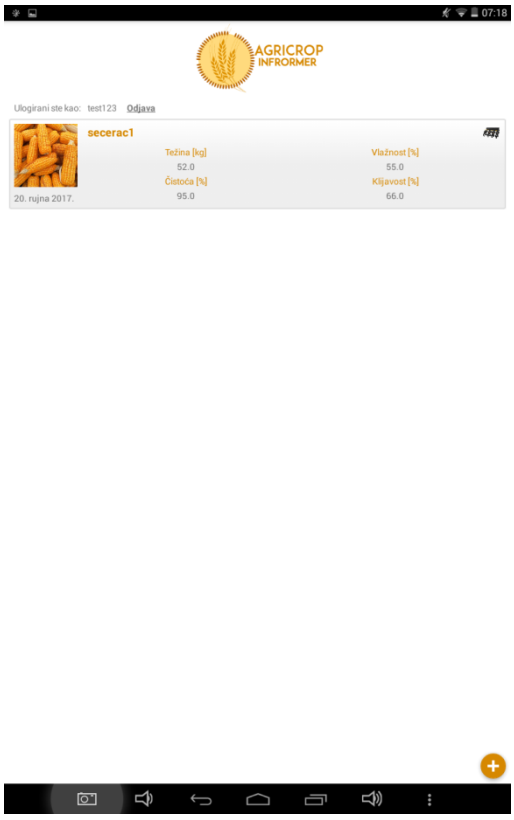
Slika 5.1 Početni zaslon aplikacije, prijava korisnika



Slika 5.2 Zaslon za registraciju korisnika

Nakon što se korisnik prijavio u sustav, otvara se početno sučelje u kojem aplikacija povlači već postojeće podatke iz baze podataka korisnika. Na tom zaslonu nalazi se gumb "+" koji otvara novi zaslon za dodavanje sijanja odnosno žetve što se vidi na slici 5.3 . Nadalje, kada korisnik ispuni potrebne podatke, klikom na gumb *Dodaj* aktivira se procedura za upis u bazu podataka. Kada je usjev spremljen u bazu podataka postaje vidljiv na početnom zaslonu aplikacije te se automatski generira datum dodavanja koji se uzima iz trenutnog vremena poslužitelja baze podataka. Važno je napomenuti da na ovom zaslonu može se upravljati samo usjevima koji se siju, zato što je žetva kraj ciklusa rasta usjeva te nema potrebe dodavati nikakva sredstva za zaštitu ili gnojiva. Izgled dodanog usjeva *secerac1* prikazan je na slici 5.4 sa svim parametrima baze podataka korisnika. Ti parametri su količina, vlažnost, čistoća i klijavost.

