



Gymnázium, Praha 6, Arabská

předmět Programování, vyučující Jan Lána

# **Univerzální mobilní aplikace nejen pro učení se na poznávačky**

## **Ročníkový projekt**

**Timotej Adamec, Jakub Kučera, Lucián Kučera, Vojtěch Sobotka**

## Prohlášení

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská<sup>14</sup> oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne 2. 3. 2020 .....

Timotej Adamec, Jakub Kučera, Lucián Kučera, Vojtěch Sobotka

## **Anotace**

Tato práce se zabývá aplikací sloužící k procvičování poznávání obrázků a případné klasifikace objektu na obrázku. Cílem bylo vytvořit aplikaci na platformu Android, ve které uživatel vytvoří seznam (dále jen poznávačka) - jednoduše vypíše věci, ke kterým chce vygenerovat obrázek a případnou klasifikaci. Aplikace za něj tyto věci následně vygeneruje a uživatel si údaje zkontroluje, popř. upraví a poznávačku uloží. Tato poznávačka se dá jednoduše uploadovat na databázi a uživatelé, kteří se budou chtít učit ze stejné poznávačky si tuto poznávačku jednoduše stáhnou dvěma kliknutími. Aplikace bude sloužit širokému spektru uživatelů, nejspíš převážně studentům a žákům jako školní pomůcka a to od základních škol až po vysoké školy. Aplikace má široké spektrum využití.

## **Annotation**

This work goes about an application used for practicing cognition of images and possible classification of an object in an image. The goal was to create an application in which the user creates a list - simply lists the things for which he wants to generate an image and any classification. The application then generates these things for him and the user checks the data, eventually edits and saves the list. This list can be easily uploaded to the database and users who want to learn from the same list can easily download the list with two clicks. The application will serve a wide range of users, probably mostly students and pupils as school aid, from primary schools to universities. The application has a wide range of applications.

## Zadání

Mobilní aplikace sloužící k procvičování poznávání nejen živočichů, také známo jako "poznávačka". Uživatel bude moci vytvořit svůj vlastní seznam rostlin nebo živočichů a používat jej k procvičování nebo bude možné, aby si stáhl seznam, který již někdo vytvořil. Uživatel také bude moci sdílet seznamy, které vytvořil.

Z hlavního menu aplikace, bude moci uživatel přejít do tvorby seznamu, kde zadá do textového pole název rostliny nebo živočicha a zobrazí se mu obrázky, ze kterých si pak jeden vybere (popřípadě se zobrazí i řád a kmen) a ten přidá do seznamu. Až bude v seznamu vše, co by do něj chtěl umístit, bude moci seznam uložit na svém zařízení a případně jej sdílet.

Testování bude možné vybráním testovací funkce z hlavního menu. Uživatel si bude moci vybrat seznam, ze kterého bude zkoušen. Testování bude probíhat tak, že se uživateli zobrazí obrázek a uživatel stiskne tlačítko (nebo se dotkne obrazovky), poté co bude vědět, zda ten obrázek poznává. Uživatel si svůj odhad zkontroluje s informacemi, které se mu zobrazí na obrazovce a bude si moci vybrat ze dvou tlačítek: „jsem si jistý“ a „nejsem si jistý“ (pravděpodobně se tato tlačítka takto jmenovat nebudou). Když si je jistý obrázek se zařadí mezi poznané obrázky a nebude z něj vícekrát zkoušen, dokud znovu nezahájí testování. Pokud stiskne tlačítko nejsem si jistý obrázek se zařadí do mezi nepoznané obrázky a bude mu zobrazen ještě jednou.

Sdílení seznamu bude fungovat tak, že se seznam nahraje na server i s obrázky a bude vygenerován kód pomocí, kterého bude moci uživatel seznam sdílet. Seznam bude možné stáhnout zadáním kódu v aplikaci a pokud seznam s patřičným kódem bude existovat, tak se stáhne na uživatelovo zařízení.

Aplikace bude běžet na systému Android a jako vzdálené úložiště a databázi použijeme Firebase.

<b>Obsah</b>	
<b>Úvod</b>	<b>5</b>
<b>Použité nástroje</b>	<b>5</b>
Firebase	5
Authentication	5
Cloud Firestore	5
jsoup	6
Google Custom Search	6
Gson	6
Picasso	6
<b>Struktura aplikace</b>	<b>6</b>
<b>Třídy</b>	<b>8</b>
<b>Vytváření poznávačky</b>	<b>9</b>
Generace poznávačky	11
<b>Ukládání seznamu</b>	<b>13</b>
<b>Správa poznávaček</b>	<b>13</b>
<b>Procvičování</b>	<b>14</b>
<b>Testování</b>	<b>16</b>
<b>Budoucnost aplikace</b>	<b>16</b>
Zkoušení ve školách	16
Jazykové varianty pro aplikaci a generaci	16
Další vylepšení	17
<b>Závěr</b>	<b>17</b>
<b>Seznam obrázků a tabulek</b>	<b>18</b>
<b>Zdroje</b>	<b>19</b>

