

STREDNÁ PRIEMYSELNÁ ŠKOLA ELEKTROTECHNICKÁ
PREŠOV

IV.SA – 20, 22

ŠK. R. 2021 – 2022

PČOZ – PET APP - PANDA

NATÁLIA PETROVÁ, MARGARÉTA RENČKOVÁ

Konzultant: Ing. Mária Hedvigová

ANOTÁCIA V SLOVENSKOM JAZYKU

Cieľom tejto práce bolo vytvoriť užitočnú aplikáciu pre majiteľov domácich zvierat s názvom Panda. Táto mobilná aplikácia má slúžiť ako elektronická zdravotná karta s informáciami pre veterinára aj majiteľa domáceho zvierat'a. Používateľ a zároveň majiteľ má jedinečné identifikačné číslo svojho domáceho zvierat'a, vďaka ktorému vie pristupovať k informáciám o zvierati v aplikácii. Veterinár na základe tohto čísla vie registrovať pre zviera očkovania, kontroly a iné vyšetrenia do databázy cez aplikáciu a následne sa tieto údaje zobrazia aj užívateľovi. Okrem toho sa v aplikácii nachádza aj niekoľko užitočných informácií o zvieracích chorobách, či priame kontakty na veterinárov. Aplikácia je robená primárne pre užívateľov s identifikačným číslom zvierat'a ale dá sa používať aj bez neho.

ANOTÁCIA V ANGLICKOM JAZYKU

The aim of this work was to create a useful application for pet owners called Panda. This mobile application is intended to serve as an electronic health card with information for both the veterinarian and the pet owner. The user and the owner also have a unique identification number of their pet, thanks to which he can access the information about the animal in the application. Based on this number, the veterinarian can register vaccinations, checks and other examinations for the animal into the database via the application, and subsequently this data will also be displayed to the user. In addition, the application also contains some useful information about animal diseases or direct contacts to veterinarians. The application is made primarily for users with an animal identification number, but can be used without it.

Čestné vyhlásenie

Vyhlasujeme, že celú prácu s názvom Pet APP - Panda sme vypracovali samostatne, s použitím uvedenej literatúry.

Sme si vedomé zákonných dôsledkov, ak v nej uvedené údaje nie sú pravdivé.

Prešov, 9. máj 2022

.....

vlastnoručný podpis

.....

vlastnoručný podpis

Pod'akovanie

Týmto by sme chceli vyjadriť pod'akovanie konzultantke Ing. Márii Hedvigovej za jej odbornú pomoc pri vytváraní tejto práce.

Taktiež by sme chceli pod'akovať všetkým ľuďom, ktorí nám povedali svoje pripomienky a vylepšenia a pomohli nám ich aplikovať do mobilnej aplikácie.

Obsah

ÚVOD.....	5
1 CIEĽ PRÁCE.....	6
2 METODIKA PRÁCE	7
3 TEORETICKÁ ČASŤ - ÚVOD DO PROBLEMATIKY	8
3.1 MOBILNÁ APLIKÁCIA	8
3.2 MOBILNÉ ZARIADENIE.....	8
3.3 DATÁBÁZA	8
3.4 VYHLADÁVACÍ JAZYK	9
3.5 ANDROID	9
3.6 IDE.....	10
3.7 ANDROID STUDIO	10
3.8 JAVA.....	10
3.9 XML	11
3.10 JSON.....	11
3.11 FIREBASE	12
4 PRAKTICKÁ ČASŤ	13
4.1 ÚVODNÉ ZOBRAZENIE APLIKÁCIE	13
4.2 NAVIGÁCIA.....	14
4.3 DATABÁZA	15
4.4 ZOBRAZENIE OČKOVANÍ A NÁVŠTEV	16
4.5 ZÁPIS DO DATABÁZY	17
4.6 ZOZNAM VETERINÁROV	18
4.6.1 Zobrazenie veterinárov v zozname	18
4.6.2 Zobrazenie samotného veterinára	19
4.6.3 Zvolenie obľúbeného veterinára	20
5 ČASTI KÓDU	21
5.1 NASTAVENIE OBRÁZKA PRE ZVIERA.....	21
5.2 NASTAVOVANIE ČÍSLA ČIPU	22
5.3 OVEROVANIE PRIPOJENIA.....	23
6 VÝSLEDKY PRÁCE	24
7 ZÁVERY PRÁCE.....	25
ZHRNUTIE	26
RESUMÉ	27
ZOZNAM POUŽITEJ LITERATÚRY	28
ZOZNAM OBRÁZKOV	29

ÚVOD

V posledných rokoch sa vo svete rozbieha trend „domácich miláčikov“ a mnohí si do svojej domácnosti zadovážia psa alebo mačku. Neuvedomujú si však, že okrem potešenia a spoločnosti tieto milé stvorenia prinášajú aj niekoľko povinností a potrebu starostlivosti o nich.

Vďaka veterinárom sa naše domáce zvieratká môžu cítiť dobre. Veterinári majú svoje ambulancie a, podobne ako doktori, svojich „pacientov“ ich informácie a zdravotné karty. Nielen ľudia ale aj zvieratá môžu mať rôzne choroby a komplikácie so zdravím. Každý majiteľ by určite bol rád, keby si mohol pozrieť prehľad o zdravotnom stave svojho miláčika a najmä by sa v ňom nestratil hneď po otvorení aplikácie a aby mu aj laicky rozumel. Návštevy a vyšetrenia u veterinára či predchádzajúce a nasledujúce očkovania, to všetko sú užitočné informácie pre majiteľa domáceho zvieratka. Očkovania sú totiž u zvierat o niečo častejšie ako u ľudí.

Nie je viac potrebné zapisovať si termíny do kalendára či sledovať pravidelne veterinárny preukaz aby sme nič nezmeškali, stačí si otvoriť aplikáciu Panda v mobilnom telefóne a máme všetky informácie o zvieratku na jednom mieste, kdekoľvek sa nachádzame.