Slika 5.3 Zaslon za dodavanje usjeva

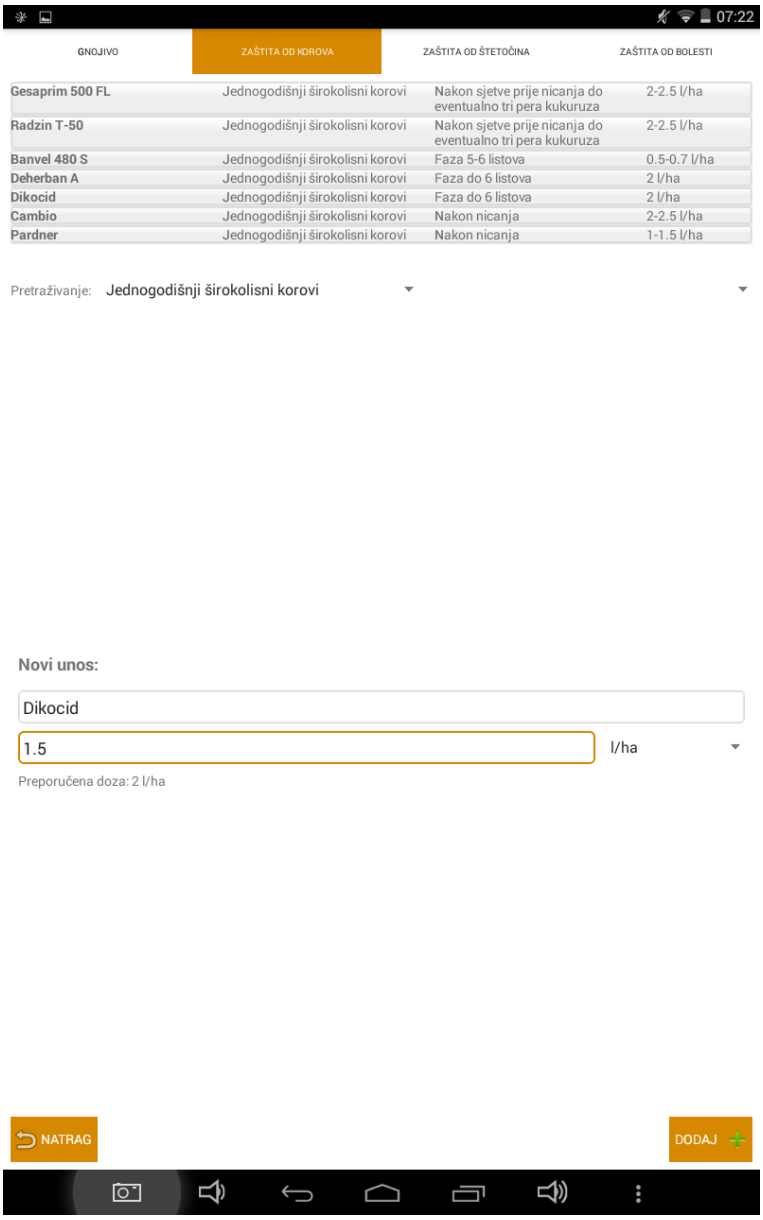


Slika 5.4 Zaslon za pregled unesenih sijanja i žetvi

Prilikom pritiska na željeni usjev otvara se novi zaslon na kojem se nalazi popis svih aktivnosti koje su rađene na tom usjevu, od gnojidbe do zaštite od korova. Pritiskom na gumb "+" otvara se novi zaslon na kojem se može odabrati željena akcija koja će se dodati u bazu podataka. Na slici 5.5 prikazano je dodavanje gnojidbe usjeva, gdje se unose količine i omjer umjetnog gnojiva, te mogućnost dodavanja drugih gnojiva kao što je stajsko i biljno.

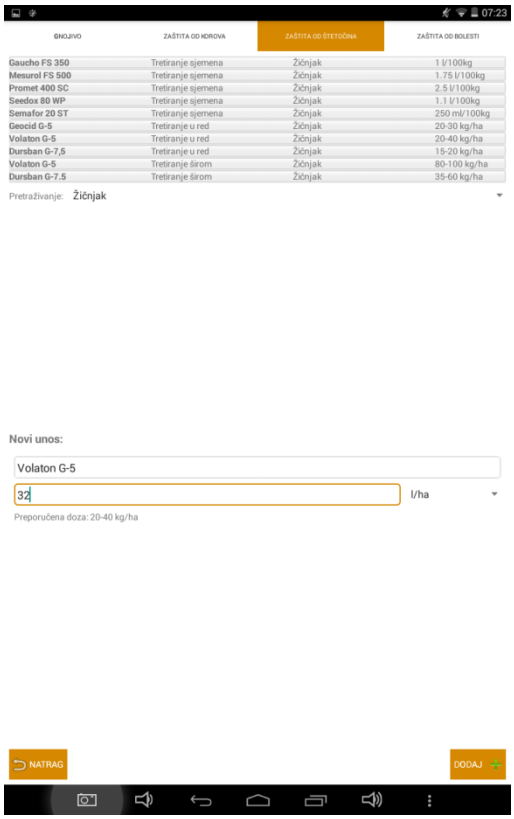
Slika 5.5 Zaslon za dodavanje gnojiva

Klikom gumba *Dodaj* aktivnost se sprema u bazu podataka trajno. Zatim slijedi gumb *Zaštita od korova* koji otvara sljedeći prozor zaslona vidljiv na slici 5.6 . Na vrhu zaslona nalazi se lista trenutno najraširenijih sredstava za zaštitu od korova. Ispod liste nalazi se mjesto za pretraživanje liste ovisno o parametru imena korova. Aplikacija također daje preporuke za sa svako određeno tretiranje kako bi pomogla korisniku povećati prinos njegovog usjeva.

