

Mendelova univerzita v Brně  
Provozně ekonomická fakulta

---

# **Smartphone aplikácia pre poloautomatickú identifikáciu objektov**

**Bakalárska práca**

Vedúci práce:  
Ing. Dita Dlabolová

Matej Kušnier

Brno 2016



Moja veľká vďaka patrí Ing. Ondrejovi Veselému, ktorý bol základným podnetom pre vznik mojej práce, a taktiež poskytol dáta a cenné rady. Ďalej Ing. Dana Michalcovej, ktorá dodala dáta nevyhnutné pre vznik aplikácie. Vďaka patrí aj vedúcej mojej práce Ing. Dite Dlabolovej za venovanie jej času na výpomoc pri spracovaní textu. V neposlednom rade by som chcel poďakovať svojej rodine a známym za veľkú podporu, najmä môjmu bratovi Bc. Jakubovi Kušnierovi za inšpiráciu a ochotu pomôcť pri riešení problémov.

### **Čestné prehlásenie**

Prehlasujem, že som túto prácu: **Smartphone aplikácia pre poloautomatickú identifikáciu objektov**

vypracoval samostatne a všetky použité pramene a informácie sú uvedené v zozname použitej literatúry. Súhlasím, aby moja práca bola zverejnená v súlade s § 47b zákona č. 111/1998 Sb., o vysokých školách v znení neskorších predpisov, a v súlade s platnou *Smernicou o zverejňovaní vysokoškolských záverečných prácach*.

Som si vedomý, že se na moju prácu vzťahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brne má právo na uzatvorenie licenčnej zmluvy a užitie tejto práce ako školského diela podľa § 60 odst. 1 Autorského zákona.

Ďalej sa zaväzujem, že pred spísaním licenčnej zmluvy o využití diela inou osobou (subjektom) si vyžiadam písomné stanovisko univerzity o tom, že predmetná licenčná zmluva nieje v rozpore s oprávnenými záujmami univerzity, a zaväzujem sa uhradiť prípadný príspevok na úhradu nákladov spojených so vznikom diela, a to až do ich skutočnej výšky.

V Brne dňa 23. mája 2016

.....

**Abstract**

Kušnier M. Smartphone application for semi-automatic object identification. Bachelor thesis. Brno, 2016

Thesis is about design and implementation of an iOS mobile application for plant identification through its traits and location. The application uses an element of information theory, information entropy, to improve identification.

**Keywords**

Smartphone application, iOS, botanical key, information theory, information entropy

**Abstrakt**

Kušnier M. Smartphone aplikácia pre poloautomatickú identifikáciu objektov. Bakalárska práca. Brno, 2016

Táto bakalárska práca sa zaoberá návrhom a implementáciou mobilnej aplikácie pre platformu iOS na identifikáciu rastlín pomocou znakov a polohy. Táto aplikácia používa prvok informačnej teórie, informačnú entropiu, ako prostriedok na zlepšenie kvality vyhľadávania.

**Klíčové slová**

Smartphone aplikácia, iOS, botanický kľúč, teória informácií, informačná entropia

## Obsah

<b>1</b>	<b>Úvod</b>	<b>8</b>
1.1	Cieľ práce . . . . .	8
<b>2</b>	<b>Botanický kľúč</b>	<b>9</b>
2.1	Určovanie znakov pomocou botanického kľúča . . . . .	9
2.2	Problematika botanického kľúča . . . . .	9
2.3	Existujúce riešenia . . . . .	9
	Knižné podoby . . . . .	9
	Webové aplikácie . . . . .	9
	Natívne mobilné aplikácie . . . . .	10
<b>3</b>	<b>Teória informácie</b>	<b>11</b>
3.1	Markov reťazec . . . . .	11
3.2	Informačný prenos . . . . .	11
3.3	Prenos diskretného signálu bez šumu . . . . .	13
	Kapacita prenosového kanálu . . . . .	13
	Informačný zdroj . . . . .	13
3.4	Informačná entropia . . . . .	13
<b>4</b>	<b>Technológie</b>	<b>16</b>
4.1	Operačný systém iOS . . . . .	16
4.2	XCode . . . . .	16
	Textový editor . . . . .	16
	Debugger . . . . .	16
	Interface Builder . . . . .	16
4.3	Objective-C . . . . .	17
4.4	Model view controller . . . . .	17
4.5	Core data VS SQLite . . . . .	17
	Core data . . . . .	17
	SQLite . . . . .	18
4.6	Core Location a Map Kit . . . . .	19
	Core Location . . . . .	19
	Map Kit . . . . .	19
	Mercatovo zobrazenie . . . . .	19
	Súradnicové systémy . . . . .	20
4.7	JSON . . . . .	20
<b>5</b>	<b>Návrh aplikácie</b>	<b>22</b>
5.1	Funkčné požiadavky . . . . .	22
5.2	Návrh identifikácie rastlín . . . . .	22
5.3	Návrh užívateľského rozhrania . . . . .	22
5.4	Návrh prechodu aplikáciou . . . . .	22

---

Hlavná obrazovka . . . . .	22
Obrazovka mapy . . . . .	24
Obrazovka znaky rastlín . . . . .	24
Obrazovka zoznam rastlín . . . . .	24
Obrazovka Detail rastliny . . . . .	24
5.5 Dátová časť aplikácie . . . . .	24
Tabuľka species_list . . . . .	24
Tabuľka photo . . . . .	24
Tabuľka trait . . . . .	24
5.6 Návrh využitia teórie informácií na urýchlenie identifikovania . . . . .	25
5.7 Návrh využitia zbieraných dát na urýchlenie a spresnenie identifikácie	27
<b>6 Implementácia</b>	<b>28</b>
6.1 Implementácia identifikácie . . . . .	28
6.2 Implementácia užívateľského rozhrania . . . . .	28
Hlavná obrazovka . . . . .	28
Obrazovka znakov . . . . .	29
Obrazovka mapy . . . . .	29
Obrazovka zoznam rastlín . . . . .	29
6.3 Implementácia zberu dát pomocou spätnej väzby . . . . .	30
<b>7 Diskusia</b>	<b>32</b>
7.1 Ďalší vývoj a vylepšenia aplikácie . . . . .	32
Rozširovanie databázy . . . . .	32
Pridávanie kritérií vyhľadávania . . . . .	32
Získavanie finančných zdrojov . . . . .	32
Lokalizácia . . . . .	32
<b>8 Záver</b>	<b>33</b>
<b>9 Literatúra</b>	<b>34</b>

# 1 Úvod

„We’re going to make some history together today.“ – Steve Jobs (Mashable, 2015)

Touto vetou Steve Jobs zahájil predstavenie prvého iPhoneu na Macworld konferencii 9. januára 2007. Koncept, ktorý predstavil, sa iba vzdialene podobal tomu, čo sa v tej dobe nazývalo smartphonmi. Tento koncept sa udržal dodnes a stal sa veľmi populárnym. Okrem firmy Apple začali smartphony vyrábať takmer všetky firmy vyrábajúce mobilné telefóny a dnes je smartphone najrozšírenejším mobilným zariadením.

Smartphony sú populárne pre veľké množstvo ponúkaných funkcií, jednoduchosť ovládania a prenositeľnosť. Vďaka týmto vlastnostiam disponujú potenciálom pre vznik aplikácií uľahčujúcich každodenné činnosti ako napríklad komunikácia, ale aj voľnočasové aktivity.

Spoznávanie prírody je veľmi obľúbenou činnosťou nie len medzi odbornou, ale aj medzi laickou verejnosťou. Táto práca sa zaoberá vytvorením aplikácie na smartphony, ktorá uľahčuje rozpoznávanie rastlín pre širokú škálu užívateľov.

Pred vyhotovením práce prebehlo stretnutie s vedúcim projektu Flower Checker Ondrejom Veselým a botaničkou Danou Michalcovou, kde boli stanovené základné požiadavky na funkčnosť aplikácie.

## 1.1 Cieľ práce

Cieľom práce je návrh a implementácia mobilnej aplikácie slúžiacej ako botanický kľúč na identifikáciu rastlín. Ďalej návrh a využitie čiastočnej automatizácie vyhľadávania v aplikácii pomocou teórie informácie na zrýchlenie a spresnenie identifikácie. Nakoniec implementácia spätnej väzby na zber dát.

Aplikácia bude vytvorená v spolupráci s projektom botanickagalerie.cz, konkrétne s botaničkou Ing. Danou Michalcovou a projektom flowerchecker.com konkrétne s Ing. Ondrejom Veselým.

Na identifikáciu sa budú používať dve základné komponenty:

- **Znaky rastliny** – Viditeľné znaky, ako napríklad farba kvetu a tvar listu, ale aj znaky ako doba kvitnutia. Tieto dáta pochádzajú z databázy webu botanickagalerie.cz.
- **Poloha rastliny** – Poloha zariadenia, z ktorej sa pomocou webu flowerchecker.com získa zoznam najčastejšie sa vyskytujúcich rastlín.

