BOTANIAI

Projektna dokumentacija

2022/2023

UM FERI

Lahovnik, Borak, Drozg, Jevšnik

Kazalo vsebine

[Kazalo slik 1](#_Toc136022522)

[Uvod 2](#_Toc136022523)

[Namen 2](#_Toc136022524)

[Izbrane tehnologije 2](#_Toc136022525)

[Razdelitev dela 2](#_Toc136022526)

[Organizacija dela 2](#_Toc136022527)

[Predvideni uporabniki 3](#_Toc136022528)

[Struktura podatkovne baze 5](#_Toc136022529)

[Podatki o rastlini 5](#_Toc136022530)

[Podatki o uporabniku 6](#_Toc136022531)

# Kazalo slik

[Slika 1: Prvotni diagram primerov uporabe 3](#_Toc136022532)

[Slika 2: Prilagojen diagram primerov uporabe 4](#_Toc136022533)

[Slika 3: Struktura dokumenta Plant 5](#_Toc136022534)

[Slika 4: Struktura dokumenta User 6](#_Toc136022535)

# Uvod

## Namen

Ideja je nastala zaradi pomembnosti rastlin in njihove pravilne skrbi. Rastline nas obkrožajo povsod na svetu, zaradi česar je pomembno, da vemo katere rastline nas neposredno obdajajo in ali nas lahko kakšne rastline potencialno življenjsko ogrožajo. S tem namenom smo ustvarili aplikacijo, ki omogoča prepoznavo različnih vrst rastlin glede na fotografijo. Na podlagi fotografije nato klasifikacijski model napove vrsto rastline in o tej vrsti si lahko preberemo več informacij. Hkrati omogoča shranjevanje rastlin v svoj virtualni vrt, kjer si lahko tudi izbere na koliko dni je potrebno rastlino zaliti in prvi dan zalivanja, na podlagi česar bo nato uporabnik prejemal potisna obvestila (push notifications). Aplikacija je torej namenjena vsem ljubiteljem rastlin, vsem, ki želijo izvedeti več informacij o razičnih vrstah rastlin ter vsem, ki se želijo naučiti več o pravilni skrbi za posamezno rastlino.

## Izbrane tehnologije

Za programske jezike smo izbrali python za klasifikacijo, JavaScript za zaledni sistem in TypeScript za mobilno aplikacijo. Na področju klasifikacije smo si pomagali z uporabo knjižnjic, kot so Tensorflow, Numpy, Scikit-learn, matplotlib in seaborn. V sklopu zalednega sistema smo izbrali ogrodje ExpressJS, s pomočjo različnih knjižnjic, kot je Mongoose za povezavo na MongoDB podatkovno bazo, dotenv za okoljske spremenljivke in Swagger za dokumentacijo API-ja. V sklopu mobilne aplikacije smo uporabili ogrodje React Native, s pomočjo različnih knjižnjic, kot je Expo, zaradi česar je bilo možno aplikacijo uporabljati v ExpoGo emulatorju, sicer smo pa uporabili nekatere druge pomembne knjižnjice, kot je expo-camera za uporabo kamere v aplikaciji in firebase, za avtentikacijo in uporabo firebase functions.

Za te tehnologije smo se odločili, zato ker jih vsi vsaj nekoliko poznamo, kar je olajšalo razvoj in zaradi česar je bilo reševanje težav, ki so nastopile lažje.

## Razdelitev dela

Ena oseba se je posvetila razvoju zalednega sistema, ena oseba se je posvetila razvoju klasifikacijskega modela, preostali dve osebi pa sta se posvetili razvoju mobilne aplikacije. Takšno razdelitev oziroma organizacijo znotraj skupine smo izbrali, saj je bilo največ dela ravno pri razvoju mobilne aplikacije.

## Organizacija dela

Pred začetkom razvoja projekta smo se v skupini dogovorili kdo bo delal na katerem delu projekta in katere tehnologije bodo izbrane. Med razvojem smo se na začetku sestali in izvedli uvoden sestanek, kjer smo šli skozi morebitne pomisleke in razložili morebitne nejasnosti. Tekom razvoje smo se vsak teden sestali na tedenskih Scrum sestankih, sicer so sestanki potekali po potrebi. V kolikor je bilo potrebnih več sestankov na teden smo jih izvedli. Za dogovarjanje smo uporabili aplikacijo Discord, kjer smo ustvarili skupinski pogovor in izvajali skupinske klice.

# Predvideni uporabniki

Predvideni uporabniki, funkcionalnosti aplikacije in integracije oziroma zunanje storitve so prikazani v naslednjem diagramu primerov uporabe, kot je bilo prvotno zamišljeno.

A picture containing text, diagram, line, drawing

Description automatically generated

Slika 1: Prvotni diagram primerov uporabe

Ker pa smo tekom razvoja naleteli na nekompatibilnost nekaterih knjižnjic, ni bilo možno vgraditi klasifikacijskega modela, zaradi česar je bilo potrebno diagram primerov uporabe nekoliko prilagoditi.

A picture containing text, diagram, line, drawing

Description automatically generated

Slika 2: Prilagojen diagram primerov uporabe

# Struktura podatkovne baze

## Podatki o rastlini

Podatki o rastlinah so zbrani v kolekciji imenovani »Plant« in imajo naslednjo strukturo:

* \_id – unikatni identifikator,
* latin – latinsko ime rastline,
* common – navadno ime rastline,
* description – daljši opis rastline,
* watering – na koliko časa bi naj rastlino zalivali,
* sunlight – kolikšni količini sončne svetlobe bi naj bila izpostavljena,
* plantingTime – v katerih letnih časih rastlino sadimo,
* soil – v kakšnih prsteh rastlina uspeva,
* wateringDetail – podrobneje opisan način zalivanja,
* fertilization – na kakšen način lahko gnojimo rastlino,
* toxicity – ali je rastlina strupena in za koga,
* folder\_num – številka direktorija, v katerem se nahajajo primerki te rastline.

A screenshot of a computer

Description automatically generated with medium confidence

Slika 3: Struktura dokumenta Plant

## Podatki o uporabniku

Podatki o uporabnikih so zbrani v kolekciji imenovani »User« in imajo naslednjo strukturo:

* \_id – unikatni identifikator,
* name – ime uporabnika,
* surname – priimek uporabnika,
* email – e-poštni naslov uporabnika,
* notifications – ali želi uporabnik prejemati potisna obvestila ali ne,
* history – zbrani rezultati klasifikacijskega modela, slika uporabljena za napoved, datum napovedi in razultat klasifikacijskega modela,
* personalGarden – zbrane rastline uporabniškega virtualnega vrta in podatki, ki so potrebni za skrb posamezne rastline.

A screen shot of a computer program

Description automatically generated with low confidence

Slika 4: Struktura dokumenta User