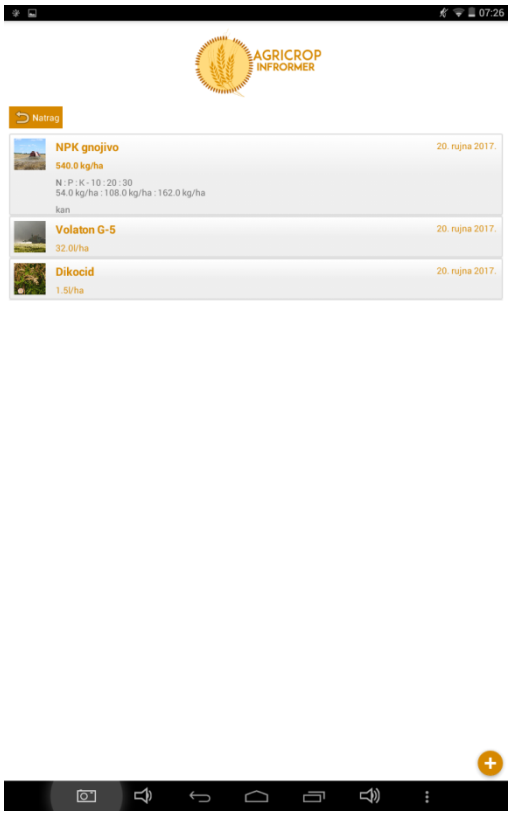


Slika 5.6 Zaslon za dodavanje zaštite od korova

Kao što je slučaj kod dodavanja aktivnosti kod zaštite od korova tako postoje i aktivnosti za dodavanje zaštite od štetočina te dodavanje zaštite od bolesti prikazano na slici 5.7 . Kada je korisnik dodao sve aktivnosti koje je planirao aplikacija ih sprema na udaljenu bazu podataka kako bi ih mogao koristiti svaki puta kada se prijavi. Završni izgled njegovog usjeva vidljiv je na slici 5.8 koja prikazuje sve dodane zaštite i gnojiva. nedostatak aplikacije je što nema mogućnost mijenjanja sadržaja pojedine aktivnosti, stoga valja biti oprezan prilikom dodavanja sadržaja u suprotnom će se on morati obrisati i ponovo dodavati s točnim parametrima.



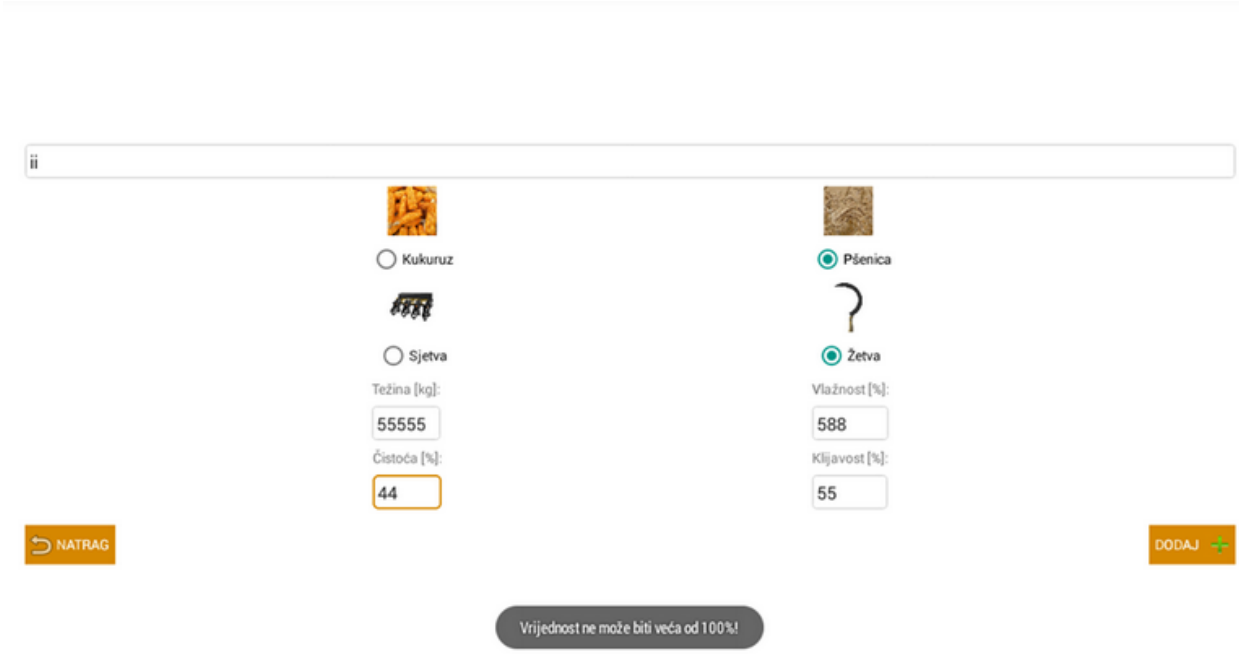
Slika 5.7 Zaslon za unos zaštite od štetočina