V dnešnom svete, kde už množstvo preukazov a papierov prechádza do elektronickej formy je práve vhodné aby do tejto formy prešiel aj preukaz zvierat.

1 CIEĽ PRÁCE

Cieľom tejto práce bolo vytvoriť mobilnú aplikáciu s názvom Panda. Táto aplikácia by mala slúžiť ako pomôcka a zjednodušenie pre majiteľov domácich zvierat, primárne psov, ktoré je nutné očkovať a zabezpečovať im rôznu veterinárnu starostlivosť.

Zaobstarať si domáce zvieraťko je zmena na dlhú dobu a jeho zaobstaraním sa zaväzujete že sa o neho budete starať po celý jeho život. Zákon o veterinárnej starostlivosti hovorí, že vlastník alebo držiteľ zvierat'a je povinný na vlastné náklady zabezpečiť vakcináciu a najmä vakcinácie proti besnote. Očkovať sa však dá aj proti mnohým iným ochoreniam. Všetky tieto údaje sa zapisujú do Veterinárneho preukazu pre spoločenské zvieratá. Aplikácia Panda ponúka tieto, a mnohé ďalšie užitočné informácie v elektronickej podobe.

V aplikácii nájdeme presný prehľad zdravotného stavu domáceho miláčika. Okrem toho tam nájdeme aj nejaké informácie o najčastejších chorobách či informácie o prvej pomoci pre zviera.

2 METODIKA PRÁCE

Mobilnú aplikáciu sme vytvorili použitím softvéru Android Studio (4.1+), s ktorým sme sa stretli a pracovali na hodinách Programovania mobilných aplikácií. Aplikácia je vytvorená pomocou programovacieho jazyka Java, v ktorom programujeme od prvého ročníka.

Dizajn aplikácie bol vytvorený pomocou aplikácie ako je Adobe Photoshop alebo CorelDRAW, kde bola vytvorená väčšina grafických prvkov a ikon, ktoré používame. Tiež bolo vytvorené logo aplikácie resp. ikona reprezentujúca aplikáciu, ktorá je použité na úvodnej obrazovke resp. splash screen.

Grafické zobrazenie aktivít a fragmentov je zabezpečené použitím značkovacieho jazyka XML. Pri navrhovaní dizajnu sme sa snažili o jednotný dizajn aplikácie tak, aby farby aj ikony aplikácie spolu ladili čo najlepšie.

Na vytvorenie databázy sme použili nástroj od spoločnosti Google – Firebase, ktorý umožňuje prevádzkovať realtime database bez poplatku. Databáza uchováva dáta vo formáte JSON. Pre prácu s databázou sú určené knižnice a metódy, ktoré už implementuje samotné Android Studio.

Pri tvorbe tohto projektu sme zväčša používali dokumentácie od Android Developers, Avšak bolo potrebné naštudovať si používanie aj iných nástrojov použitých v projekte. Navštívili sme aj komunity pre vývojárov ako je napríklad Stack Overflow.

Vzhľadom na to, že bol projekt robený v tíme využívali sme verziovací nástroj – Git, cez poskytovateľa hostingu na vývoj softvéru a správu verzií – GitHub, Inc. Zvolili sme používanie tohto nástroja, pretože sme s týmto nástrojom pracovali v 3. ročníku na hodinách projektového manažmentu.

3 TEORETICKÁ ČASŤ - ÚVOD DO PROBLEMATIKY

3.1 MOBILNÁ APLIKÁCIA

Mobilná aplikácia je softvérová aplikácia vytvorená špeciálne pre smartfóny, tablety, emulátory, hodinky a ďalšie mobilné zariadenia, ktorá si vyžaduje inštaláciu, čím sa líši od bežného webu. (Macho, 2020)

Na rozdiel od tých webových využívajú mobilné aplikácie plný potenciál výkonu smartphonu. Väčšinou vďaka tomu fungujú rýchlejšie a pracujú so vstavanými funkciami telefónu, ako je určovanie polohy, push notifikácia, mikrofón či fotoaparát. (Lang, 2020)

3.2 MOBILNÉ ZARIADENIE

Mobilné zariadenie je malý prenosný elektronický bezdrôtový prístroj s vlastným napájaním (bežne lítium iónová batéria) a rôznymi aplikáciami. Často je vybavený dotykovým displejom a/alebo miniatúrnou klávesnicou. (Wikipedia Contributors, 2022)

Väčšina týchto zariadení sa môže pripojiť k internetu a prepojiť s inými zariadeniami, ako sú napríklad automobilové zábavné systémy alebo náhlavné súpravy, cez Wi-Fi, Bluetooth, mobilné siete alebo NFC (Near Field Communication). Mobilné zariadenia môžu používať mobilné operačné systémy (napr. Android, iOS), ktoré umožňujú inštaláciu a spustenie aplikácií tretích strán. (Hmn Wiki Community)

3.3 DATÁBÁZA

Databáza je množina štruktúrovaných dát alebo informácií uložených v počítačovom systéme, takým spôsobom, že počítačový program alebo človek môže použiť dopytovací jazyk (napr. SQL) na získavanie týchto informácií.

V každej databáze existuje opis štruktúry dát a typu dát, ktoré sú v databáze: tento opis sa nazýva logická schéma. Táto schéma popisuje objekty, ktoré sú v databáze a vzťahy medzi nimi. (Wikipedia Contributors, 2021)

3.4 VYHLÁDÁVACÍ JAZYK

Vyhľadávací jazyk alebo dopytovací jazyk (angl. query language) je jazyk na manipuláciu s údajmi (data manipulation language), ktorým môžu používatelia získavať údaje uložené v databáze. Tvorí sa systémom príkazov a presne definovaným významom a pravidlami na ich využívanie v dialógu s počítačom pri vyhľadávaní v uložených databázach. Medzinárodne uznávaným štandardom je vyhľadávací jazyk SQL. (Wikipedia Contributors, 2014)

3.5 ANDROID

Android je rozsiahla open source platforma, ktorá vznikla najmä pre mobilné zariadenia. Zahŕňa v sebe operačný systém (založený na jadre Linux), middleware, používateľské rozhranie a aplikácie. Vyvíja ho konzorcium Open Handset Alliance, ktorého cieľom je progresívny rozvoj mobilných technológií, ktoré budú mať výrazne nižšie náklady na vývoj a distribúciu, a zároveň spotrebiteľom prinesú inovatívne používateľsky prívetivé prostredie. Pri vývoji systému boli brané do úvahy obmedzenia, ktorými disponujú klasické mobilné zariadenia ako výdrž batérie, menšia výkonnosť a málo dostupnej pamäte. Zároveň bolo jadro Androidu navrhnuté pre prácu na rôznom hardvéri. Systém tak môže byť použitý bez ohľadu na chipset, veľkosť či rozlíšenie obrazovky.