## Úvod

Myšlenka na vytvoření aplikace na procvičování poznávání obrázků vzešla ve chvíli, když jsme na hodinách biologie dostali už poněkolkáté poznávačku se zástupci různých členění. Počet zástupců na jeden test se lišil, ale jakkoli byla poznávačka rozsáhlá, učení se na něj bylo vždy velmi obtížné a každý student se na učení připravoval individuálně, takže tyto obtíže absolvoval kdokoli, kdo se na test připravoval. Jelikož to bylo takto náročné, některé studenty to odradilo natolik, že si nechali dát za pět, než aby si tím zdoluhavým a namáhavým procesem projít. Tehdy mě napadla myšlenka, že určitě existuje způsob, který by přípravu na tento typ testů zjednodušil. A tak jsem se porozhlédl po aplikacích apod., ale nic jsem nenašel. Kdybychom to vzali po matematické stránce, jak moc by tato aplikace mohla pomoci? V případě, že bychom dopad aplikace minimalizovali, výsledek je stále obdivuhodný. Délka přípravy učebního materiálu se zrychlí čtyřnásobně (minimálně). Tuto práci nadále nebudou muset dělat znovu i spolužáci, tudíž místo dvaceti příprav se uskuteční jen jedna. Průběh rychlosti učení se zdvojnásobí, jelikož je automatizovaný. Zbylých pár žáků, kteří by se původně vůbec nepřipravovali, se připraví taktéž. Tudíž, efektivita přípravy na test by se v rámci třídy zvýšila minimálně 160 krát, nemluvě o následných výsledcích v testu. Takže jsem se rozhodl tuto aplikaci realizovat a dali jsme dohromady tým.

## POUŽITÉ NÁSTROJE

### 1.1 Firebase

Firebase poskytuje široký sortiment nástrojů pro vývojáře od vývoje až po kvalitu nebo analytiku. Firebase je zároveň škálovatelný a rozvíjí ho společnost Google, takže máme záruku stability a podpory. Z nabídky nástrojů jsme zatím využili dvou: Authentication a Cloud Firestore.

#### 1.1.1 Authentication

Firebase nabízí většinu populárních přihlašovacích metod, z nichž další budou přidány do naší aplikace. Na Firebase Authentication můžeme jako vývojáři manuálně spravovat uživatele a jejich účty.

#### 1.1.2 Cloud Firestore

Cloud Firestore je novější varianta pro Firebase databáze využívající NoSQL technologie. V případě rozvoje a přechodu na web je možné využívat stejné škálovací databáze. Umožňuje jednoduchou manipulaci s daty v reálném čase.

## 1.2 jsoup

Jsoup je knihovna pro jazyk Java sloužící k programatickému přístupu k internetovým stránkám a následné extrakci dat ze stránek.

## 1.3 Google Custom Search

Google Custom Search nabízí využití služeb vyhledávání Google Search. V našem případě jsme využili Google Custom Search Site Restricted JSON API. Vyhledávání v našem případě slouží k nacházení správných výsledků z uživatelského vstupu. Pro případ nenalezeného obrázku z Wikipedie je implementován druhý vyhledávací způsob bez restrikce na Wikipedii, který využívá Google Custom Search JSON API, ten je však omezený na 10 000 hledání za den.

## 1.4 Gson

Gson je knihovna s účelem jednoduchého konvertování objektů do formátu JSON. Také dokáže přeměnit JSON string na objekt.

## 1.5 Picasso

Picasso je mocný nástroj na zobrazování obrázku pomocí internetové adresy, změnu jejich velikosti. Může také obrázky zobrazit ze souborů a nebo z jejich číselných kódů.

### STRUKTURA APLIKACE

Každá aplikace obsahuje takzvané aktivity, které dále mohou obsahovat fragmenty. Každá aktivita slouží specifickému účelu a každý fragment slouží užšímu využití. V našem případě aplikace obsahuje následující aktivity a fragmenty:

- PracticeActivity
  - V této aktivitě si uživatel může zvolit, jestli si bude procvičovat vědomosti z celé poznávačky nebo z množiny zástupců, které se ještě nenaučil.
  - PracticeActivity2
    - Zde si uživatel procvičuje své znalosti. Aktivita reaguje na kliknutí na jednotlivé prvky a obnovuje scénu podle předešlé aktivity uživatele.
- ListsActivity
  - Tato aktivita zahrnuje manipulaci s veškerými poznávkami a také tlačítko odkazující na CreateListActivity, kde si uživatel vytváří poznávku.