Na automatizáciu vyhľadávania bude využitá znalosť polohy rastliny a štatistické dáta o znakoch slúžiacich na identifikáciu. Dáta získané zo spätnej väzby budú slúžiť na zlepšenie automatizácie.



## 2 Botanický klúč

Botanický klúč je pomôcka pre laickú, ale aj odbornú verejnosť slúžiaca k určovaniu rastlín na základe jej vonkajších znakov.

### 2.1 Určovanie znakov pomocou botanického klúča

Určovanie rastlín pomocou botanického klúča má zväčša podobu otázok, na ktoré osoba odpovedá, čím znižuje množinu rastlín, o ktoré sa môže jednať, až pokiaľ sa nedostane ku hľadanému výsledku. Medzi najčastejšie používané znaky pri určovaní o akú rastlinu sa jedná, patria napríklad listy, kvety, stonka, ale aj poloha rastliny, nadmorská výška, podnebie a pod.

### 2.2 Problematika botanického klúča

Základné požiadavky na botanický klúč sú presnosť a rýchlosť určovania. Na dosiahnutie týchto požiadaviek je potrebný výber správnych otázok. Je potreba myslieť aj na chyby, ktorých sa užívatelia pri výbere dopúšťajú. Týchto chýb sa dopúšťajú najmä laici z dôvodu nedostatku zreteľa na jednotlivé znaky, alebo zlým výkladom znakov. Napríklad, ak používateľ určuje mak, ktorého kalich pred rozvinutím kvetu opadáva, považuje často jeho korunu za jednoduché okvetie, pretože kalich nevidí (Faustus, Polívka, 1984). Práve zlepšením rýchlosti a presnosťou identifikácie rastlín sa bude práca zaoberať. Budú na to použité prostriedky popísané v ďalších kapitolách.

### 2.3 Existujúce riešenia

V súčasnej dobe existuje veľké množstvo riešení problematiky botanického klúča v rôznych podobách.

#### **Knižné podoby**

Knižné podoby sú najstaršie riešenia botanického klúča. Vznikali najmä pred masovým používaním osobných počítačov. Jeden z príkladov knižnej podoby je kniha *Botanický klúč* (Kochan 2013). Ich nevýhodou je ich nekompaktnosť, najmä ak obsahujú väčšie množstvo záznamov a taktiež pomalšie listovanie medzi stranami.

#### **Webové aplikácie**

Webová aplikácia je dnes asi najrozšírenejšia podoba. Tieto formy riešia problematiku botanického klúča viacerými spôsobmi. Niektoré weby, ako napríklad *botanickagalerie.cz*, určujú rastliny na základe vonkajších znakov rastlín. Iné weby používajú na identifikáciu fotografiu rastliny, či už pomocou rozpoznávania obrazu, alebo pomocou odborníkov. Výhody webových aplikácií sú najmä možnosť

ukladania väčšieho množstva záznamov bez nutnosti zmenšenia kompaktnosti pre koncového užívateľa a multiplatformné využitie. Najväčšou nevýhodou je nutnosť internetového pripojenia.

### **Natívne mobilné aplikácie**

V súčasnej dobe najviac rozširujúca sa platforma už taktiež ponúka rôzne formy botanického kľúča, dokonca niektoré webové aplikácie, ako napríklad flowerchecker, ponúkajú natívnu mobilnú aplikáciu.

Mobilné aplikácie, podobne ako tie webové, ponúkajú rôzne prístupy k riešeniu problematiky. Aplikácia Garden používa rozpoznávanie obrazu na identifikáciu rastliny. Aplikácia iPflanzen ponúka možnosť identifikácie na základe znakov rastliny. Takýto spôsob identifikácie rastlín je pre užívateľa veľmi pohodlný, avšak býva veľmi nepresný. Výhodami aplikácií sú kompaktnosť a možnosť uchovávania dát v zariadení. Nevýhodami aplikácií sú najčastejšie potreba pripojenia k internetu, alebo potrebná určitá znalosť v odbore pre presnú identifikáciu znakov.

V českom a slovenskom jazyku je iba veľmi malé množstvo aplikácií s problematikou botanického kľúča. Jedným s príkladov je aplikácia botanický kľúč (Kozák, 2014). Táto aplikácia určuje rastliny iba na základe vonkajších znakov. K dispozícii je 7 typov vonkajších znakov. Užívateľ však musí zadať presné údaje o danom znaku aby bola vyhledaná správna rastlina, čo sťažuje identifikáciu najmä pre chyby v odhade používateľa.

### 3 Teória informácie

Teória informácie je matematická disciplína, najčastejšie používaná v komunikácii, ktorá skúma informácie, kódovanie a prenos. Jej základy položil Shannon vo svojej publikácii „A Mathematical Theory of Communication“, v ktorej vysvetlil komunikáciu pomocou Markovho reťazca (Markov Chain).

Základnou jednotkou informácie je bit, ktorý má 2 hodnoty. Jeden bit informácii sa dá uložiť napríklad pomocou klopných obvodov. Na uloženie  $N$  bitov je potreba  $N$  súčiastok a keďže bit má 2 hodnoty  $M$  súčiastok dokáže uložiť  $2^M$  hodnôt. Počet bitov potrebných na uloženie  $N$  hodnôt teda vyjadríme pomocou logaritmu o základe 2 (Shannon, 1948).

$$\log_2 N$$

#### 3.1 Markov reťazec

Markov reťazec je stochastický proces s konečnou množinou stavov  $S = \{s_1, s_2, \dots, s_r\}$ . Proces začína v jednom z týchto stavov a prechádza rekurzívne z jedného stavu do druhého. Proces prechádza zo stavu  $s_i$  do stavu  $s_j$  s pravdepodobnosťou  $p_{ij}$  a táto pravdepodobnosť je nezávislá na predošlom stave, v ktorom sa nachádzal. Táto pravdepodobnosť sa nazýva prechodová pravdepodobnosť. Proces môže ostať v stave  $s_i$  s prechodovou pravdepodobnosťou  $p_{ii}$  (Grinstead, 2006).

$$P = \begin{pmatrix} 0.65 & 0.28 & 0.07 \\ 0.15 & 0.67 & 0.18 \\ 0.12 & 0.36 & 0.52 \end{pmatrix}$$

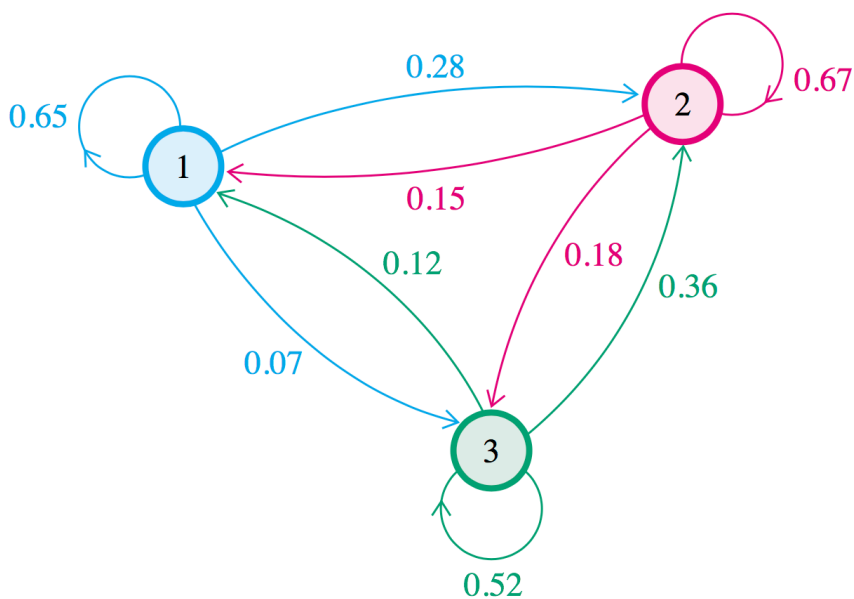
#### 3.2 Informačný prenos

Na prenos informácií zo zdroja ku príjemcovi sa používajú komunikačné systémy, ktorých schéma je znázornená na obrázku 1. Komunikačné systémy sa delia na diskkrétne, spojité a zmiešané (Shannon, 1948).