Samotná platforma Android dáva k dispozícii nielen operačný systém s používateľským prostredím pre koncových používateľov, ale aj kompletne riešenie nasadenia operačného systému (špecifikácia ovládačov a pod.) pre mobilných operátorov

a výrobcov zariadení a v neposlednom rade pre vývojárov aplikácií poskytuje efektívne nástroje pre ich vývoj - Software Development Kit. (Wikipedia Contributors, 2022)

3.6 IDE

Prostredie (často sa používa skratka IDE z angl. Integrated development environment) je software uľahčujúci prácu programátorov, väčšinou zameraný na jeden konkrétny programovací jazyk. Obsahuje editor zdrojového kódu, kompilátor, prípadne interpreter a väčšinou i debugger. (Wikipedia Contributors, 2020)

Mnohé moderné vývojové prostredia obsahujú aj prehliadač tried, inšpektor objektov a diagram hierarchie tried na použitie pri objektovo orientovanom vývoji softvéru. (2022)

3.7 ANDROID STUDIO

Android Studio je oficiálne integrované vývojové prostredie (IDE) pre operačný systém Android od spoločnosti Google, postavené na softvéri IntelliJ IDEA od spoločnosti JetBrains navrhnuté špeciálne na vývoj softvéru pre platformu Android. Je k dispozícii na stiahnutie pre operačné systémy Windows, MacOS a Linux. Android Studio nahrádza Eclipse Android Development Tools (E-ADT) ako primárne IDE pre natívny vývoj aplikácií pre platformu Android. (Wikipedia Contributors, 2021)

3.8 JAVA

Je objektovo orientovaný programovací jazyk. Je vyvíjaný spoločnosťou Oracle (pôvodne Sun Microsystems). Jeho syntax vychádza z jazykov C a C++. Je určený na to, aby programátorom umožnil písať raz, spustiť kdekoľvek (WORA - Write once, run

anywhere). Zdrojové programy sa nekompilujú do strojového kódu, ale do medzistupňa, tzv. "byte-code", ktorý nie je závislý od konkrétnej platformy. Tento byte-code neskôr vykonáva a spracováva interpreter, Java Virtual Machine. (Wikipedia Contributors, 2021)

3.9 XML

XML znamená eXtensible Markup Language, v preklade rozšíriteľný značkovací jazyk. Umožňuje jednoduché vytváranie konkrétnych značkovacích jazykov na rôzne účely a široké spektrum rôznych typov údajov.

Jazyk je určený predovšetkým na výmenu údajov medzi aplikáciami a na zverejňovanie dokumentov. Jazyk umožňuje opísať štruktúru dokumentu z hľadiska vecného obsahu jednotlivých častí a nezaobrá sa sám osebe vzhľadom dokumentu alebo jeho časti.

Zvláštnym ho robí to, že je široko prijímaný ako štandard so schopnosťou existencie na akejkolvek platforme. (Wikipedia Contributors, 2017)

3.10 JSON

JSON (JavaScript Object Notation) je formát na výmenu dát, ktorý používajú rozhrania API (aplikačné programové rozhrania). Umožňuje zoskupiť veľké množstvo dát do jedného balíka textu a v tejto podobe ich odovzdať do inej služby. Všetky dátové hodnoty majú priradený názvy, alebo kľúče a sú skombinované so špeciálnymi znakmi a to všetko spolu tvorí dátové objekty.

Je úplne všeobecný a môže slúžiť na prenos dát v ľubovoľnom programovacom alebo skriptovacom jazyku, pritom sú dáta pomerne čitateľné aj pre človeka. (Wikipedia Contributors, 2022)

3.11 FIREBASE

Firebase je platforma vyvinutá spoločnosťou Google na vytváranie mobilných a webových aplikácií. (Wikipedia Contributors, 2022)

Okrem toho ponúka tento nástroj aj možnosť cloud úložiska, vytvárania real-time database a niekoľko ďalších možností.