Slika 5.8 Završni zaslon usjeva sa popisom svih aktivnosti odrađenih na njemu

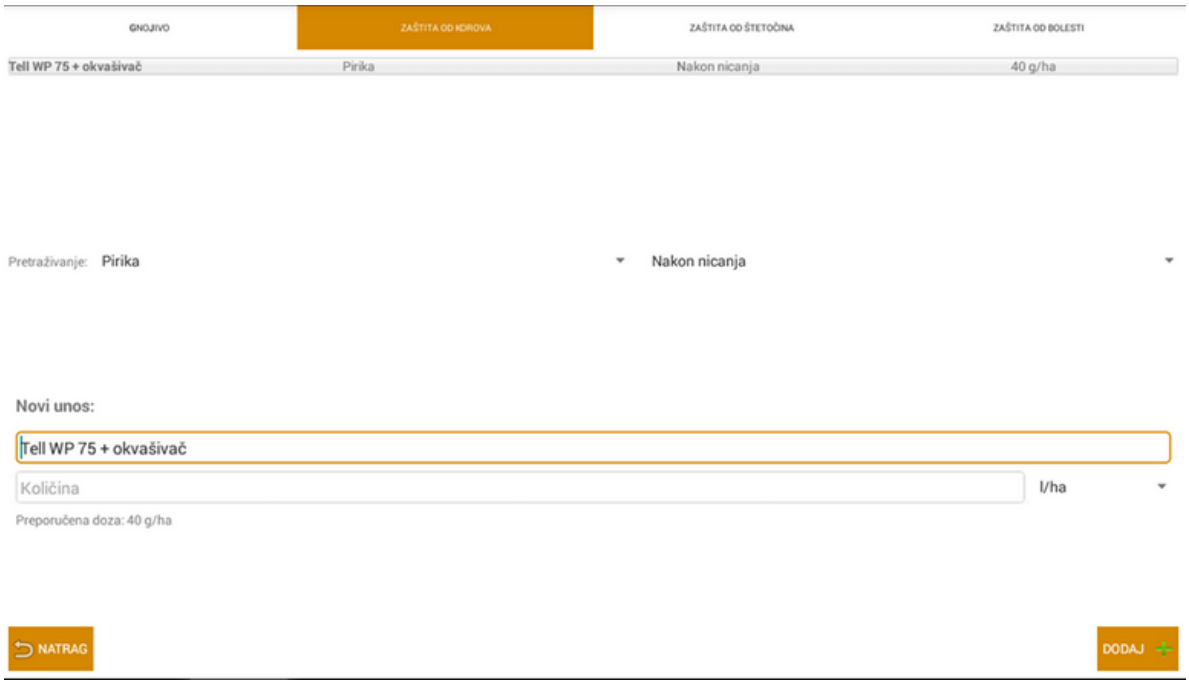
5.2. Testiranje i analiza rada aplikacije

Prilikom testiranja aplikacije biti će provjerena cijela funkcionalnost aplikacije, od dodavanja usjeva u bazu podataka do dodavanja zaštitnih sredstava i program preporuke. Za dodavanje usjeva koriste se parametri vlažnosti, klijavosti, čistoće i težine, kojima se definira opis usjeva. Kako bi se ispitala ispravnost za parametar vlažnosti zrna biti će unesen broj veći od 100% vidljivo na slici 5.9.