- MyListsFragment
  - V tomto fragmentu se uživateli zobrazují lokálně uložené poznáváčky, jak ty vytvořené ostatními uživateli a následně staženými uživatelem nebo poznáváčky vytvořené uživatelem samotným. Fragment reaguje na kliknutí na tlačítka s různými funkcemi, která operují s konkrétními poznávkami, jako např. mazání, sdílení nebo procvičování.
- SharedListsFragment
  - V druhém fragmentu ListsActivity si uživatel může procházet nebo vyhledat poznáváčky vytvořené a uploadované ostatními uživateli a následně si je stáhnout a procvičovat si je.
- CreateListActivity
  - Prostředí pro vytváření poznáváčky samotné. Obsahuje pole pro název a výpis věcí, které se chce uživatel naučit. Dále také účelový jazyk pro poznáváčku a možnost generace klasifikace (viz PopActivity). Funguje zde také asynchronní funkce pro generaci upravitelného seznamu.
- PopActivity
  - Tato aktivita se zobrazí ve chvíli, kdy uživatel zaškrtně generaci klasifikace. Aktivita zjišťuje, které klasifikace si přeje uživatel vygenerovat.
- TestActivity
  - Toto je zatím aktivita bez využití. Aktivita bude fungovat jako prostředí pro zkoušení žáků ve školách.
- AccountActivity
  - Aktivita obsahující informace zobrazující se uživateli o uživateli a možnost odhlášení uživatele.
  - Tlačítko “Change language” otevírá alert dialog, který umožňuje změnit jazyk aplikace. Jazyk aplikace je měněn pomocí String setů a knihovny Locale. Kód pro změnu String setu převzat z Youtube(Ativ Pervaiz,2018).
- AuthenticationActivity
  - Obsahuje metody pro manipulaci s uživatelským účtem pro Firebase a Google přihlášení.



- ExamsActivity
  - Obsahuje uživatelské testy. V této aktivitě si uživatel může vybrat, jestli se podívá na výsledky konkrétního testu, smaže test, spustí a nebo zastaví test. Také uživatel vidí obrázek na ukázkou z daného testu.
- TestUserActivity
  - Zde uživatel vyplní a odevzdá test. Může se vracet tam a zpátky v testu. Má zde obrázek zástupce.
- UserResultActivity
  - Zde uživatel vidí výsledky jednotlivých testů. Uživateli se zobrazí výsledek a jméno vyplňujícího, každý uživatel má zde zobrazeno jenom první odevzdání testu, pokud něco odevzdal.

## TŘÍDY

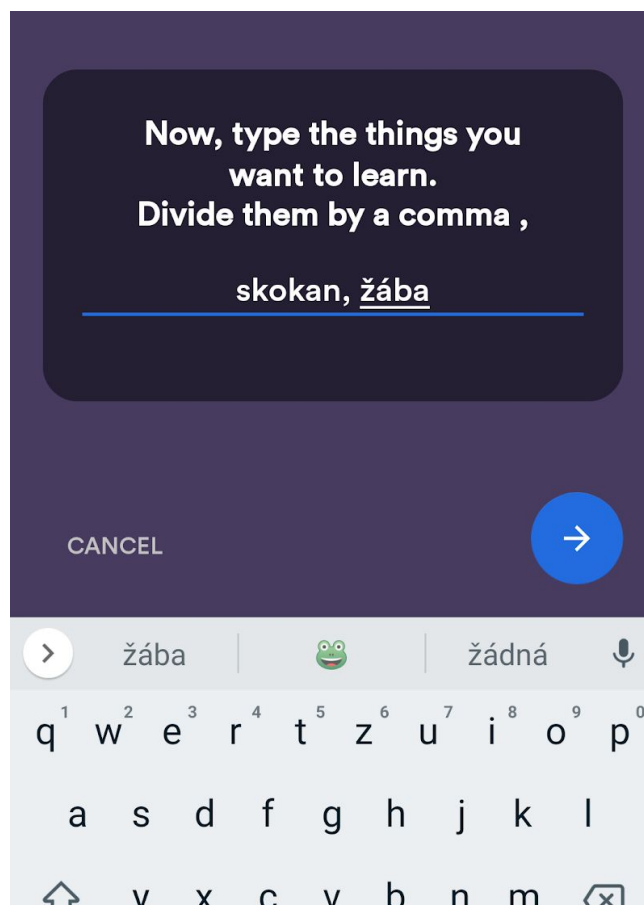
- PoznavackaInfo
  - Tato třída uchovává informace o poznávkách potřebné k jejich smazání, sdílení, či procvičování. Obsahuje následující parametry: název poznávky, ID poznávky, název a ID autora, který poznávku vytvořil a umístění a url obrázků, který je vykreslován v My lists fragmentu a Share list fragmentu.
- Zastupce
  - Tato třída uchovává informace o jednotlivých zástupcích v poznávce nutné k jejich procvičování, tím je počet parametrů, pole s parametry, obrázek a url obrázku.
- PoznavackaDbObject
  - Reprezentuje poznávku v databázi má parametry: Název poznávky, ID poznávky, autorovo ID, obsah poznávky, url titulního obrázku, cestu titulního obrázku na zařízení a autorovo jméno.
- PreviewPoznavacka
  - Reprezentuje databázi na zařízení, v podobě nutných parametrů jako je id poznávky v databázi a název autora. Všechny objekty toho typu se zobrazí v