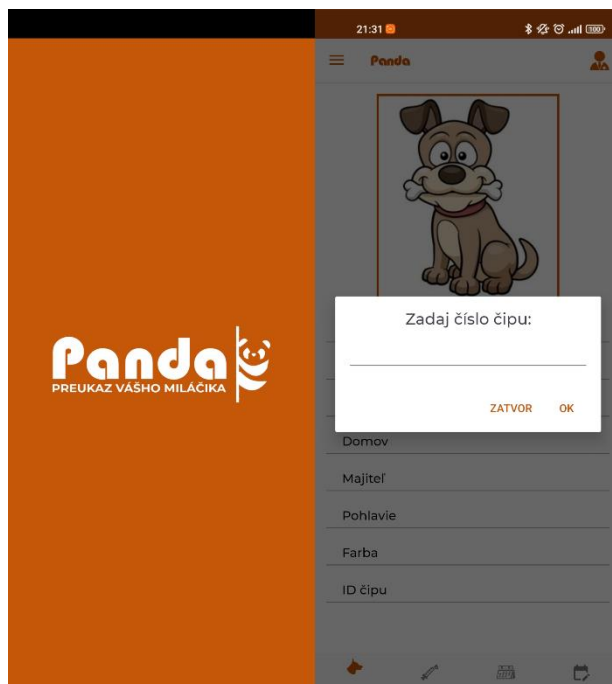
4 PRAKTICKÁ ČASŤ

4.1 ÚVODNÉ ZOBRAZENIE APLIKÁCIE

Aplikáciu tvoria dve aktivity. Prvá aktivita sa spúšťa hneď po samotnom spustení celej aplikácie a tvorí tzv. splash screen – uvítaciu obrazovku pre užívateľa. Druhá aktivita nasleduje po nej a spúšťa už fungovanie samotnej aplikácie. Aktivita obsahuje bočné menu tzv. drawer menu, spodné menu tzv. bottom menu a tiež kontajner pre fragmenty, v ktorom sa následne zobrazia podľa potreby. Pre prispôsobenie horného toolbaru sme pôvodný zakázali a vytvorili vlastný, ktorý sa tiež definuje pri vytváraní hlavnej aktivity.

Po spustení sa zobrazí alertDialog, ktorý požaduje od používateľa aby zadal číslo čipu. Používateľ má možnosť tento alertDialog aj zatvoriť, avšak aplikácia ho upozorní, že ak nezadá číslo čipu tak sa nebudú zobrazovať informácie o zvierati.

Po zadaní sa číslo uloží do premennej, ktorá je typu SharedPreferences (*SharedPreferences je ľahká a jednoduchá dátová štruktúra, ktorá ukladá dáta do páru kľúč – hodnota v systéme Android* (Joshi, 2019)), aby sa zachovalo aj po zavretí aplikácie.



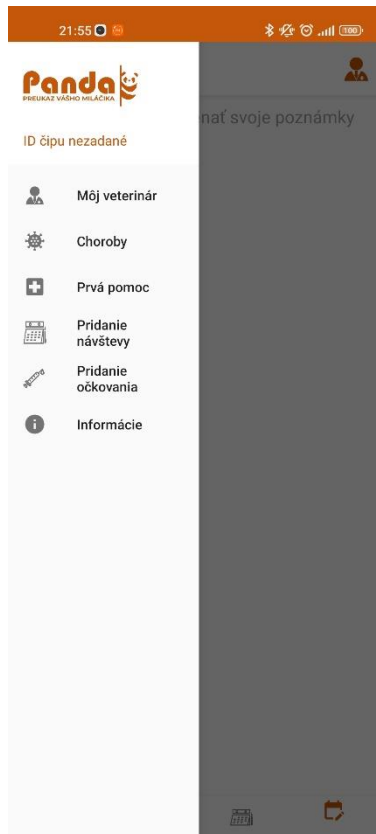
Obr. 1: Splash screen a obrazovka nasledujúcej aktivity

4.2 NAVIGÁCIA

Navigáciu tvoria v aplikácii dve menu. Prvým z nich je drawer menu (tiež bočné menu), ktoré definujeme v triede hlavnej aktivity (MainActivity). V tomto menu je definovaný objekt, kde je uvádzaný text, tzv. TextView, ktorý je prepojený s hlavnou aktivitou, aby sa tam mohlo zobrazovať číslo čipu. Vytvárame a definujeme tiež 6 položiek, kde každá položka otvára iný fragment. Pre rozloženie a funkciu tohto menu nám tiež pomáha externá knižnica Drawerlayout.

Druhým je bottom menu (tiež spodné menu), ktoré je tiež definované v triede hlavnej aktivity. Definujeme 4 položky, ktoré tvoria najdôležitejšie poslanie aplikácie. Ak je položka aktívna, tak jej farba je oranžová, ak je neaktívna tak je sivá. Zároveň po kliknutí na ňu sa zobrazí malá animácia kliknutia.

Pri spolupráci týchto dvoch menu je tiež zabezpečené, aby v prípade že je zobrazený fragment spodného menu nebola označená žiadna položka v bočnom menu ako aktívna.



Obr. 2: Drawer (bočné) menu, v pozadí spodné menu

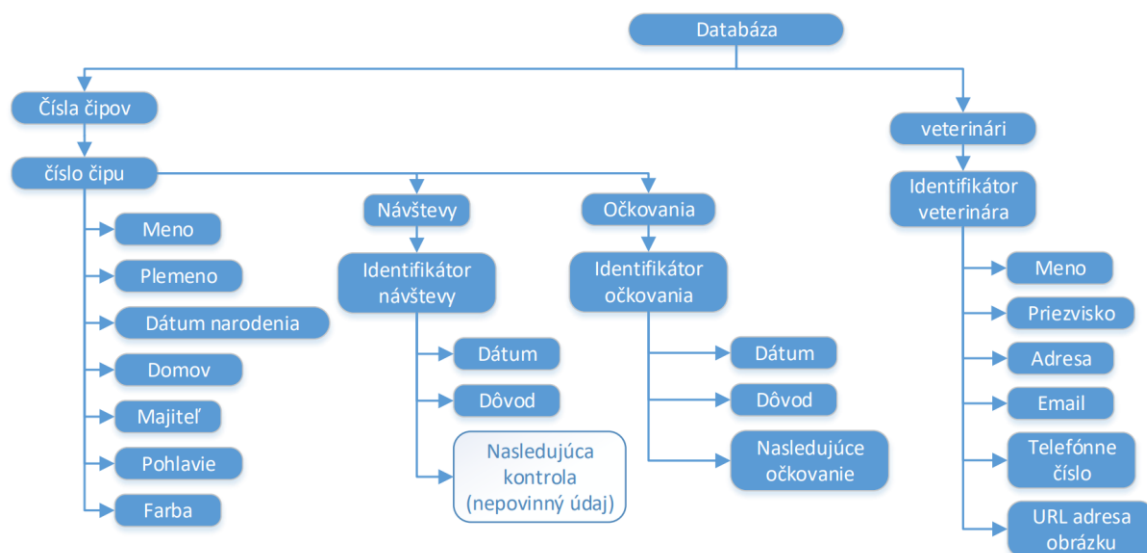
4.3 DATABÁZA

Databáza mobilnej aplikácie Panda je prevádzkovaná a spracovávaná v nástroji Firebase od spoločnosti Google. Realtime Database ukladá a pracuje s dátami vo formáte JSON v reálnom čase, čo sa môže ukázať ako komplikáciá pri vývoji aplikácií.