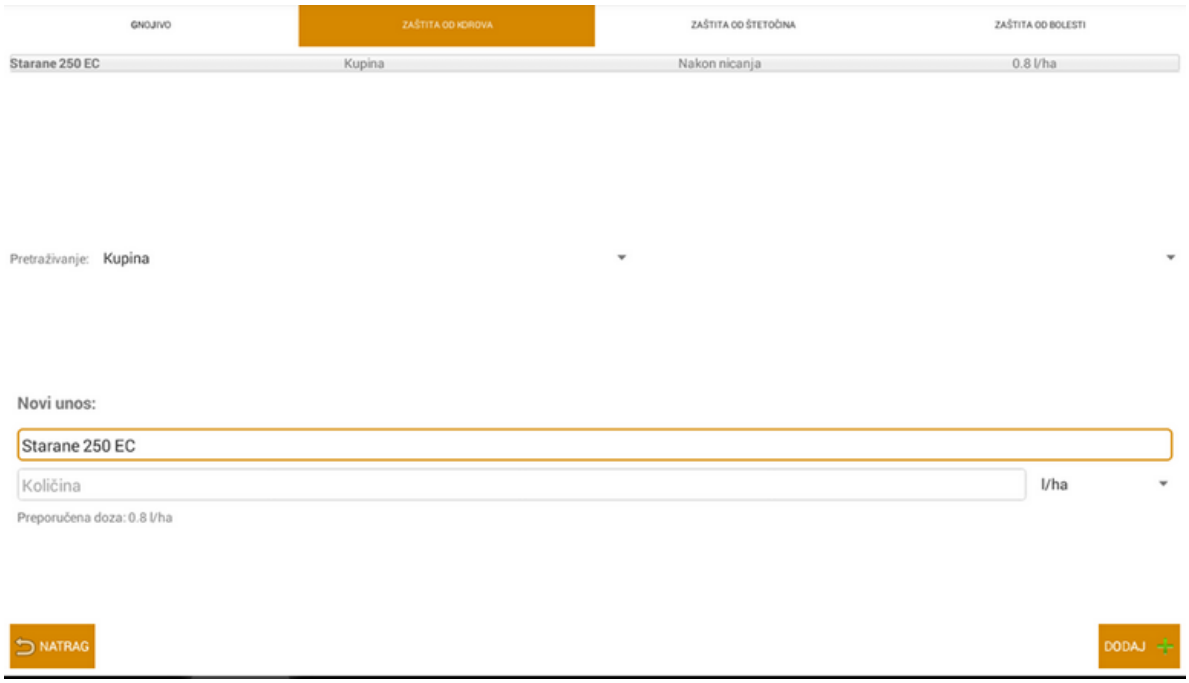


Slika 5.9 Zaslon testiranja unosa usjeva u bazu podataka

Aplikacija javlja grešku sa podsjetnikom kako vrijednost vlažnosti ne može biti veća od 100%. Tek nakon što se broj ispravi unutar dozvoljenih granica usjev se sprema u bazu podataka čime je aplikacija testom potvrdila da uspješno radi u segmentu dodavanja usjeva u bazu podataka. Nadalje, kada je usjev učitan u bazu podataka može se testirati funkcionalnost zaštite od bolesti, korova, nametnika i gnojidbe. Prilikom testiranja pojedinog segmenta biti će uzet jedan parametar iz baze podataka prikazano na slici 5.10 te slici 5.11.