recyclerview a umožní smazat poznávačku z firestoru, nebo ji stáhnout na zařízení.

- TestObject
  - Reprezentuje test, který uživatel vytvořil.
- ActiveTestObejct
  - Reprezentuje aktivní test, do kterého se může každý připojit
- ResultObject
  - Reprezentuje testový výsledek uživatel. Má dvě části jedna je v databázi a druhá je zobrazena v zařízení.
- AnswerObejct
  - Reprezentuje jednotlivé otázky testu.
- Google search objects
  - Složka google\_search\_objects obsahuje řadu tříd, do kterých se ukládají JSON objekty stažené pomocí Google Custom Search. Google Custom Search nabízí řadu informací, avšak pro naše účely stačily dvě URL - na první odkaz na Wikipedii a na obrázek.
- StorageManagerClass
  - Jak vypovídá název, tak tato třída se stará o ukládání objektů a obrázků na zařízení a také o jejich čtení. Kód čtení a ukládání obrázku byl založen na kódu z webu Stack Overflow (Takur, 2016).

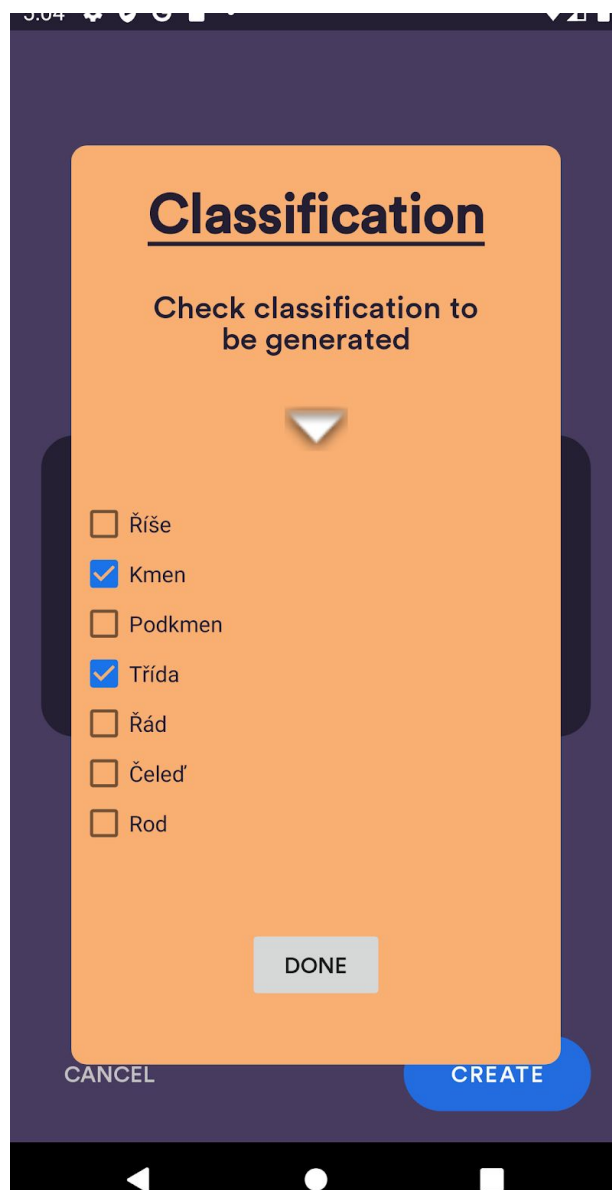
## **VYTVÁŘENÍ POZNÁVAČKY**

Při vytváření poznávačky nás od uživatele zajímají následující parametry: název poznávačky; výpis věcí, které se chce naučit; jazyk, ve kterém chce poznávačku vytvořit; pokud chce vygenerovat i klasifikaci, popř. kterou konkrétně.