Keď sú pridávané dáta, môže sa stať, že aplikácia spadne pretože je to neočakávaná zmena, ktorú aplikácia nedokáže spracovať. Avšak aj to sa nám podarilo zabezpečiť či už podmienku if alebo blokom try catch.

Databáza obsahuje dve základné entity. Prvá z nich obsahuje čísla čipov zvierat. Následne sú k týmto čipom priradené ďalšie údaje ako meno zvieraťa, majiteľ, adresa, plemeno a ďalšie. Podentita sú návštevy, ktorá má niekoľko podentít delených podľa identifikátora návštevy. Jedna návšteva môže mať potom ešte dve alebo tri hodnoty. Podentita sú aj očkovania, ktorá na rozdiel od návštev má na konci stále 3 hodnoty.

Ďalšia zo základných entít obsahuje informácie o veterinároch, podľa identifikátora veterinára. Okrem základných informácií ako meno veterinára, adresa a ďalšie je v databáze uvedená aj URL adresa na obrázok veterinára. Obrázok je uložený na cloudovom úložisku, ktoré tiež poskytuje Firebase.



Obr. 3: Štruktúra databázy

4.4 ZOBRAZENIE OČKOVANÍ A NÁVŠTEV

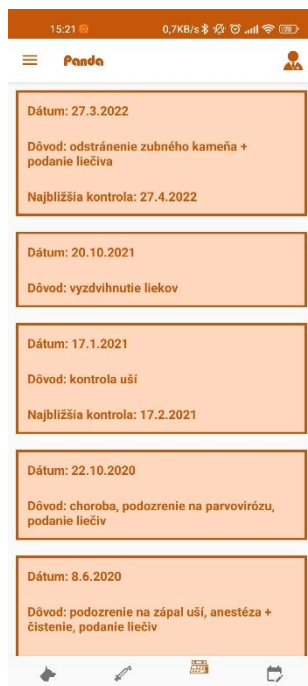
Zobrazenie je riešené pomocou fragmentov v hlavnej aktivite. Ich fungovanie je podobné avšak, rozhodli sme sa oddeliť ich do osobitných tried, ktoré spolupracujú, pretože ako každá aplikácia aj táto predpokladá vývoj a aktualizácie. Ak by sme v budúcnosti chceli pokračovať vo vývoji aplikácie a začali by sa očkovania a návštevy viac odlišovať, tak aktualizácia aplikácie bude jednoduchšia.

Pri vytváraní zobrazenia sa dáta najprv obnovia zo SharedPreferences a následne sa pokúsime aj o ich aktualizáciu. Po aktualizácii sa uložia aktualizované dáta opäť do SharedPreferences.

Sťahované dáta sa v programe ukladajú do osobitného objektu pre očkovania a návštevy pre jednoduchšiu prácu a lepší prehľad. Po načítaní dát nasleduje ich samotné zobrazenie.

Vytvoríme objekt, ktorý dedí vlastnosti a metódy z triedy ArrayAdapter, ktorá zabezpečuje zobrazovanie položiek v zozname.

Vo fragmente očkovaní sa zobrazuje v jednej položke údaj o dátume očkovania, o vakcíne a tiež o nasledujúcom očkovaní. Vo fragmente návštevy sa zobrazia informácie o dátume návštevy, dôvode a môže sa zobrazit' aj nepovinný údaj o nasledujúcej kontrole.



Obr. 4: Zobrazenie návštev

4.5 ZÁPIS DO DATABÁZY

V bočnom menu nájdeme aj dve položky, ktoré umožňujú zápis do databázy. V prvej môžeme zapísať návštevu a v tej druhej očkovanie. Táto časť slúži skôr pre veterinárov, ktorí môžu cez zadané heslo zapísať do databázy potrebné údaje pre zvierá.

Pri zapisovaní návštevy uvádzame ako povinný údaj dátum a dôvod kontroly a ako nepovinný údaj môže veterinár zadať aj nasledujúcu kontrolu. Povinný údaj je tiež ID čipu zvierat'a, aby sa tento údaj vedel v databáze kam zaradiť.

Všetky zapisované údaje majú pevný formát, ktorý je zvyčajne naznačený v textovom vstupe cez atribút hint. Okrem formátu údajov sa kontroluje aj heslo. Ak údaje nie sú v správnom formáte alebo je heslo nesprávne tak sa zápis do databázy nevykoná.

Pred zápisom do databázy sa ešte overuje pripojenie na internet prostredníctvom funkcie ping (*(angl. Packet InterNet Groper)* umožňuje preveriť funkčnosť spojenia medzi dvoma sieťovými rozhraniami (počítača, sieťového zariadenia) v počítačovej sieti, ktorá používa rodinu protokolov TCP/IP (Wikipedia Contributors, 2021)). Ak je ping úspešný, tak sa vykoná zápis a vymažú sa polia do ktorých sme zadávali hodnoty.

Na rovnakom princípe funguje aj zápis hodnôt pre očkovanie avšak už s inými hodnotami. V tomto prípade sú všetky uvedené hodnoty povinné.

The screenshot shows a mobile application interface with a white background and an orange header. The header contains the time '15:35', a data speed indicator '0,2KB/s', and various status icons. Below the header is a navigation bar with a hamburger menu icon, the word 'Panda' in orange, and a user profile icon. The main content area is titled 'Zapísanie návštevy zvierat'a' in orange. It contains a form with the following fields: '*Dátum:' with the value '1.1.2022', '*Dôvod:' (empty), 'Nasledujúca kontrola:' with the value '1.1.2022', '*ID čipu:' (empty), and '*Heslo:' (empty). A small red asterisk note at the bottom of the form states '* Pole označené hviečkou je povinné'. At the bottom of the form is an orange button labeled 'Odošli'. The bottom of the screen features a standard Android navigation bar with icons for back, home, and recent apps.