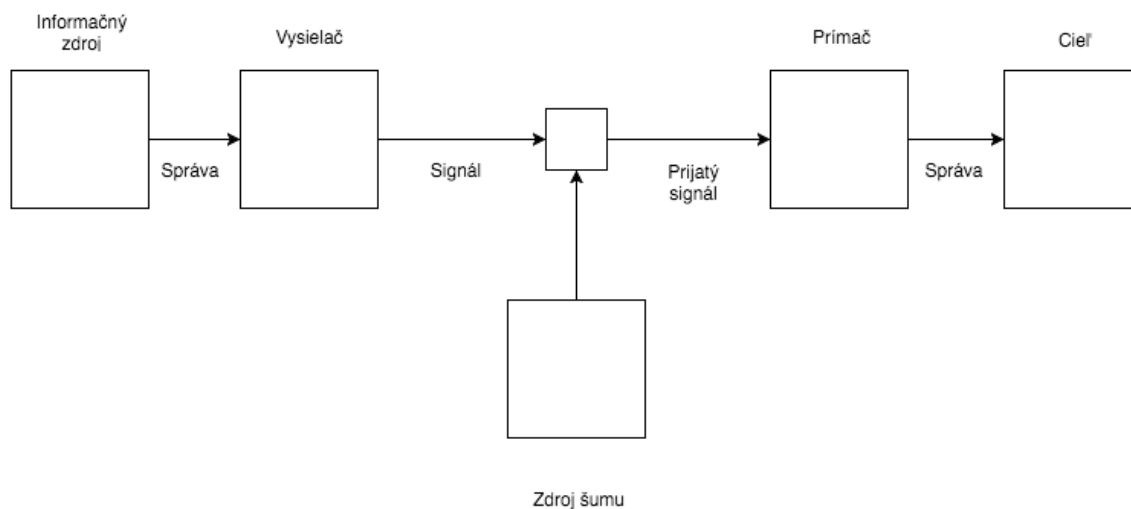
- **Diskkrétne.** V diskkrétnych systémoch je správa aj signál sekvencia diskkrétnych symbolov. Typickým príkladom je telegraf, kde správa je sekvencia znakov a signál sekvencia bodiek a medzier.
- **Spojité.** V spojitých systémoch je správa aj signál spojitá funkcia, napríklad rádio a televízia.
- **Zmiešané.** V zmiešaných systémoch sa vyskytuje diskkrétny aj spojitý prvok. Príkladom je PCM.

Komunikačný systém sa skladá z 5 častí:

- **Informačný zdroj** je zdrojom správy určenej na prenos. Správa môže mať rôzne podoby, napríklad sekvencia znakov, spojitá funkcia a pod.



Obrázok 1: grafické znázornenie Markovho reťazca, (Grinstead, 2006)



Obrázok 2: schéma komunikačného systému z „A Mathematical Theory of Communication“ (Shannon, 1948) str. 3

- **Vysielač** správu spracuje do podoby vhodnej na prenos pomocou prenosového kanálu.
- **Prenosové médium** slúži na prenos kódovanej správy od vysielača ku prijímaču. Prenosový kanál môže mať rôzne podoby, napríklad koaxiálny kábel, vzduch, optický kábel a pod.

- **Prijímač** zachytáva signál poslaný vysielačom a dekóduje ho do pôvodnej podoby.
- **Cieľ** je osoba alebo vec, pre ktorú je správa určená.

### 3.3 Prenos diskrétného signálu bez šumu

#### Kapacita prenosového kanálu

V kanáli, kde všetky prenášané symboly majú rovnakú prenosovú dĺžku a prenášame znaky z množiny, ktorá obsahuje 64 znakov, každý prenášaný znak obsahuje 6 bitov informácií. Ak je kanál schopný prenášať  $n$  bitov, má prenosovú kapacitu  $6n$  bitov. To však neznamená, že kanál prenáša vždy  $6n$  bitov.

#### Informačný zdroj

Pri prenose textu, napríklad pomocou telegrafu, ide o prenos znakov vo forme písmen. Všetky písmená však nemajú rovnaké zastúpenie. V texte písanom v slovenskom jazyku má písmeno „o“ väčšie zastúpenie ako písmeno „n“ a dvojica hlások „pr“ má vyššie zastúpenie ako dvojica hlások „la“ (Štefánik 2014). Táto skutočnosť umožňuje šetriť prostriedky pri prenose vhodne zakódovanej správy. Budeme premýšľať o zdroji ako o generátore textu znak po znaku. Postupnosť vygenerovaných znakov závisí na určitej pravdepodobnosti. Takýto proces generovania postupnosť znakov na základe pravdepodobností sa nazýva stochastický proces. V slovenskom jazyku však nie je pravdepodobnosť výskytu určitého znaku úplne nezávislá. Napríklad po medzere je nulová pravdepodobnosť, že ďalší vygenerovaný znak v poradí bude opäť medzera. Po spoluhláske je pravdepodobnosť, že bude nasledovať samohláska vyššia, ako pravdepodobnosť ďalšej spoluhlásky. Toto sa dá popísať pomocou komplikovanejšej štruktúry, kde pravdepodobnosť vygenerovania určitého znaku je závislá na predošlom znaku. Táto štruktúra by sa dala definovať ako množina prechodových pravdepodobností  $p(i,j)$ ,  $\forall(i,j) \in S$ , kde  $S$  je množina všetkých možných znakov. Platí:

$$p(i) = \sum_j p(i,j) = \sum_j p(j,i)$$

$$\sum_i p(i) = \sum_{i,j} p(i,j) = 1$$

Podobne ako so znakmi môžeme pracovať aj so slovami. Popísaná štruktúra odpovedá Markovmu reťazcu.

### 3.4 Informačná entropia

Uvažujeme o informačnom zdroji ako o diskretnom Markovom reťazci produkujúcom prvky s konečnej množiny  $S = \{s_1, s_2, s_3, s_4\}$ . Dokážeme zmerať hodnotu neurčitosti a tým počet bitov potrebných na prenos tejto správy.

V prípade, že pravdepodobnosť jednotlivých prvkov je rovnaká  $s_i = 1/4$ , na zakódovanie správy sú potrebné minimálne 2 bity ( $\log_2 4$ ). Na prenos 100 znakov je teda potrebných 200 bitov.

Ak pravdepodobnosť jednotlivých prvkov nie je rovnaká, je možné znížiť počet bitov potrebných na ich prenos. Berieme príklad, kde pravdepodobnosti prvkov sú  $s_1 = 1/2$ ,  $s_2 = 1/4$ ,  $s_3 = 1/8$ ,  $s_4 = 1/8$ . Keďže prvok má  $s_1$  značne vyššiu pravdepodobnosť výskytu, môžeme do prvého bitu zakódovať informáciu, či sa jedná o tento prvok, alebo jeden z ostatných troch prvkov množiny. V druhom bite bude obsiahnutá informácia, či sa jedná o prvok  $s_2$  s pravdepodobnosťou  $1/4$  a tretí bit bude obsahovať informáciu, či sa jedná o prvok  $s_3$ , alebo  $s_4$  (Grafické zobrazenie popísaného rozhodovacieho stromu je znázornené na obrázku 4.). V takomto prípade bude na prenos prvku priemerne stačiť  $0,5 * 1 + 0,25 * 2 + 0,125 * 3$ , čo je 1,375 bitu. Na prenos 100 znakov bude namiesto 200 potrebných priemerne 137,5 bitov. Priemerný počet bitov potrebných na prenos prvku sa nazýva informačná entropia. Entropia sa vyjadří ako:

$$H = \sum_{i=1}^n p_i \cdot U$$

kde  $U$  predstavuje úroveň rozhodovacieho stromu, v ktorom sa nachádza prvok  $i$ . Keďže ide o binárny rozhodovací strom, úroveň stromu sa určí pomocou logaritmu o základe 2 (Khan Academy, 2016).

$$U = \log_2 N$$

kde  $N$  predstavuje počet prvkov v danej úrovni, ktorý sa vyjadří pomocou pravdepodobnosti

$$N = \frac{1}{p_i}$$

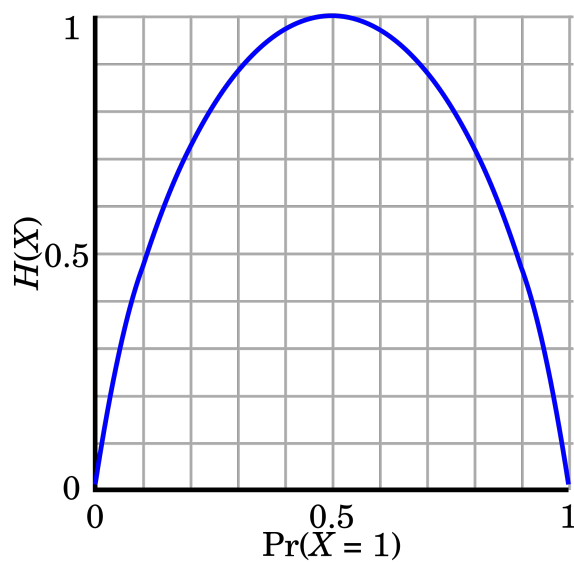
Po dosadení do vzorca vznikne:

$$H = \sum_{i=1}^n p_i \cdot \log_2 \frac{1}{p_i}$$

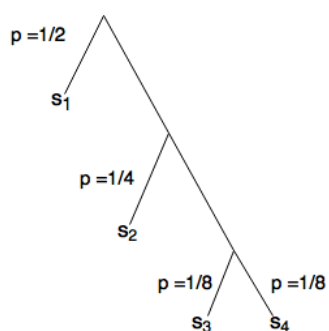
Čo sa dá taktiež vyjadriť ako:

$$H = - \sum_{i=1}^n p_i \cdot \log_2(p_i)$$

Na obrázku 3 je znázornený graf závislosti informačnej entropie na pravdepodobnosti výskytu prvku.