Obr. 1: Snímek obrazovky vytváření poznávky

Pole “Title”, tedy název poznávky, je přímočarý, kromě zobrazovací funkce nemá další využití. V poli “The representatives” uživatel wpisuje věci, které oddělí čárkou. Při volbě jazyka má uživatel na výběr z několika možností. Momentálně je dostupná angličtina (která funguje i pro vyhledávání pro latinské názvy) a čeština. Jazyk je tu pro zúžení možností pro generaci nesprávných údajů. Posledně je tu možnost, jestli chce uživatel vygenerovat klasifikaci. Při kliknutí na generaci klasifikace se uživateli zobrazí okno, kde se načte všechna klasifikace a uživatel pouze zaškrtně, které klasifikace chce vygenerovat.



Obr. 2: Snímek obrazovky generace a volby klasifikace

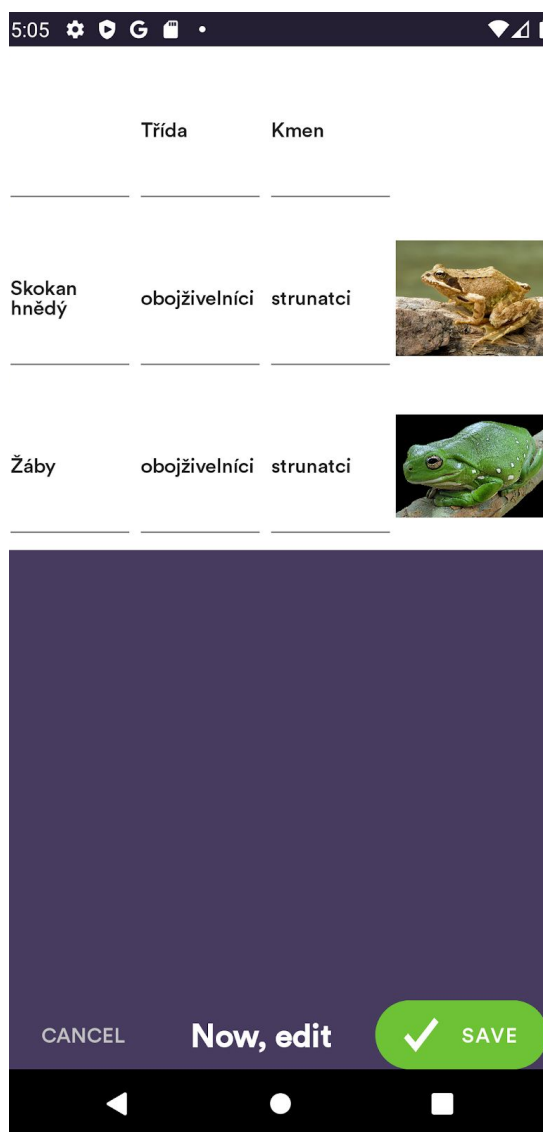
## 1.6 Generace poznávačky

Základem generace je Google Custom Search Site Restricted JSON API, který prohledává wikipedii pro správný článek a obrázek. Druhým stavebním kamenem je samotná stránka Wikipedie. Nutno podotknout, že jakýkoli scraping je povolený a hospodaření s daty o pár slovech z Wikipedie nevyžaduje zvláštních oprávnění.

Pro každou věc je spuštěn proces, který vezme jazyk a věc jako parametry pro vyhledávání Google Search API. API vrátí JSON objekt, ze kterého naše aplikace vytáhne URL jak obrázku, tak stránky Wikipedie. Obrázek se stáhne a následně zobrazí. V případě, že by uživatel udělal překlep a nenašli se žádné výsledky, proběhne přesměrování na správnou stránku. Google Search totiž může obsahovat i String “correctedQuery”, kde navrhuje opravený vstup, který jsem použil pro přesměrování na nové, opravené vyhledávání bez toho,

aniž by se uživatel o cokoli staral. Pro klasifikaci je nutné pokračovat dále na zdrojovou stránku Wikipedie. Zde je nutné najít patřičnou tabulku s klasifikací. Jelikož kódová struktura Wikipedie není konzistentní, bylo potřeba udělat specifický postup pro případ zvlášť. Po zkoušení různých vyhledávání jsem dospěl k výsledku, že stránky se stávají strukturně konzistentními až v rámci jazyka. To bylo dalším důvodem, proč se uživatele zeptat na účelový jazyk poznávačky. Generace možností se nyní liší pouze v tom, že hledá všechny možné klíče v tabulce a dá na výběr uživateli. V tabulce hledáme tyto klíče klasifikace, předem stanovené uživatelem. Pokud se najde shodný klíč, našli jsme hodnotu pro patřičnou klasifikaci. Abychom opravili možné překlapy uživatele. Tento proces se opakuje pro každou uživatelem zadanou věc.

Uživatel může následně upravit nebo dopsat jakákoli data. Pokud je uživatel spokojen, uloží poznávku pomocí tlačítka “SAVE”.