Obr. 5: Zápis návštevy do databázy

4.6 ZOZNAM VETERINÁROV

4.6.1 Zobrazenie veterinárov v zozname

V bočnom menu sa tiež nachádza položka s názvom Môj veterinár, ktorá zobrazuje veterinárov, ktorí sú zapísaní v databáze. Aj keď je v databáze uvádzané meno a priezvisko osobitne, pre používateľa sa tieto dáta upravujú do formátu „MVDr. + meno + priezvisko“, aby to bolo preňho zrozumiteľnejšie a estetickjšie. Pre buttony bol tiež zvolený jednotný dizajn, ktorý ladí k celému dizajnu aplikácie a zároveň sme zvolili pre nich nepravidelný tvar aby aplikáciu trochu oživil.

Na začiatku sa obnovia údaje veterinárov zo SharedPreferences a následne ak je to možné dáta sa obnovia aj online a aj dáta v SharedPreferences sa aktualizujú. Vlastnosti o veterinároch sa ukladajú do samotných objektov pre prehľad a lepšiu prácu s nimi.

Podobne ako pri očkovaníach a návštevách bolo potrebné vytvoriť inštanciu objektu, ktorý dedí vlastnosti z triedy ArrayAdapter, pre zobrazenie zoznamu týchto buttonov s menami veterinárov. V tomto objekte sa zabezpečuje, aby po kliknutí na button sa otvoril ďalší fragment s podrobnosťami o veterinárovi.



Obr. 6: Zoznam veterinárov

4.6.2 Zobrazenie samotného veterinára

Zobrazovanie je riešené formou jedného fragmentu, v ktorom sa údaje menia podľa potreby. Medzi údajmi o veterinárovi, ktoré získavame z databázy je aj URL adresa obrázka veterinára. Z adresy následne zobrazujeme obrázok pomocou externej knižnice Picasso. Knižnica má tiež veľkú výhodu v tom, že obrázky, ktoré sú sťahované z URL adresy sú v zariadení zároveň aj ukladané. Pre telefónne číslo a email sú nastavené tiež atribúty, ktoré zabezpečujú, že po kliknutí na číslo sa otvorí aplikácia Telefón a po kliknutí na email sa otvorí predvolená emailová aplikácia.



Obr. 7: Zobrazenie veterinára

4.6.3 Zvolenie obľúbeného veterinára

Pri zobrazení fragmentu sa menia tiež ikony na toolbare – namiesto veterinára sa zobrazuje šípka späť a ikona srdiečka.

Šípka späť nás vráti na predchádzajúci zoznam veterinárov.

Ikona srdca umožňuje zvoliť veterinára ako obľúbeného. Preddefinované je ikona nastavená na nevyplnené srdiečko. Po kliknutí sa zmení na vyplnené a zároveň sa daný veterinár uloží ako obľúbený. Následne kdekoľvek po kliknutí v aplikácii na ikonu veterinára sa zobrazí ten veterinár, ktorý bol zobrazený vo fragmente pri kliknutí na ikonu srdiečka. Ako obľúbený veterinár môže byť vždy zvolený len jeden z veterinárov.

5 ČASTI KÓDU

5.1 NASTAVENIE OBRÁZKA PRE ZVIERA

Keď používateľ klikne na obrázok psa, ktorý sa zobrazuje preddefinovane ponúkne mu možnosť si vybrať vlastný obrázok, resp. keď si už obrázok vybral a nastavil ponúkne mu možnosť obrázok vymazať.

Využívame funkciu AlertDialog, v ktorom nastavíme tri možnosti a to – vybrať obrázok z galérie, vymazať fotku alebo zrušiť AlertDialog.

Ak používateľ chce vybrať obrázok z galérie, tak vytvoríme objekt typu Intent, ktorému nastavíme požadovanú vlastnosť pre začatie akcie. Spustí sa proces otvorenia galérie a výberu obrázka. Po výbere sa zavolá metóda *startActivityForResult()*, v ktorej nastavíme čo sa stane s obrázkom. Nastavujeme ho do ImageView vo fragmente a zároveň ho zakódujeme do podoby vhodnej na ukladanie.

Ak si používateľ zvolí fotku vymazať tak sa na jej miesto nastaví pôvodná fotka, ktorá bola zobrazená preddefinovane.

```
private void selectImage() {
    final CharSequence[] options = {"Vybrať obrázok z galérie", "Vymazať fotku", "Zrušiť"};
    AlertDialog.Builder builder = new AlertDialog.Builder(this.getActivity());
    builder.setTitle("Pridaj si fotku miláčika!");
    builder.setItems(options, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int item) {
            if (options[item].equals("Vybrať obrázok z galérie")) {
                Intent intent = new Intent(Intent.ACTION_PICK);
                intent.setType("image/*");
                startActivityForResult(intent, IMAGE_PICK_CODE);
            } else if (options[item].equals("Vymazať fotku")) {
                imageDog.setImageResource(R.drawable.dog);
                SharedPreferences.Editor editor = settings.edit();
                editor.putString("IMG", "");
                editor.apply();
            } else if (options[item].equals("Zrušiť")) {
                dialog.dismiss();
            }
        }
    });
    builder.show();
}
```

Obr. 8: Metóda pre správu obrázka pre zviera

5.2 NASTAVOVANIE ČÍSLA ČIPU

Ak v aplikácii nie je nastavené číslo čipu, vytvorí sa AlertDialog. Užívateľ má dve možnosti – potvrdiť teda tlačidlo OK alebo možnosť zatvoriť AlertDialog.

Ak si vyberie možnosť OK overujeme, či užívateľ napísal nejaké číslo. Ak áno tak sa číslo uloží do premennej `cisloCipu` a zároveň do `SharedPreferences`. Zároveň sa číslo čipu nastaví aj v bočnom menu. Ak používateľ neuviedol nič tak sa vypíše správa, že zadal zlé číslo čipu. Následne sa v hlavnej aktivite opäť načíta fragment s domovskou stránkou, aby sa aktualizovalo číslo čipu a podľa pripojenia sa stiahli aj dáta o zvierati.