Obrázok 3: grafické závislosti entropie na pravdepodobnosti výskytu prvku, (Shannon, 1948)



Obrázok 4: grafické znázornenie rozhodovacieho stromu zakódovania nepravidelnej množiny

## 4 Technologíe

### 4.1 Operačný systém iOS

iOS je mobilný operačný systém od firmy Apple Inc. Je určený výhradne pre zariadenia iPhone a iPad.

Na vývoj natívnych aplikácií pre operačný systém iOS sa používajú programovacie jazyky Objective-C a Swift. Oficiálne vývojové prostredie sa nazýva XCode, existujú však aj alternatívy ako napríklad AppCode od JetBrains. Prostredie vývoja je navrhnuté podľa konceptu MVC (Model, View, Controller).

### 4.2 XCode

XCode je integrované vývojárske prostredie od spoločnosti Apple Inc. Poskytuje niekoľko nástrojov, ktoré výrazne uľahčujú programovanie:

#### Textový editor

Textový editor výrazne zjednodušuje prácu vývojára vďaka svojim schopnostiam:

- zarovnávanie textu podľa vnorenia
- farebné odlíšenie rôznych prvkov programovacieho jazyka, objective-c, ale aj C a C++
- automatické zvýrazňovanie chýb pri písaní
- zobrazovanie návrhov

#### Debugger

Pri spustení programu pomocou XCode sa automaticky spustí aj debugger. Debugger dáva možnosť vidieť všetky objekty a hodnoty, bežiacie vlákna a jednotlivé sekcie pamäte. Taktiež je možnosť stanoviť breakpoint, miesto v ktorom sa aplikácia pozastaví a vývojár môže posúvať aplikáciu po jednotlivých inštrukciách, aby vedel čo presne sa v aplikácii deje a mohol tak odhaliť prípadné chyby.

#### Interface Builder

Interface Builder je nástroj na budovanie grafického užívateľského rozhrania. Používa storyboardy, v ktorých sa definuje priebeh obrazoviek aplikáciou. Dáva možnosť vkladať do jednotlivých obrazoviek základné grafické prvky, ako napríklad label, tabuľka, obrázok a pod. Okrem grafických prvkov je možnosť určovania konstant, ktoré sa používajú na zarovnanie prvkov v obrazovke pomocou Auto-Layoutu.



## 4.3 Objective-C

Objective-C je objektový programovací jazyk primárne určený na písanie iOS a OSX aplikácii od spoločnosti Apple Inc. Keďže je Objective-C nadstavbou jazyka ANSII C, umožňuje vkladanie kódu napísaného v tomto jazyku. Trieda napísaná v Objective-c sa skladá z deklarácie triedy [`@interface`](obvykle sa vkladá do hlavíčkového súboru s príponou `.h`) a definície triedy [`@implementation`](obvykle sa vkladá do implementačného súboru s príponou `.m`). V prípade Objective-C 2.0 si môžete nechať automaticky vygenerovať nastavovacie metódy (`settre`) a návratové metódy (`gettre`) pomocou direktívy `@property`(Kochan 2013).

V Objective-c 2.0 je k dispozícii automatická správa pamäte (Automatic reference counter). Tá počíta referencie na objekt, ktorý je alokovaný v pamäti. Tá v prípade, že programátor neurčí explicitne inak, zvýši počet referencií na objekt pri uložení odkazu na daný objekt do premennej. Objekt ostáva alokovaný ak je počet referencií väčší ako 0. Keď počet referencií klesne na 0, ARC pamäť dealokuje.

## 4.4 Model view controller

Model view controller (MVC) je návrhový vzor, ktorý objektom priraduje jednu z troch rolí: model, view a controller. Taktiež určuje, ako jednotlivé objekty komunikujú medzi sebou. Aplikácie používajúce MVC sú prehľadnejšie, jednoduchšie sa spravujú a keďže jednotlivé časti komunikujú cez rozhranie, objekty sa dajú recyklovať.

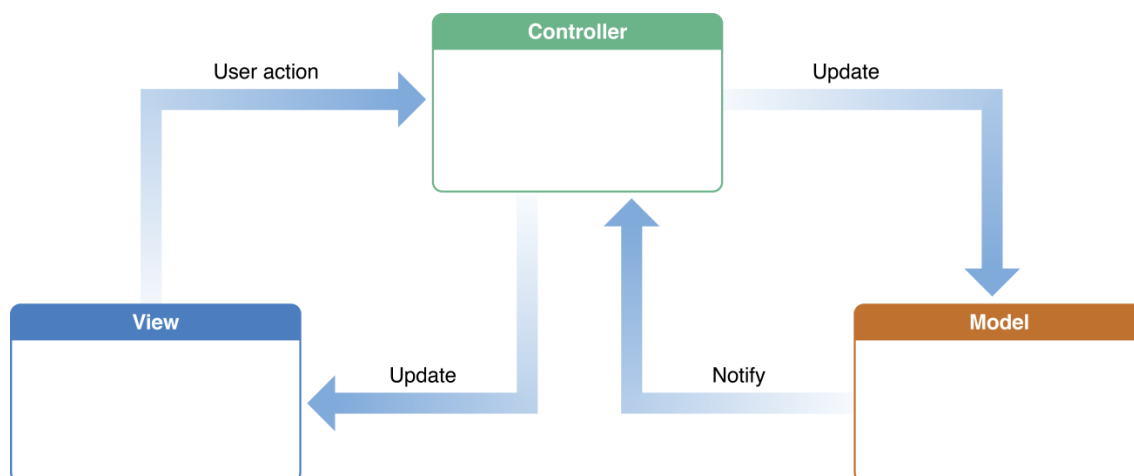
- **Model objekt** zapuzdruje dáta používané v aplikácii a stará sa o ich správu. Model objekty by mali zaistiť väčšinu operácií nad dátami, ako napríklad ukladanie dát, načítavanie dát, výpočty a pod.
- **View** je jediný objekt, ktorý je viditeľný pre užívateľa. Slúži na interpretáciu dát, ktoré sú uložené v model objektoch. View objekty vedia ako sa vykresliť a ako reagovať na akciu užívateľa. View objekt nikdy nepristupuje k dátam z model objektov priamo, iba cez Controller.
- **Controller** slúži ako prostredník medzi data a view objektami.

## 4.5 Core data VS SQLite

Na uloženie perzistentných dát ponúka platforma iOS viacero možností. Najviac používané sú Core data framework a SQLite relačná databáza.

### Core data

Je framework na správu perzistentných dát pomocou objektov. Poskytuje automatizované riešenia operácií na správu a prácu s perzistentne ukladanými dátami



Obrázok 5: schéma model view controller, (Apple Developer, 2016)

pomocou objektov (Apple Developer, 2016). Core data pozostáva z niekoľkých vrstiev, takzvaný Core Data stack.

- **NSPersistentStore** je objekt, ktorý interpretuje dáta uložené na disku (Nahavandipoor 2014).
- **NSPersistentStoreCoordinator** je objekt, ktorý spravuje ukladanie a načítavanie informácií z a do persistent storu (NSPersistentStore). Coordinator je mostom medzi manage object context a persistent store.
- **NSManagedObjectModel** popisuje datový model.
- **NSManagedObjectContext** zaisťuje správu dát a vytvára kópie objektov. Tieto môžu byť ľubovoľne upravované bez toho, aby sa zmeny prejavili na uložených dátach. Všetky zmeny sú trackované, takže sa dajú vrátiť späť.

Výhody Core data:

- Core data je rýchlejšie. Najmä pri starších modeloch iPhoneov (Dr. Dobb's, 2014).
- Vďaka objektovému prístupu sa s Core data jednoduchšie pracuje (Dr. Dobb's, 2014).

## SQLite

SQLite je open source multiplatformna relačná databáza. SQLite ukladá dáta do tabuliek, ktoré obsahujú jeden alebo viacej stĺpcov. Každý stĺpec obsahuje dáta určitého dátového typu. Výhody SQLite:

- SQLite používa o 40% – 100% menej operačnej pamäte ako Core data (Dr. Dobb's, 2014),

- SQLite používa približne štyrikrát menej úložného priestoru ako Core data (Dr. Dobb's, 2014).
- Multiplatformná podpora (Dr. Dobb's, 2014).

## 4.6 Core Location a Map Kit

Core Location a Map Kit frameworky slúžia na vytváranie aplikácií pre určovanie polohy a prácu s mapou.