Slika 5.10 Zaslon testiranja zaštite od pirike

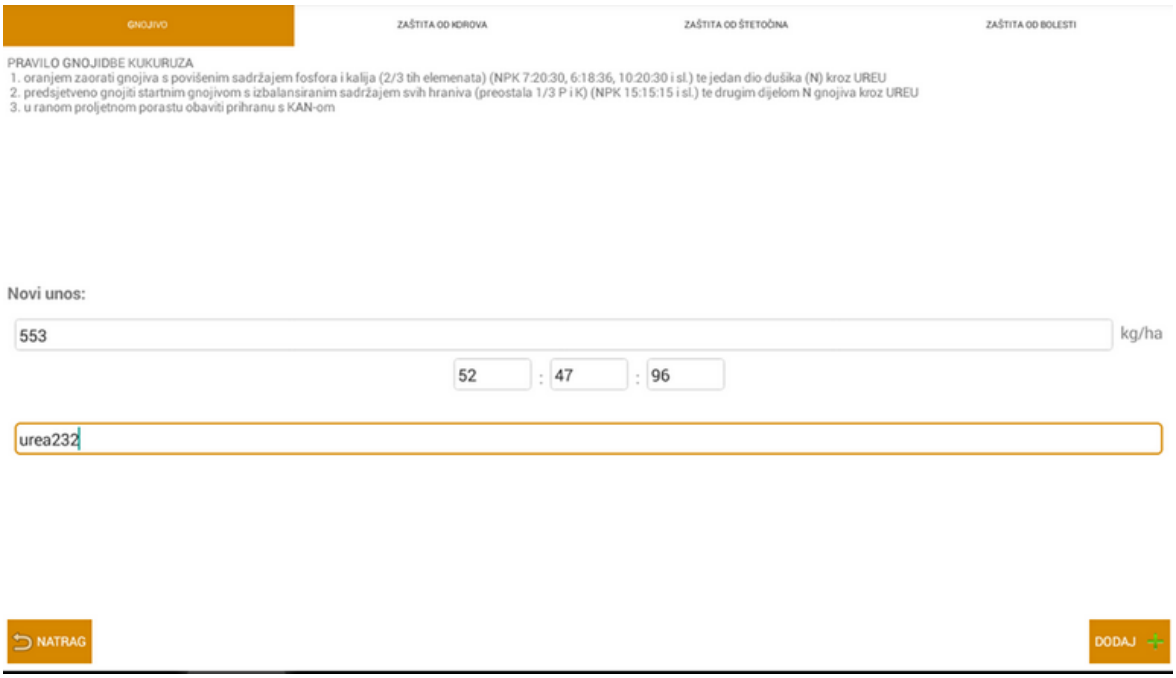


Slika 5.11 Zaslon testiranja zaštite od kupine

Za određeni unos parametara, kao što su pirika i kupina, aplikacija nudi točno sredstvo u cilju zaštite te preporuku u kojoj mjeri bi se trebalo koristiti u cilju najveće učinkovitosti. Dobiveni rezultati dokazuju uspješan rad aplikacije u području zaštite od korova. Također, u segmentu zaštite od nametnika kao parametar testiranja bit će nametnik moljac vidljivo na slici 5.12.

Slika 5.12 Zaslon testiranja zaštite od nametnika moljca

Rezultat testiranja potvrđuje ispravnost rada aplikacije u dijelu zaštite od nametnika, te vidljivo je kako aplikacija predlaže sredstva u suzbijanju nametnika moljca i za pojedino sredstvo predlaže optimalnu dozu korištenja. Na kraju slijedi testiranje programa preporuke prilikom gnojidbe usjeva. Testiranje aplikacije na unos gnojiva prikazano je na slici 5.13 .



Slika 5.13 Zaslon testiranja unosa gnojiva

Iz rezultata testiranja programa preporuke prilikom gnojidbe usjeva aplikacija pruža pravila po kojima bi korisnik trebao koristiti umjetna gnojiva te u kojem optimalnom omjeru. Cjelokupno testiranje i analiza aplikacije potvrđuje kako aplikacija radi dobro u svim njezinim segmentima te njezino korištenje pomaže poljoprivredniku u njegovanju i praćenju njegovog usjeva.

6. ZAKLJUČAK

Prije izrade aplikacije bilo je potrebno prikupiti sva potrebna predznanja o Android programiranju te poljoprivrednoj industriji vezanoj uz kukuruz i pšenicu. Namjena aplikacije bila je pomoći svakodnevnom čovjeku, poljoprivredniku u uspješnom i učinkovitijem njegovanju usjeva putem gnojidbe i zaštita. Ovom aplikacijom upravo je to i postignuto, gdje je testiranjem aplikacije utvrđeno kako aplikacija radi u svakom njezinom segmentu bez ikakvih problema. Prilikom unosa podataka sjetve ili žetve, korisnik ih trajno sprema u udaljenu bazu podataka koju može dohvatiti svakom prijavom u njezin sustav. Nadalje, buduća svrha aplikacije nakon korištenja kroz nekoliko godina bila bi izrada modela po kojem bi se analizirali svi prikupljeni podaci od svih korisnika i napravilo optimalno rješenje za sijanje određene kulture i njegovu zaštitu. Također, aplikacijom je ostvareno davanje savjeta korisniku prilikom odabira određenog sredstva u zaštiti i suzbijanju od nametnika, bolesti, korova te gnojidbi tijekom rasta biljke.

Velika zahvala društvu agronoma Općine Čepin na nesebičnoj pomoći oko prikupljanja podataka vezanih uz ratarsku kulturu te kolegi Kristijanu Štefančiću oko pomoći pri uklanjanju grešaka u aplikaciji i pomoći pri realizaciji koda spajanja aplikacije na udaljenu bazu podataka koje su bile izvan osobnih okvira znanja programiranja.