Ak si užívateľ vyberie možnosť zavrieť AlertDialog, tak sa vypíše správa, ktorá hovorí, že ak používa aplikáciu bez čísla čipu tak sa mu nebudú zobrazovať žiadne informácie o zvierati.

```
if (cisloCipu.equals("")) {
    AlertDialog.Builder builder = new AlertDialog.Builder(context: this);
    final View customLayout = getLayoutInflater().inflate(R.layout.vstup, root: null);
    builder.setView(customLayout);
    builder.create();
    final EditText cislo = customLayout.findViewById(R.id.vstupCisloCipu);
    builder.setPositiveButton(text: "OK", (dialog, which) -> {
        if (cislo.getText().toString().equals("")) {
            Toast.makeText(context: this, text: "Nesprávne číslo čipu", Toast.LENGTH_SHORT).show();
        } else {
            cisloCipu = cislo.getText().toString();
            SharedPreferences.Editor editor = settings.edit();
            editor.putString("CISLOCIPU", cislo.getText().toString());
            editor.apply();
            getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container, new HomeFragment()).commit();
            cisloCipu_Drawer.setText("ID čipu: " + cisloCipu);
        }
    });
    builder.setNegativeButton(text: "Zatvor", (dialog, which) -> {
        Toast.makeText(context: this, text: "Používaš aplikáciu bez čísla čipu, informácie o zvierati sa nebudú zobrazovať",
            Toast.LENGTH_LONG).show();
    });
    builder.show();
}
```

Obr. 9: Nastavenie čísla čipu

5.3 OVEROVANIE PRIPOJENIA

Pred zápisom do databázy overujeme pripojenie prostredníctvom nástroja ping, ktorý odosiela prázdny paket na IP adresu 8.8.8.8, čo je DNS služba google. Táto adresa sa veľmi často používa na overovanie dostupnosti pripojenia.

```
public boolean isOnline(){
    Runtime runtime = Runtime.getRuntime();
    try {
        Process ipProcess = runtime.exec( command: "/system/bin/ping -c 1 8.8.8.8");
        int    exitValue = ipProcess.waitFor();
        return (exitValue == 0);
    }
    catch (IOException e)          { e.printStackTrace(); }
    catch (InterruptedException e) { e.printStackTrace(); }

    return false;
}
```

Obr. 10: Metóda overujúca pripojenie

6 VÝSLEDKY PRÁCE

Výsledkom tejto práce je aplikácia, ktorá uvádza papierovú verziu preukazu domáceho zvierat'a do jeho elektronickej podoby. Okrem toho je rozšírená o niekoľko ďalších užitočných informácií, ktoré môžu uľahčiť a zdokonaľiť starostlivosť o domácich miláčikov. Aplikácia je po úprave, naplnení databázy a tiež po vhodnom rozhraní a nástroja na správu databázy pre veterinára, vhodná pre nasadenie do reálneho sveta.

Aplikácia je vytvorená, tak aby bola pre používateľa zrozumiteľná. Používateľ musí vedieť v aplikácii nájsť vždy to, čo potrebuje. Aplikáciu môžu používať aj starší používatelia a preto je jednoduchá a zároveň prepracovaná. Je určená na to, aby užívateľ vždy vedel nájsť potrebné informácie, vždy keď má pri sebe svoje mobilné zariadenie bez ohľadu na pripojenie, s výnimkou prvotného nastavenia čísla čipu, kedy sa musia stiahnuť informácie.

7 ZÁVERY PRÁCE

Ciele, ktoré sme si vytýčili pre túto prácu, sme aj splnili. Mobilná aplikácia Panda spĺňa všetky funkcie, ktoré sme jej pri plánovaní stanovili. Taktiež sme jej dali nádych jednotného dizajnu, ktorý spolu dobre ladí vo všetkých častiach aplikácie a zároveň pasuje aj k téme aplikácie. Prepojenie z databázou funguje takisto bezproblémovo. Dáta sú zobrazované v reálnom čase t. j. vtedy keď zapíšeme nový údaj do databázy ja možné ho aj bez aktualizácie vidieť aj v aplikácie.

Užívateľ môže zadať číslo čipu svojho domáceho miláčika a hneď po načítaní uvidí všetky jeho informácie a podrobnosti o očkovaní, kontrolách a celkovo o zdravotnom stave svojho miláčika. Aplikácia tiež ponúka možnosť vlastného zápisu poznámok v ľubovoľnej dĺžke, ktoré sa v aplikácii upravujú a užívateľ ich môže vidieť aj bez pripojenia na internet.

Pri priebežnom testovaní funkcií tohto projektu sme sa stretli s viacerými chybami, ktoré bolo pre bežné používanie aplikácie potrebné odstrániť. Tieto chyby sa vyskytovali prevažne pri prepojení s databázou, ale mnohokrát aj pri zobrazovaní obrázkov alebo práci so zobrazovaním už získaných údajov z databázy v zozname. Nakoniec bolo potrebné ošetriť chybné vstupy, ktoré by mohol užívateľ zadať či už nechtiac alebo zámerne.

Aplikácia má potenciál aj pre uvedenie na trh a reálne prepojenie z veterinárnymi ambulanciami, avšak počas jej tvorby nám napadlo ešte niekoľko ďalších funkcií a vylepšení, ktoré by bolo možné doprogramovať. Okrem nových funkcií by sme pred uvedením vylepšili ešte dizajn aplikácie a tiež bezpečnosť napr. aby bola databáza chránená pred neoprávneným vstupom.

Pri tvorbe tejto aplikácie sme uplatnili poznatky z mnohých odborných predmetov na našej škole, ale aj napriek tomu sme sa prostredníctvom nej naučili aj mnoho nových vecí. Jej programovanie bolo pre nás okrem povinnosti aj radosťou keďže nás programovanie mobilných aplikácií baví.