### Core Location

Core Location je framework, ktorý používa hardware zariadenia na zistenie súčasnej polohy zariadenia. Dostupnosť polohových služieb závisí na dostupnosti hardwaru v zariadení, ktoré musí byť povolené užívateľom. IOS zariadenie s GPS službami používa 2G, 3G, 4G, EDGE a ďalšie technológie na zistenie polohy zariadenia. Core Location ponúka niekoľko služieb na monitorovanie polohy zariadenia:

- **Štandardná lokalizačná služba** je najčastejším spôsobom získavania polohy. Funguje na OSX aj na iOS. Umožňuje užívateľovi nakonfigurovať presnosť určenej polohy a vzdialenosť, po ktorej keď sa zariadenie odchýli od pôvodnej polohy, dostane aplikácia notifikáciu o novej polohe. Po spustení služby berie do úvahy užívateľom zadané parametre a podľa nich vyberie najvhodnejší spôsob určenia polohy (Apple Developer, 2016).
- **Služba významnej zmeny polohy** vracia polohu a pošle notifikáciu na zmenu, keď nastane väčšie odchylenie od polohy. Nevýhodou tejto služby je, že nadmerne spotrebúva energiu.
- **Region monitoring** umožňuje sledovať prechod hraníc zadaných geografických regiónov a bluetooth beacon regiónov

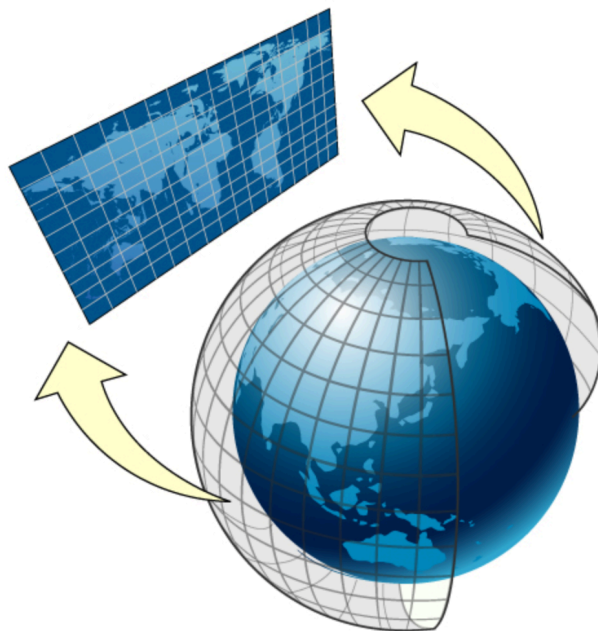
### Map Kit

Map Kit je framework, pomocou ktorého je možné do aplikácie vložiť plne funkčnú mapu. Táto mapa ponúka viacero funkcií ako štandardná mapa, satelitná mapa a ich kombinácia, približovanie, otáčanie a reagovanie na gestá užívateľa. Map Kit používa Mercatovo zobrazenie.

### Mercatovo zobrazenie

Mercatovo zobrazenie je druhom valcového zobrazenia. Valcové zobrazenia pozostávajú z poludníkov, paralelných rovnako od seba vzdialených rovných čiar znázorňujúcich zemepisnú výšku, ktoré sa pretínajú a zvierajú pravý uhol s rovnobežkami, rovnými čiarami zobrazujúcimi zemepisnú šírku. Geografické valcové zobrazenie sa

docieli tak, že rozprestrieme pomyselný valec obopínajúci zemeguľu (Apple Developer, 2016).



Obrázok 6: znázornenie premietnutia zemeguľe do roviny (Apple Developer, 2016)

### Súradnicové systémy

Map Kit špecifikuje dáta pomocou troch súradnicových systémov:

- Súradnicový bod pozostáva zo zemepisnej šírky a výšky. Súradnica je hlavným spôsobom špecifikovania polohy na zemeguli. Súradnicovému bodu odpovedá premenná `CLLocationCoordinate2D` (Snyder, 1987).
- Bod na mape predstavuje x a y súradnicu na Mercatovom zobrazení. Body sa často používajú na výpočty namiesto súradníc, pretože zjednodušujú výpočty v aplikácii. Bod na mape sa definuje premennou `MKMapPoint`
- Bod na grafickej jednotke spojený so súradnicovým systémom view objektu. Tento bod sa definuje pomocou štruktúry `CGPoint`.

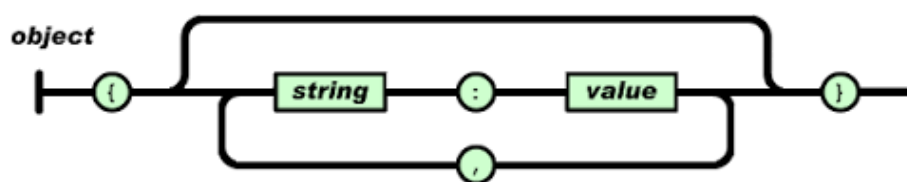
## 4.7 JSON

JSON je jednoduchý formát na výmenu dát. Je veľmi rozšírený a to najmä vďaka tomu, že funguje multiplatformne a je jednoducho čitateľný pre človeka aj počítače. JSON má dve základné dátové štruktúry

- **Kolekcia párov názov a hodnota.** Táto štruktúra je podobná dátovému typu slovník v objective-c. Každý objekt/hodnota má priradený názov.
- **Zoradený zoznam hodnôt.** Táto štruktúra je podobná dátovej štruktúre pole v jazyku objective-c.

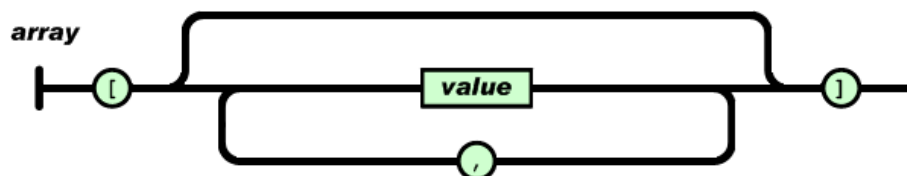
JSON používa na realizáciu týchto štruktúr:

**Objekt.** Je to neusporiadaná množina tvorená z dvojice názov a hodnota. Objekt je ohraničený zľava ľavou zloženou zátvorkou a sprava pravou zloženou zátvorkou. Za názvom nasleduje dvojbodka a jednotlivé páry názov a hodnota sú oddelené čiarkou.



Obrázok 7: štruktúra objektu

**Pole** je zoradená kolekcia hodnôt. Začína sa ľavou hranatou zátvorkou a končí pravou hranatou zátvorkou, pričom jednotlivé hodnoty sú oddelené čiarkou.



Obrázok 8: štruktúra pola

**Hodnoty** sú reťazce, čísla, booleanovské true a false, objekty a polia. Tieto štruktúry sa môžu vnorovať.

## 5 Návrh aplikácie

### 5.1 Funkčné požiadavky

Na základe dohodnutej funkčnosti s Ing. Veselým boli aplikácii stanovené tieto funkčné požiadavky:

- získavanie a úprava polohy zariadenia ako parameter vyhľadávania
- pridávanie a úprava znakov rastlín ako parametrov vyhľadávania
- zobrazenie zoznamu všetkých rastlín v databáze s názvom rastliny a fotografiou
- vyhľadávanie rastlín na základe stanovených kritérií

### 5.2 Návrh identifikácie rastlín

Na identifikáciu rastlín bude použitá poloha a vonkajšie znaky rastliny.

Samotná identifikácia nebude predstavovať orezávanie množiny možných rastlín na základe určených znakov, ale postupné pridávanie váhy rastlinám s daným znakom. Týmto spôsobom sa zamedzí možnosti, že sa vylúči správna rastlina pre chybné stanovenie znaku užívateľom.