Obr. 3: Snímek obrazovky vygenerovaného seznamu poznávačky

## UKLÁDÁNÍ SEZNAMU

Po kliknutí na tlačítko “SAVE” po vytvoření poznávačky se vytvoří v internal storage zařízení nová složka s názvem rovnému ID vytvořené poznávačky. Do této složky se nejprve uloží všechny obrázky, protože seznam zástupců nejde převést do JSON-u, pokud obsahuje obrázky. Obrázky jsou pojmenovávány podle názvu, toho co zastupují, a ukládány ve formátu PNG. Poté je seznam zástupců převeden na string ve formátu JSON a uložen ve složce s obrázky jako textový soubor. Nakonec se do seznamu poznávaček přidá vytvořená poznávačka a je přepsán soubor obsahující poznávačky, aby obsahoval i tu nově vytvořenou.

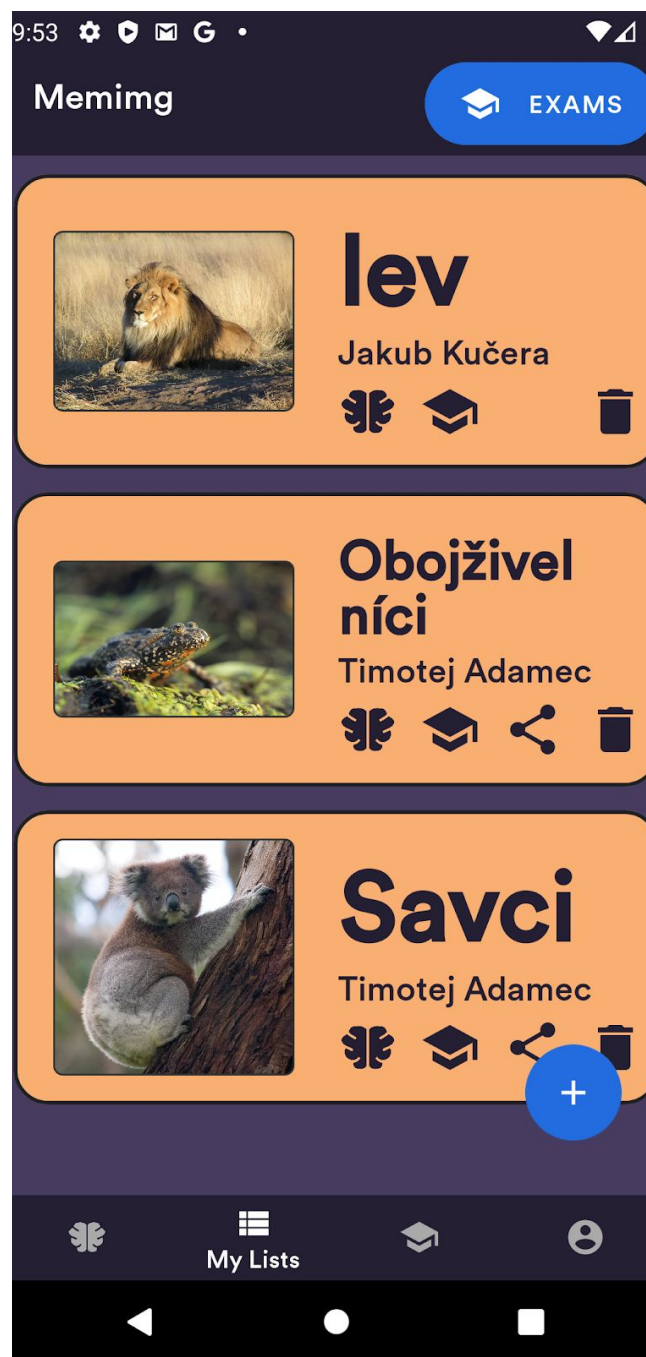
## SPRÁVA POZNÁVAČEK

V My lists fragmentu může uživatel spravovat všechny poznávačky, které jsou uloženy na jeho zařízení. Při vytvoření fragmentu se do recycler view v již zmíněném fragmentu načte seznam poznávaček uložených na tomto zařízení. Seznam poznávaček v internal storage zařízení v textovém souboru “poznavačka.txt”, ve kterém je pole PoznavackaInfo objektů uložených jako JSON, se po přečtení převede na objekt pomocí knihovny Gson. Načtené poznávačky jsou poté zobrazeny v recycler view.

V tomto okně může uživatel poznávačky označit k procvičení tak, že při otevření procvičovacího fragmentu se načte právě ta označená poznávačka. Podobnou funkci má i tlačítko, které uživatele kromě označení poznávačky uživatele přenese přímo do Practice fragmentu. Další tlačítko slouží ke sdílení poznávačky. Poznávačka se sdílí pomocí firebase-firestore API, který umožňuje poznávačku sdílet v podobě objektu. Objekt PoznavackaDbObjekt převezme obsah poznávačky, název, uuid z PoznavackaInfo a zbytek se generuje podle uživatele aby měl určitou možnost vlastnictví.