ZHRNUTIE

Cieľom tejto aplikácie bolo zjednodušiť prístup k zdravotnej karte našich domácich miláčikov a zároveň na jednom mieste poskytnutie ďalších užitočných informácií. Aplikácia je vhodná pre každého majiteľa domáceho zvierat'a a najmä pre psov, keďže informácie sú najmä o starostlivosti o nich. Používateľ vie rýchlo prístupit' k informáciám o zvierati a o jeho zdravotnom stave. Okrem toho vie tiež pristupovať a priamo odtiaľ využívať informácie o svojom veterinárovi alebo v prípade potreby informácie o iných veterinároch. Pri vypracovaní tejto práce sme sa naučili lepšie pracovať s Android Studiom, databázami a taktiež knižnicami pre spomenuté nástroje. Takisto sme rozšírili svoje znalosti o nástroji Firebase, s ktorým sme pracovali po prvýkrát. Nadobudli sme nové poznatky ohľadom vývoja mobilných aplikácií a vďaka chybám, ktoré sa vyskytli počas tvorby projektu, vieme čomu sa vyvarovať v budúcnosti.

RESUMÉ

The aim of this application was to simplify access to the health card of our pets and at the same time to provide other useful information in one place. The application is suitable for every pet owner and especially for dogs, as the information is mainly about caring for them. The user can quickly access information about the animal and its health. In addition, he can access and directly use information about his veterinarian or, if necessary, information about other veterinarians. During the elaboration of this work, we learned to work better with Android Studio, databases and also libraries for the mentioned tools. We also expanded our knowledge of the Firebase tool we worked with for the first time. We have gained new knowledge about the development of mobile applications and thanks to the mistakes that occurred during the creation of the project, we know what to avoid in the future.

ZOZNAM POUŽITEJ LITERATÚRY

- Hmn Wiki Community.** Mobilné zariadenie. *Hmn Wiki*. [Online]
https://hmn.wiki/sk/Mobile_device.
- Joshi, Ashish. 2019.** SharedPreferences in android tutorial. *Technopoints*. [Online] 3. 1 2019. <https://technopoints.co.in/sharedpreferences-android-tutorial/>.
- Lang, Patrik. 2020.** Mobilná aplikácia vs. responzívne webové aplikácie. Aké sú výhody a nevýhody? *eWay-Blog*. [Online] 9. 7 2020. <https://www.eway-crm.com/sk/blog/podnikanie/mobilna-aplikacia-vs-responzivne-webove-aplikacie-ake-su-vyhody-a-nevyhody/>.
- Macho, Tomáš. 2020.** Mobilné aplikácie v skratke. *Elite Blog - Monday Lovers*. [Online] 17. 1 2020. <https://www.eliteml.sk/blog/mobilne-aplikacie-v-skratke/>.
- 2022.** Softvérové nástroje, systémy nástrojov, aplikačný softvér. *sukachOFF*. [Online] 12. 1 2022. <https://sukachoff.ru/sk/vosstanovlenie/k-instrumentalnym-programmnym-sredstvam-otnosyatsya-instrumentalnye/>.
- Wikipedia Contributors. 2022.** Android (operačný systém). *Wikipedia*. [Online] 2. 5 2022.
[https://sk.wikipedia.org/wiki/Android_\(opera%C4%8Dn%C3%BD_syst%C3%A9m\)](https://sk.wikipedia.org/wiki/Android_(opera%C4%8Dn%C3%BD_syst%C3%A9m)).
- **2021.** Android Studio. *Wikipedia*. [Online] 27. 11 2021.
https://sk.wikipedia.org/wiki/Android_Studio.
- **2021.** Databáza. *Wikipedia*. [Online] 7. 9 2021.
<https://sk.wikipedia.org/wiki/Datab%C3%A1za>.
- **2022.** Firebase. *Wikipedia*. [Online] 21. 4 2022.
<https://en.wikipedia.org/wiki/Firebase>.
- **2021.** Java (programovací jazyk). *Wikipedia*. [Online] 30. 12 2021.
[https://sk.wikipedia.org/wiki/Java_\(programovac%C3%AD_jazyk\)](https://sk.wikipedia.org/wiki/Java_(programovac%C3%AD_jazyk)).
- **2022.** JavaScript Object Notation. *Wikipedia*. [Online] 24. 1 2022.
https://cs.wikipedia.org/wiki/JavaScript_Object_Notation.
- **2022.** Mobilné zariadenie. *Wikipedia*. [Online] 10. 2 2022.
https://sk.wikipedia.org/wiki/Mobiln%C3%A9_zariadenie.
- **2021.** Ping. *Wikipedia*. [Online] 1. 4 2021. <https://sk.wikipedia.org/wiki/Ping>.
- **2014.** Vyhľadávací jazyk. *Wikipedia*. [Online] 16. 11 2014.
https://sk.wikipedia.org/wiki/Vyh%C4%BEd%C3%A1vac%C3%AD_jazyk.
- **2020.** Vývojové prostredie. *Wikipedia*. [Online] 22. 6 2020.
https://sk.wikipedia.org/wiki/V%C3%BDvojov%C3%A9_prostredie.
- **2017.** XML. *Wikipedia*. [Online] 13. 5 2017. <https://sk.wikipedia.org/wiki/XML>.

ZOZNAM OBRÁZKOV

Obr. 1: Splash screen a obrazovka nasledujúcej aktivity.....	13
Obr. 2: Drawer (bočné) menu, v pozadí spodné menu	14
Obr. 3: Štruktúra databázy	15
Obr. 4: Zobrazenie návštev	16
Obr. 5: Zápis návštevy do databázy	17
Obr. 6: Zoznam veterinárov	18
Obr. 7: Zobrazenie veterinára	19
Obr. 8: Metóda pre správu obrázka pre zviera.....	21
Obr. 9: Nastavenie čísla čipu	22
Obr. 10: Metóda overujúca pripojenie	23