### 5.3 Návrh užívateľského rozhrania

Návrh užívateľského rozhrania by mal prebehnúť ešte pred samotnou implementáciou. Pri návrhu je potrebné brať do úvahy rôzne užívateľské scenáre a funkcie aplikácie. Základnými prvkami užívateľského rozhrania sú:

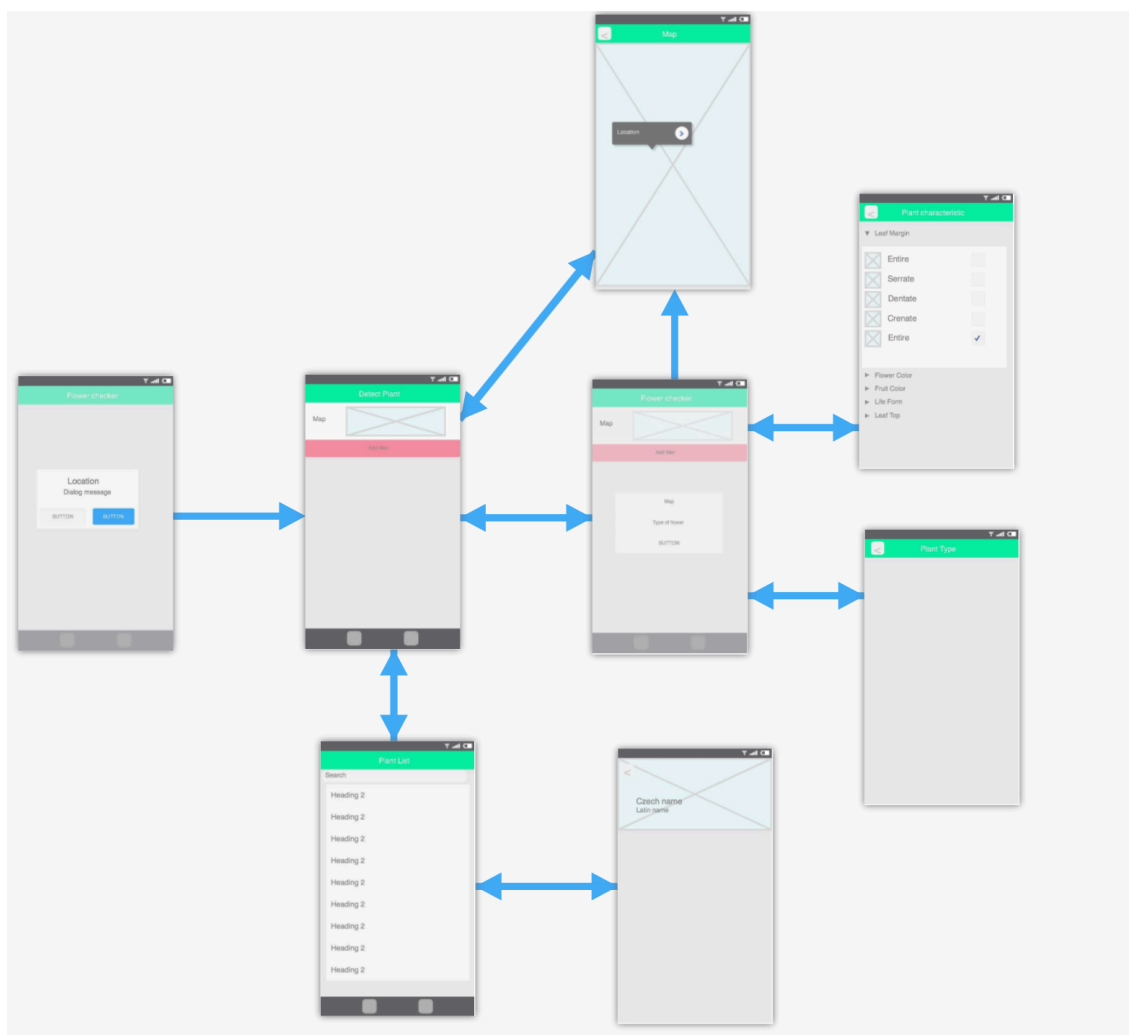
- **Prehľadnosť.** Užívateľské rozhranie má užívateľom pomôcť pochopiť a používať funkcionality aplikácie, nie jej konkurovať (?)
- **Jasnosť obsahu.** Všetok text by mal byť jasne prečitateľný, ikony by mali jasne vyjadrovať svoju podstatu, grafické ozdoby by mali vhodne zapadať do prostredia a na prvkoch by malo byť na prvý pohľad jasné, že sú interaktívne (Apple Inc., 2015).
- **Hĺbka(?)**

### 5.4 Návrh prechodu aplikáciou

Na vymodelovanie návrhu prechodu aplikáciou bola použitá služba fluidui.com. Návrh prechodu aplikáciou je znázornený na obrázku 9.

#### Hlavná obrazovka

Ihneď po spustení aplikácia prejde do obrazovky vyhľadávania. Nie je odporúčané používať splash screeny (úvodné obrazovky), ale miesto toho rovno zobrazíť najviac



Obrázok 9: Schéma prechodu obrazoviek

používanú funkciu aplikácie (Apple Inc., 2015). Hlavná obrazovka obsahuje navrchu pripnutú tabuľku s aktívnymi parametrami vyhľadávania a tlačidlom na pridanie parametra. Po kliknutí na už existujúci parameter, aplikácia prejde do upravovania parametra a po kliknutí na pridanie nového parametra sa zobrazí dialógové okno s možnosťami na pridanie.

Pod tabuľkou s parametrami bude tabuľka s výsledkami hľadania, ktoré zodpovedajú zadaným parametrom. Po kliknutí na výsledok vyhľadávania sa zobrazí detail rastliny.

Pri prvom spustení sa užívateľovi zobrazí dialógové okno, ktoré užívateľa žiada o povolenie na používanie polohových zariadení pre aplikáciu. Toto je nevyhnutnou súčasťou systému iOS.

### Obrazovka mapy

Po kliknutí na pridanie, alebo úpravu polohy ako parameter vyhľadávania, aplikácia prejde do obrazovky s mapou, ktorá ukazuje súčasnú nastavenú polohu zariadenia a ponúka možnosť zmeny polohy tapnutím na hociktoré miesto mapy.

### Obrazovka znaky rastlín

Po výbere znakov rastlín ako parametru vyhľadávania aplikácia prejde do obrazovky s vybranými znakmi. Tá obsahuje tabuľku, ktorej bunky predstavujú typy znakov. Bunky su expandovateľné. Po tapnutí sa bunka expanduje a zobrazia sa jednotlivé znaky s názvom a náčrtom znaku, ak je k dispozícii. Po tapnutí na znak sa parameter označí ako aktívny.

### Obrazovka zoznam rastlín

Obrazovka obsahuje abecedne usporiadanú tabuľku všetkých rastlín. Každá bunka tabuľky obsahuje názov rastliny a fotografiu, ak je k dispozícii. Po tapnutí na rastlinu aplikácia prejde do detailu rastliny.

### Obrazovka Detail rastliny

Obrazovka obsahuje fotografie rastliny, názov a niektoré informácie o rastline, ako je výška, doba kvitnutia a pod.

## 5.5 Dátová časť aplikácie

Databáza obsahuje jednotlivé rastliny, ich znaky a fotografie. Na obrázku 10 je znázornený dátový model aplikácie.

### Tabuľka `species_list`

Tabuľka obsahujúca jednotlivé druhy sa nazýva `species_list`. Obsahuje id druhu, latinský názov, český názov a autora.

### Tabuľka `photo`

Tabuľka `photo` obsahuje id fotografie, id druhu, názov súboru s fotografiou a meno autora fotografie.

### Tabuľka `trait`

Tabuľka `trait` slúži ako prepojenie medzi tabuľkou `species_list` a jednotlivými znakmi druhov. Stĺpce v tabuľke obsahujú id konkrétneho znaku z tabuľky daného typu znakov. Každá tabuľka konkrétneho typu znakov obsahuje id znaku, české a anglické názvy znaku. Jednotlivé typy znakov v tabuľke `trait`:



- **Altitude** – nadmorská výška rastliny, obsahuje stupeň nadmorskej výšky a nie údaj o nadmorskej výške v metroch
- **Author** – meno autora, ktorý druh pridal do databázy
- **Biotope** – biotop, v ktorom rastlina rastie
- **Blossom** – mesiace, v ktorých druh kvitne
- **Compound\_leaf** – druh zloženého listu
- **Corrola** – vzrast kuruny/okvetia
- **Corrola\_type** – typ zrastenej koruny
- **Division** – hlavná taxonomická skupina
- **Family** – čeľaď druhu
- **flower\_color** – farba kvetu
- **flower\_symetry** – symetria kvetu
- **fruit** – typ plodu
- **fruit\_color** – farba plodu
- **inflorescence** – súkvetie
- **leaf\_arrangement** – usporiadanie listu na stonke
- **leaf\_base** – báza jednoduchého listu
- **leaf\_hairiness** – ochlpenie listu
- **leaf\_margin** – okraj listu
- **leaf\_top** – vrchol jednoduchého listu
- **simple\_feaf** – tvar jednoduchého listu
- **life\_form** – vrchol jednoduchého listu

## 5.6 Návrh využitia teórie informácií na urýchlenie identifikovania

Rozmiestnenie jednotlivých znakov nie je rovnomerné. Niektoré znaky prevažujú nad ostatnými a naopak niektoré majú výrazne menšie zastúpenie. Práve táto skutočnosť sa dá využiť pri urýchlení identifikácie.

