UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

OSOBNÝ MOBILNÝ ASISTENT PRE DIABETIKOV

Bakalárska práca

Bratislava, 2016 **Jaroslav Ištok**

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

OSOBNÝ MOBILNÝ ASISTEN PRE DIABETIKOV

Bakalárska práca

Študijný program:	Aplikovaná informatika
Študijný odbor:	2511 Aplikovaná informatika
Školiace pracovisko:	Katedra aplikovanej informatiky
Školiteľ:	Ing. František Gyárfáš, PhD.

Bratislava, 2017 **Jaroslav Ištok**

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Jaroslav Ištok
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: aplikovaná informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Osobný mobilný asistent pre diabetikov
Personal Mobile Assistant for Managing Diabetes

Cieľ: Cieľom aplikácie bude poskytnúť diabetikom, najmä deťom, osobného mobilného asistenta. Aplikácia bude riadiť a kontrolovať plán každodenných činností, pripomienkovať časy merania, pichanie inzulínu, prehliadky, tolerančné testy. Aplikácia bude priebežne zbierať a vyhodnocovať údaje a vytvárať zdravotný profil pacienta pre potreby lekára, generovať štatistiky a prezentovať grafy. Súčasťou aplikácie bude databáza užitočných údajov, ako sú napríklad jedlá s ich nutričnými hodnotami, či dôležité informácie o diabete. Projekt bude realizovaný ako mobilná aplikácia s webovým rozhraním. Umožní synchronizáciu údajov do cloudu a prístup k nim cez webový prehliadač. Vyvíjaná bude pre mobilnú platformu Android. Plánované technológie: Java, PHP, MySQL, Javascript, HTML5, CSS, Ajax.

Vedúci: Ing. František Gyarfaš, CSc.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.

Dátum zadania: 10.10.2016

Dátum schválenia: 17.10.2016 doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Čestné vyhlásenie

Čestne vyhlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením vedúceho bakalárskej práce, s použitím uvedenej literatúry a zdrojov dostupných na internete.

V Bratislave, dňa 23.05.2015

_____

Meno a priezvisko

Poďakovanie

Text poďakovania

Abstrakt

Text abstraktu v slovenskom jazyku

Kľúčové slová: kľúčové slová

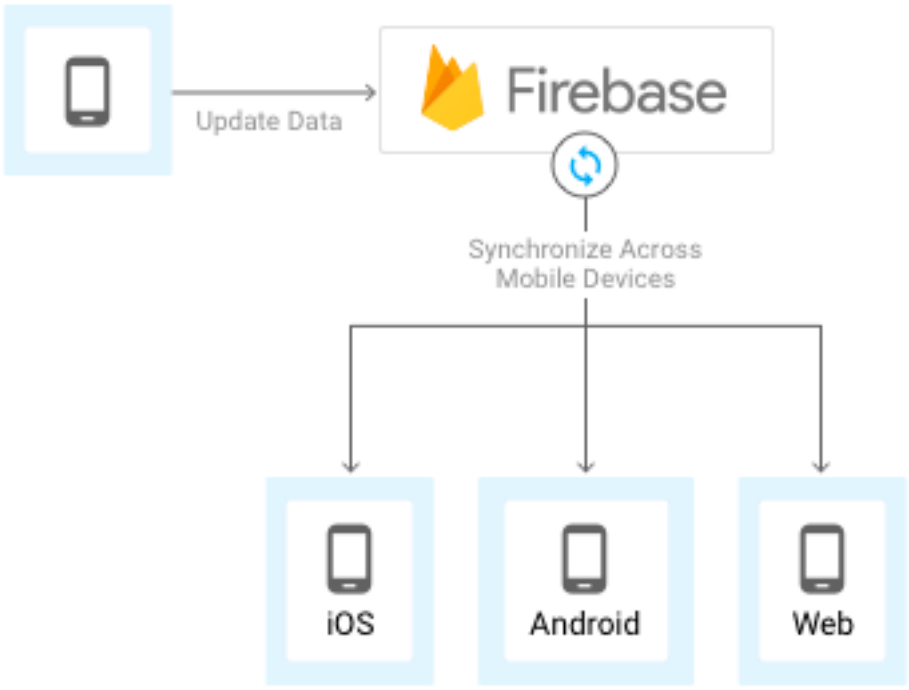
Abstract

Abstract in English language

Key words: keywords.