Další tlačítko je k smazání poznávačky. Po kliknutí na toto tlačítko se uživateli zobrazí okno, které se zeptá, jestli opravdu chce poznávačku spojenou s tímto tlačítkem smazat. Když odpoví ano, tak bude poznávačka spolu se všemi obrázky, které k ní patří odstraněna. Mimo jiné se u každé poznávačky zobrazuje obrázek prvního zástupce.

Poslední tlačítko, které se vyskytuje na horním panelu, slouží k přechodu do aktivity na správu testů.



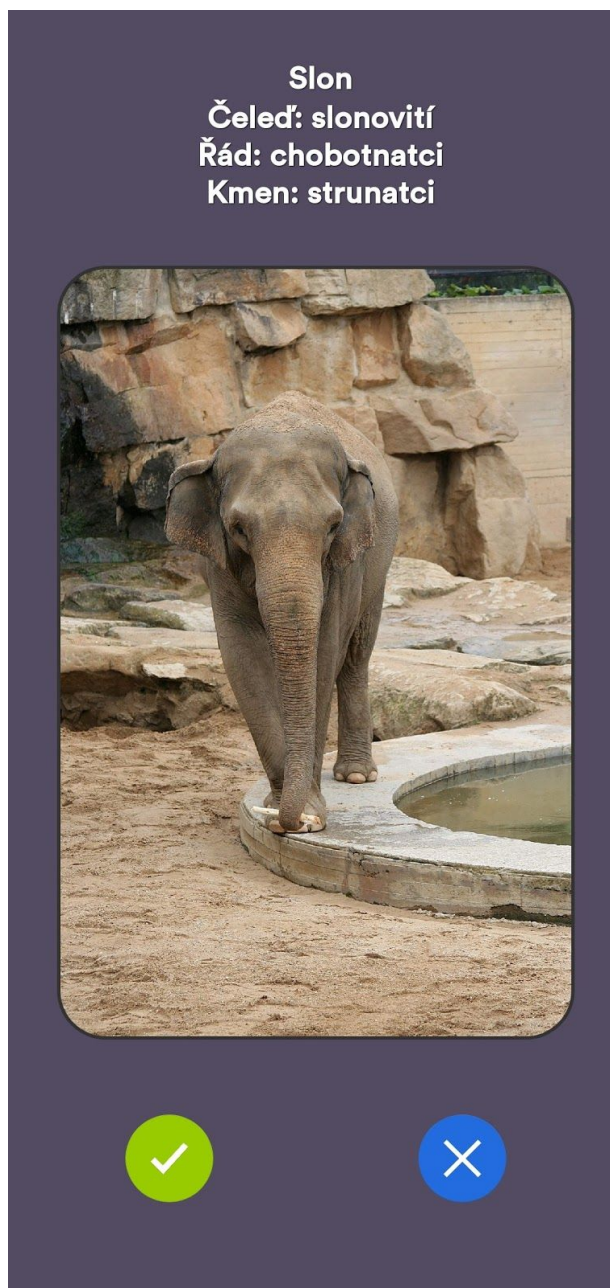
Obr. 4: Snímek obrazovky správy poznávaček

## PROCVIČOVÁNÍ

Poté, co si uživatel, vybere poznávačku, kterou si chce procvičovat, aplikace ho přenesse do procvičovacího prostředí (PracticeActivity). Na základě průzkumu na nejoblíbenější metodu procvičování jsem zvolil jednu takovou metodu a implementoval ji. Aplikace zobrazuje uživateli vždy jen jednu položku. Nejdříve se uživateli zobrazí pouze obrázek a nic více. Uživatel si v tuto chvíli ujasňuje, co vidí a do jaké klasifikace se řadí. Když má rozmyšleno, klikne a aplikace mu zobrazí název, klasifikace a zároveň dvě tlačítka: fajfka - uživatel věděl;

křížek - uživatel nevěděl. V případě, že uživatel věděl, zástupce v poznávačce se mu už nezobrazí. V opačném případě zůstává zástupce mezi ostatními nenaučenými zástupci. Uživateli se dokola zobrazují nenaučení zástupci v náhodném pořadí.

Když uživatel ukončí poznávačku předčasně, tak si aplikace jeho postup zapamatuje. Zástupci, u kterých uživatel neodpověděl, že je poznává, se uloží na zařízení do textového souboru ve formátu JSON stringu. Při opětovném procvičování stejné poznávačky se pole zástupců načte ze zařízení a uživatel si může vybrat, zda chce být vyzkoušen ze všech zástupců, nebo jen těch, které se nestihl naučit před ukončením procvičování.