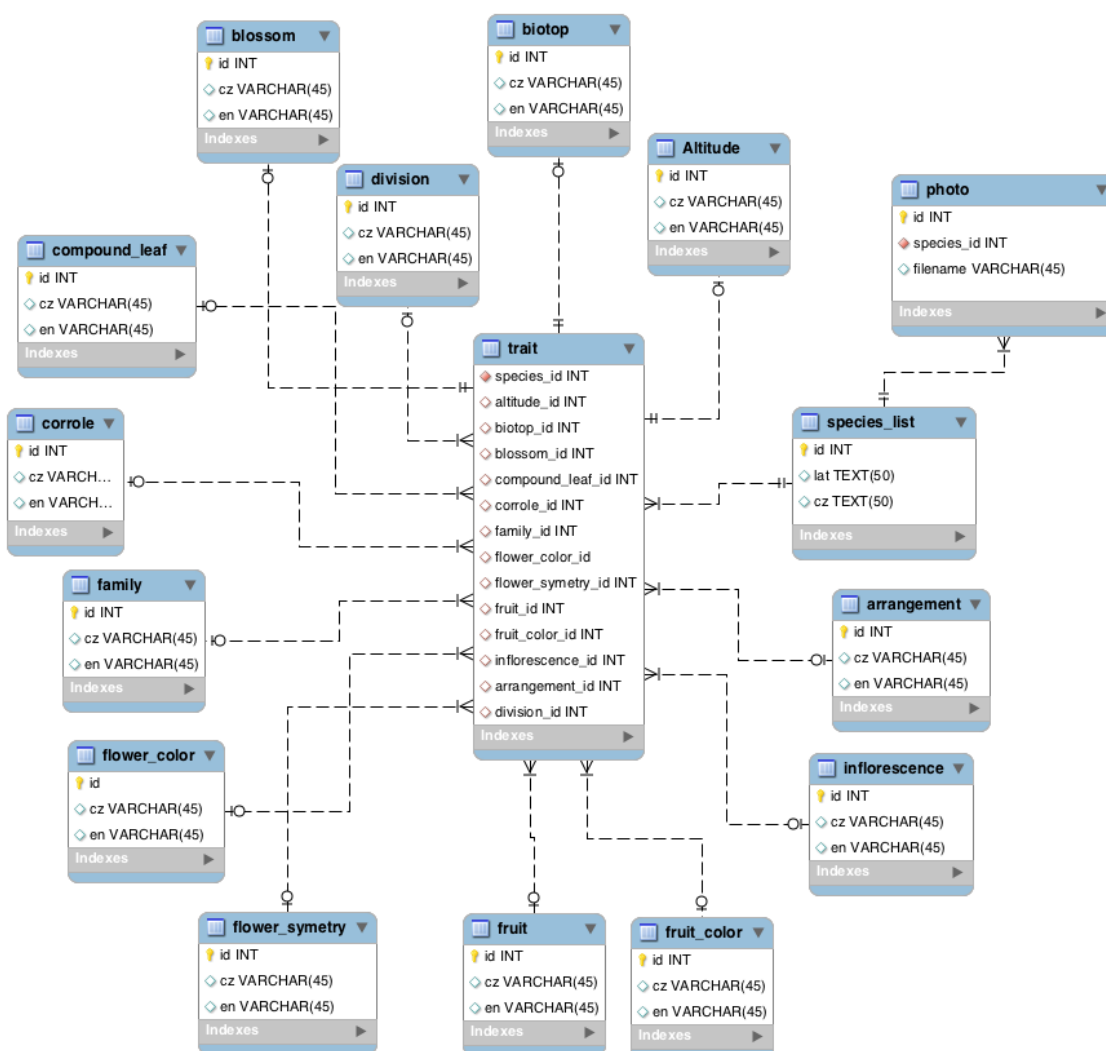
Ak bude užívateľ určovať znaky s nižšou hodnotou informácie, bude sa množina možných výsledkov orezávať rýchlejšie a dopracuje sa skôr ku výsledku.

Na určenie množstva informácie, respektíve informačnej entropie sa používa vzorec

$$H = - \sum_{i=1}^n p_i \cdot \log_2(p_i)$$

Množinu  $I = \{1, 2, \dots, i\}$  predstavujú všetky rastliny, u ktorých je záznam s daným typom znaku. Množina  $P_I = \{p_1, p_2, \dots, p_i\}$  je množina pomerov výskytu jednotlivých znakov k celkovému počtu znakov daného typu.

V tabuľke 5.6 je znázornený počet jednotlivých znakov z typu altitude a ich pomer k celkovému súčtu. Po dosadení všetkých znakov typu altitude do vzorca na výpočet entropie vyšiel výsledok 2,198.



Obrázok 10: schéma databázy

id	počet	početnosť
1	145	0,258
2	152	0,270
3	133	0,237
4	95	0,169
5	37	0,066

Tabuľka 1: Tabuľka rozmiestnenia jednotlivých znakov typu altitude a ich početnosť

## 5.7 Návrh využitia zbieraných dát na urýchlenie a spresnenie identifikácie

Zobierané dáta obsahujú informácie o akú rastlinu sa jedná a aké znaky určil v procese identifikácie užívateľ. Aplikácie je určená aj pre laickú verejnosť, ktorá má minimálne, alebo žiadne znalosti v odbore botaniky. Niektoré znaky vyžadujú určitú znalosť, preto ich užívatelia nesprávne určia, čo má za následok určenie nesprávnej rastliny.

Po nazbieraní určitého množstva dát sa určí percentuálna úspešnosť určovaní jednotlivých znakov. Podľa tejto úspešnosti dostanú jednotlivé znaky pridelené váhy od 0 do 1, čo by malo zamedziť identifikácií nesprávnych rastlín, a tým spresniť vyhľadávanie.

## 6 Implementácia

Aplikácia bola implementovaná ako natívna aplikácia pre iOS operačný systém. Implementácia bola realizovaná pomocou jazyka objective-c vo vývojárskom prostredí XCode.

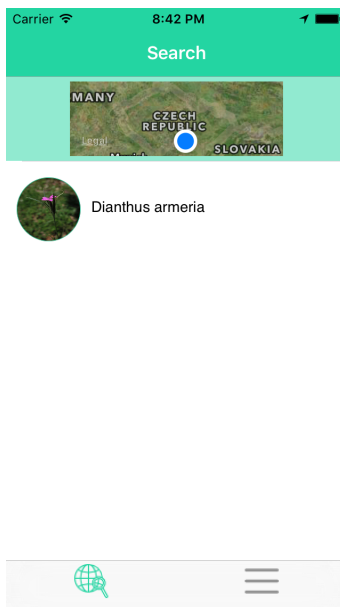
### 6.1 Implementácia identifikácie

Získaná poloha sa odošle na server FlowerCheckeru, ktorý vráti zoznam najčastejšie sa vyskytujúcich rastlín v danej oblasti s pravdepodobnosťou výskytu. Podrobné dáta o rastlinách sa vyhľadajú v lokálnej databáze a uložia sa do výsledkov, pričom pravdepodobnosť výskytu rastliny sa uloží ako váha danej rastliny. Keď užívateľ zadá znak, do výsledkov sa uložia všetky rastliny s daným znakom. Ak je už daná rastlina obsiahnutá, zvýši sa jej váha. Užívateľovi sa zobrazujú rastliny s najvyššou hodnotou váhy ako možné výsledky.

### 6.2 Implementácia užívateľského rozhrania

#### Hlavná obrazovka

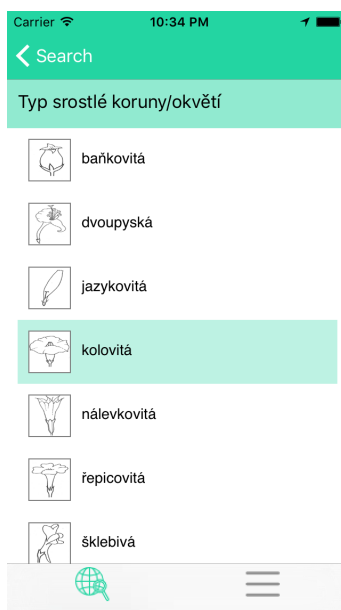
Hlavná obrazovka obsahuje zoznam použitých kritérií a zoznam výsledkov vyhľadávania. Oba zoznamy boli implementované ako tabuľka pomocou UITableView. Tabuľka výsledkov zobrazuje jednotlivé výsledky zoradené podľa váhy, ktorá je priamo úmerná pravdepodobnosti, že sa jedná o danú rastlinu. Bunka tabuľky obsahuje obrázok rastliny a názov. Hlavná obrazovka je znázornená na obrázku 11.



Obrázok 11: hlavná obrazovka

### Obrazovka znakov

Obrazovka znakov je implementovaná ako UITableView. Jednotlivé znaky sú usporiadané podľa vypočítanej entropie, čo má za následok rýchlejšiu identifikáciu rastliny. Bunky tabuľky sú expandovateľné na gesto tapnutia. Pri označení daného znaku sa vyfarbí na zeleno, aby bolo jasne vidieť, ktorý znak sa používa ako atribút pri vyhľadávaní. Zároveň môže byť označený najviac jeden znak z daného typu. Obrazovka znakov je znázornená na obrázku 12.