LITERATURA

[1] Agromedia http://www.agromedia.rs/blog/agrokutak/5-najboljih-aplikacija-za-modernog-poljoprivrednika

[2] FarmingCalulatorPro https://play.google.com/store/apps/details?id=samf.lwc

[3] AgriAPP https://play.google.com/store/apps/details?id=com.criyagen&hl=en

[4] Agrimind https://play.google.com/store/apps/details?id=br.com.agrimind.compendioagrimind

[5] Agro klub http://www.agroklub.com

[6] Agro klub –kukuruz http://www.agroklub.com/sortna-lista/zitarice/kukuruz-115/

[7] Kukuruz http://www.savjetodavna.hr/adminmax/publikacije/kukuruz.pdf

[8] Agro klub –pšenica https://www.agroklub.com/sortna-lista/zitarice/psenica-108/

[9] Pšenica http://www.savjetodavna.hr/adminmax/publikacije/pšenica.pdf

[10] Android Studio https://en.wikipedia.org/wiki/Android_Studio

[11] Scott Means W. XML in a Nutshell Treće izdanje. SAD: Naklada O'Reilly Media; 2009.

[12] World Wide Web Consortium https://www.w3.org

[13] Java https://bs.wikipedia.org/wiki/Java_(programski_jezik)

[14] Java_Programiranje_1_i_2, skripta

[15] Techtarget http://searchsqlserver.techtarget.com/definition/SQL-Server

[16] Link za preuzimanje Android Studio https://developer.android.com/studio/index.html

[17] Udemy- stranica za učenje programiranja u Android Studio https://www.udemy.com/

[18] AMC centar za učenje https://learning.acm.org/webinar/index.cfm

[19] IEEE obrazovne aktivnosti
http://www.ieee.hr/ieeesection/obrazovne_aktivnosti?@=2jd8r#news_11380

[20] IEEE edukacijski centar https://www.ieee.org/education_careers/index.html

SAŽETAK

Cilj ovog završnog rada bio je ostvariti aplikaciju koja će pomoći poljoprivredniku prilikom sijanja, održavanja i žetve njegovog usjeva što je i ostvareno. Korisniku se omogućuje unos parametara njegovog usjeva i trajno spremanje na udaljenu bazu podataka te preporuku stručnjaka kako bi bilo optimalno koristiti zaštitna sredstva prema parametrima bolesti, korova i nametnika. Aplikacija je ostvarena u razvojnom okruženju Android Studio, a zasniva se na opisnom jeziku koji definira izgled zaslona i izvornom jeziku koji povezuje aplikaciju s bazom podataka i s aktivnostima unutar nje. Testiranje aplikacije pokazuje da aplikacija radi točno u svim područjima, te korisniku pruža korisne savjete prilikom njegovanja usjeva. U budućnosti aplikacija ima mjesta za poboljšanje, te dodavanje novina koje bi dodatno pomogle poljoprivredniku u praćenju usjeva.

Ključne riječi: Android, Java, mobilna aplikacija, poljoprivreda, zaštitna sredstva.

ABSTRACT

The main purpose of this bachelor's thesis was to develop a computer application that would help agriculturists with sowing, maintaining and harvesting crop. The purpose of this bachelor's thesis was successfully accomplished. The application user is presented with user interface allowing them to input various crop parameters, which would then be stored in a remote database. The application would respond with expert advice as to how to optimally apply crop protective agents based on various crop ailment and parasite parameters. The Application was developed by the use of Android Studio IDE (Integrated Development Environment) and is based on a markup language which defines the presented visual properties of the application, and is also based on program source code that connects the application with the database and activities therein. Testing the application has shown that it has performed within expectation in all cases and that the expert advice was correctly issued. Of course, there is room for improving this solution and program updates that could substantially improve the uses this application offers to agriculturists.

Key words: Android, Java, mobile application, agriculture, protective resources.

ŽIVOTOPIS

Toni Horvat, rođen 10.01.1995. godine u Osijeku. Živi u Čepinu s obitelji. Završio srednju Elektrotehničku i prometnu školu u Osijeku 2013. godine te sudjelovao na državnom natjecanju iz elektrotehnike u Zagrebu. Trenutno je apsolvent treće godine preddiplomskog studija računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek. Posjeduje vještine rada na računalu (Word, Excel, Power Point, Internet, C, C++, C#, MySQL) te posjeduje znanje Java programskog jezika 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 izrađene aplikacije