Obr. 5: Snímek obrazovky procvičování poznávačky



## TESTOVÁNÍ

V naší aplikaci může uživatel vytvořit test pro ostatní uživatele. Uživatel si vybere z poznávek uložených na jeho zařízení a přidá ji do složky testy. V složce testy si může uživatel vybrat kdy zahájí test, kdy ho ukončí, smazat nebo se dostat do výsledku testu těch, co vyplnili tento test.

Testy se ukládají na firebase. Zahájením testu se vygeneruje jedinečný 9 místný kód a test se přidá do firebase. Test se uloží v collectionu active tests. Když zadáte kód a test je aktivní, dostanete se do testu. Při deaktivaci testu se dokument v collectionu active test smaže a přestává být dostupný. Smazání testu proběhne tak, že se nejdříve smaže collection výsledků daného testu a poté se vymaže složka testu v uživatelově test collection. Test vyplňujete formou textu, každé správně vyplněné políčko se rovná jednomu bodu. Každý uživatel může odevzdat test pouze jednou.

Při tvorbě testování nejnáročnější bylo vymyslet konstrukci databáze, aby všechno cesty zůstaly jedinečné a nenastaly duplikáty. Každý test při vytvoření má vždy jedinečné id a je uložený v collectionu o názvu id uživatele, který je také jedinečný. Výsledky všech testu mají zvláštní collection s názvem id testu. Id testu je jedinečné v celé databázi, díky firebase.

## BUDOUCNOST APLIKACE

### 1.7 Zkoušení ve školách

Jedním z prvotních návrhů na funkce aplikace, byla také možnost využití aplikace pro učitele jako pomůcka při zkoušení studentů z poznávek. Aplikace má už hotové testovací oddělení, takže je možné aplikaci vyzkoušet ve školním prostředí. Myšlenka je taková, že uživatelé (žáci) dostanou od učitele PIN, který je přenesen do testovacího prostředí a žáci vyplní test. Výsledky se následně zobrazí v učitelově aplikaci. S drobnými úpravami by se mohly výsledky také odeslat učiteli na e-mail.

### 1.8 Jazykové varianty pro aplikaci a generaci

Dalším z našich primárních cílů je expandovat použitelnost aplikace do více zemí světa a zpřístupnit tak možnost generace poznávek lidem mluvícím jiným jazykem. Ve finální verzi k odevzdání je už zabudována podpora více jazyků, a je zde možné vybírat z češtiny a angličtiny, pro přidání dalšího jazyka je tedy pouze nutné zajistit kvalitní překlad.

## **1.9 Další vylepšení**

Dalšími možnými vylepšeními je přidání metod na procvičování, zpětnou vazbu uživateli na to, jak si vede při trénování na poznávačku nebo drobnějšími vylepšeními, jako úprava seznamu poté, co je vytvořený a tak podobně.

### **ZÁVĚR**

S výsledkem práce jsme spokojeni. I když aplikace ještě není plnohodnotná, je funkční a dosáhli jsme cílů, které jsme si stanovili. Chtěli jsme, aby aplikace byla univerzální a uživatelsky nenáročná. S nynější aplikací je spektrum využití rozsáhlé. Aplikace je vhodná pro děti prvního stupně základních škol až po dospělé lidi. Původním cílem bylo edukativní funkčnost, jako samoučení se, pomůcka do výuky pro učitele, apod. Aplikace ale má i zábavní funkci. Například, skupina hokejových fanoušků si udělá seznam českých hráčů v NHL a budou hádat, v jakém týmu tito hráči momentálně hrají. Seznam si pak mohou (pokud byl tvůrcem uploadován) stáhnout a zkoušet se sami. Na aplikaci hodláme dále pokračovat a zdokonalovat její funkcionalitu, přehlednost a ovladatelnost.

## SEZNAM OBRÁZKŮ A TABULEK

Obr. 1: Snímek obrazovky vytváření poznávačky	10
Obr. 2: Snímek obrazovky generace a volby klasifikace	11
Obr. 3: Snímek obrazovky vygenerovaného seznamu poznávačky	12
Obr. 4: Snímek obrazovky správy poznávaček	14
Obr. 5: Snímek obrazovky procvičování poznávačky	15

## ZDROJE

- Takur, B. (5. Říjen 2016). *Saving and Reading Bitmaps/Images from Internal memory in Android*. Načteno z Stack Overflow: <https://stackoverflow.com/questions/17674634/saving-and-reading-bitmaps-images-from-internal-memory-in-android>
- Atif Pervaiz. “How to make our app to support multiple languages? - Android Studio” YouTube video, 5. Únor 2018 . <https://www.youtube.com/watch?v=zILw5eV9QBQ>