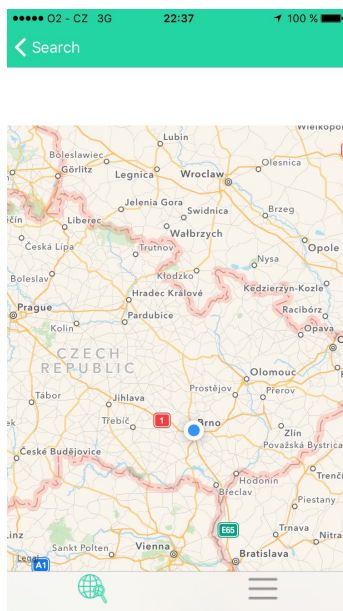
Obrázok 12: obrazovka znakov

### Obrazovka mapy

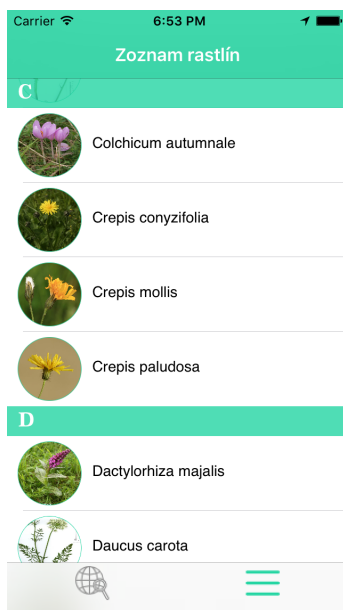
Mapa je instanciou triedy MKMapView. Mapa sa dá približovať pomocou gesta. Tapnutím sa zmení atribút polohy vo vyhľadávaní na súradnice zodpovedajúce miestu tapnutia. Obrazovka mapy je znázornená na obrázku 13.

### Obrazovka zoznam rastlín

Obrazovka obsahuje tabuľku všetkých rastlín uložených v databáze. Zoznam je abecedne uložený a má formu zoznamu kontaktov pre iOS, kde jednotlivé sekcie obsahujú hlavičku so začiatočným písmenom prvku, podľa ktorého sú usporiadané. Jednotlivé bunky v zozname obsahujú obrázok a názov rastliny. Ak nie je obrázok k dispozícii, zobrazí sa predvolený obrázok. Obrazovka zoznamu je znázornená na obrázku 14.



Obrázok 13: obrazovka mapy



Obrázok 14: obrazovka zoznamu rastlín

### 6.3 Implementácia zberu dát pomocou spätnej väzby

Počas procesu identifikácie sa ukladajú znaky, ktoré užívateľ označil. Po úspešnej identifikácii môže užívateľ označiť ktorá z identifikovaných rastlín je tá, ktorú hľadal. Užívateľom označené znaky rastliny a správna odpoveď sa prevedú do JSON tvaru a pošlú na server, kde optimalizácia váh na základe presnosti určovania znakov. JSON

má nasledujúcu štruktúru:

```
{
  "plant": "25",
  "traits": {
    "altitude": "2",
    "compound_leaf": "4",
    "corolla": "1",
    "division": "4",
    "flower_color": "5",
    "simple_leaf": "9"
  }
}
```

Pomocou týchto dát sa vyhodnotí presnosť určovania jednotlivých znakov užívateľmi. Úspešnosti určovania znakov sa použijú na spresnenie identifikácie rastlín

## 7 Diskusia

Dáta obsahujúce znaky rastlín získané z webu Botanickagalerie.cz obsahovali veľke množstvo duplikátov (desiatky tisícov), čo výrazne spomaľovalo SQL dotazy nad dátami a tým aj proces vyhľadávania. Tieto duplikáty bolo treba odstrániť pre plynulý chod aplikácie.

Rozmiestnenia znakov teda aj hodnota informačná entropia sa pri jednotlivých typoch znakov značne líši, čo urýchljuje identifikáciu.

### 7.1 Ďalší vývoj a vylepšenia aplikácie

#### Rozširovanie databázy

Momentálne obsahuje databáza cez 4000 druhov českých rastlín. Znaky sú však obsiahnuté len pre lúčne rastliny, čo je niečo menej ako 200 druhov. Postupne budú pridávané znaky aj ku ostatným druhom rastlín. Po pridaní znakov pre všetky rastliny v databázi budú pridávané ďalšie rastliny, vrátane rastlín nevyskytujúcich sa v Českej republike.

#### Pridávanie kritérií vyhľadávania

Do aplikácie bude pridaná možnosť spravenia fotografie rastliny a identifikácia danej rastliny na základe tejto fotografie. Identifikácia prebehne za pomoci služby Flower Checker.

#### Získavanie finančných zdrojov

Správa a postupný vývoj aplikácie bude vyžadovať aj finančné prostriedky. Na ich získavanie bude do aplikácie pridaná možnosť zakúpenia rozšírenej databázy.

#### Lokalizácia

Aplikácia bude preložená najskôr do anglického jazyka, neskôr aj iných jazykov. Na preklad budú použité finančné zdroje získané nákupom užívateľov v aplikácii.



## 8 Záver

Zadanie bakalárskej práce bolo splnené. Výstupom je aplikácia na určovanie rastlín fungujúca ako botanický kľúč. V samotnej práci je popísaná teória a technológie potrebné pre splnenie zadania, neskôr návrh a implementácia aplikácie.

Ďalším krokom bude distribúcia aplikácie, na ktorú bude použitá distribučná platforma AppStore, ktorá však môže prebehnúť až po získaní súhlasu na publikáciu niektorých dát od autorov. Počas prevádzky bude aplikácia zbierať dáta od užívateľov. Tieto dáta budú použité v ďalších verziách aplikácie na vylepšenie vyhľadávania. Aplikácia bude taktiež naďalej vyvíjaná a vylepšovaná.

## 9 Literatúra

APPLE INC. *iOS Human Interface Guidness* San Jose: iBooks, 2015 [cit. 2016-05-18]  
Dostupné z: <https://itun.es/cz/pyfv0.l>. str. 39.

APPLE DEVELOPER *Core Data Programming Guide: What Is Core Data?* In: [online]. 2016 [cit. 2016-05-21]. Dostupné z:  
<https://developer.apple.com/library/watchos/documentation/Cocoa/Conceptual/CoreData/index.html>.

APPLE DEVELOPER *Getting the User's Location* In: [online]. 2016 [cit. 2016-05-18].  
Dostupné z:  
<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/LocationAwarenessPG/CoreLocation/CoreLocation.html>.

DEAN GERAUX *iOS Data Storage: Core Data vs. SQLite* In: [online]. 2016 [cit. 2016-05-18]. Dostupné z: <http://www.drdoobs.com/mobile/ios-data-storage-core-data-vs-sqlite/240168843?pgno=2>.

FAUSTUS L., POLÍVKA L. *Botanický klíč : klíč k určování 1000 nejdůležitějších cévnatých rostlin 2. vyd* Praha: SPN 1984.  
uuid:38043900-f9d4-11e2-9923-005056827e52.

CHARLES M. GRINSTEAD, J. LAURIE SNELL *Grinstead and Snell's introduction to probability*. Florida: Orange Grove Texts Plus, 2006, s. 405-406  
ISBN 9781616100469.

KHAN ACADEMY *Information entropy | Modern information theory* In: [online]. 2016 [cit. 2016-05-18]. Dostupné z: <https://www.khanacademy.org/computing/computer-science/informationtheory/moderninfotheory/v/information-entropy>.

KOCHAN, STEPHEN G. *Programming in Objective-C. Sixth edition*. Pearson Education, 2013, ISBN 0321967607, s. 140.

KOZÁK J. *Aplikácia k určovaniu rastlín pre platformu iOS* Brno: Bakalárska práca, Mendlová univerzita, 2014.

STAN SCHROEDER *Watch Steve Jobs' announcement of the first iPhone exactly 8 years later* In: [online]. 2016 [cit. 2016-05-21]. Dostupné z: <http://mashable.com/2015/01/09/iphone-steve-jobs/XGM8Hr8N.gqN>.

NAHAVANDIPOOR V. *Programming in Objective-C. Sixth edition*. O'Reilly Media, Inc., 2014,  
ISBN 9781449372422 .

SHANNON C. E. *A mathematical Theory of Communication* The Bell System Technical Journal, vol. 27, 1948  
ISBN 978-80-251-2820-6 , str. 379–432.

SNYDER J. P. *Map projection - a working manual*. Washington: U.S. Government printing office, 1987, ISBN 87-600250, s. 37.

ŠTEFÁNIK, J. – RUSKO, M. – POVAŽANEC, D. *Frekvencia slov, grafém, hlások a ďalších elementov slovenského jazyka* Bratislava: Jazykovedný časopis, 50, 1999, s. 88–90.