Obsah

Zoznam obrázkov



Úvod

Účel aplikácie

Aplikácia poskytne diabetikom nástroj na zjednodušenie každodenného života. Bude mať tri primárne úlohy. Bude pripomínať používateľovi počas dňa časy merania glykémie a pichania inzulínu. Umožní zaznamenávať si namerané hodnoty cukru v krvi, hodnoty pichnutého inzulínu, druh inzulínu a ďalších údajov. Aplikácia bude vedieť tieto údaje vyhodnocovať a generovať z nich grafy. Všetky údaje budú synchronizované a prístupné aj cez webové rozhranie aplikácie, ktoré bude slúžiť ako administrácia. Hlavným cieľom aplikácie je odbremeniť diabetika od neustáleho sledovania času, čo je obzvlášť dôležité u detí a teenagerov, ktorí si ešte nemajú dostatok zodpovednosti.

1 Analýza

1.1 Cukrovka

1.1.1 Úvod do problematiky

Po konzumácii jedla sa všetky sacharidy obsiahnuté v jedle vstrebajú do krvi. Zložené sacharidy (škrob, laktóza, sacharóza) pomalšie, pretože sa musia najskôr rozložiť v tráviacom trakte. Rýchle sacharidy (fruktóza, glukóza) sa vstrebú rýchlejšie do krvného obehu. Na to aby svaly vedeli využiť energiu z glukózy v krvi je potrebný hormón inzulín. Inzulín má 3 základné funkcie. Umožňuje svalovým bunkám využívať glukózu ako zdroj energie, tukovým tkanivám ukladať glukózu v podobe tuku (presne takto vzniká nadváha) a pečeni pomáha premienať glukózu na glykogén a vytvárať zásoby. Glykogén je krvný cukor a telo ho používa ako zdroj energie, keď máme nedostatok glukózy v krvi, jeho zásoby sú obmedzené. Inzulín je tvorený v pankreasi (podžalúdková žľaza) B-bunkami. Hormón, ktorý má presne opačný účinok ako inzulín je glukagón. Glukagón je tvorený v A-bunkách podžalúdkovej žľazy a hovorí pečeni, aby premenila uložený glykogén späť na glukózu. Tento hormón sa vylučuje, keď máme nedostatok glukózy v krvi.

1.1.2 Delenie cukrovky

Cukrovka je ochorenie pri ktorom je zmenšená alebo úplne zastavená prirodzená tvorba inzulínu v pankreasi. B-bunky reagujú na prítomnosť glukózy v krvi a produkujú inzulín na jeho zníženie a A-bunky reagujú na nízku hladinu glukózy v krvi. Poznáme dva základné typy cukrovky. V skutočnosti sú to úplne rozdielne ochorenia s podobnými následkami.

Cukrovka 1. Typu patrí medzi auto imúnne ochorenia. Imunitný systém človeka ničí B-bunky v pankreasi a tým sa zastaví tvorba inzulínu. Pri cukrovke prvého typu je nutná liečba pomocou syntetického inzulínu, ktorý musí diabetik prijímať buď pomocou inzulínového pera alebo pomocou inzulínovej pumpy. Ani jedno riešenie nie je dokonalé.

Cukrovka 2. Typu je ochorenie spôsobené čiastočne genetickou predispozíciou a čiastočne nesprávnym stravovaním. Je spôsobené tzv. Inzulínovou rezistenciou, čo je jav

pri ktorom bunky prestanú korektne reagovať na inzulín, čiže začnú byť rezistentné. Pri tomto type v mnohých prípadoch nie je potrebná liečba pomocou inzulínu a častokrát stačí úprava stravy a životosprávy.

Medzi najzávažnejšie riziká cukrovky patrí hypoglykémia a hypoglykemická kóma, čo je stav akútneho nedostatku glukózy v krvi (pod 2mmol/l). Tento stav je veľmi nebezpečný pre diabetika a preto by mal mať pri sebe zdroj rýchleho cukru, v ťažkých prípadoch sa pichá priamo do krvi hormón glukagón, ktorý vynúti syntézu glykogénu v pečeni. Tu je nutné upozorniť na konzumáciu alkoholu. Alkohol znižuje hodnotú glykémie. V prípade zvýšenej hladiny alkoholu v krvi sa v pečení primárne odburáva alkohol, čo blokuje premenu glykogénu na glukózu. Toto môže spôsobiť veľké komplikácie v prípade hypoglykémie, pretože je takmer nemožné zdvihnúť hladinu glukózy v krvi a každý diabetik si na to musí dávať veľký pozor. Hypoglykémia tiež ovplyvňuje náladu a môže spôsobovať prejavy agresie, na čo treba brať ohľad pri diabetikoch. Druhým „tichším“ nebezpečenstvom je hyperglykémia (vysoká hodnota cukru v krvi, nad 10mmol/l), ktorá má za následok postupné usádzanie glukózy sa v kĺboch, obličkách, môže poškodiť zrak, alebo obličky. Vysoká glykémia je spôsobená nadmernou konzumáciou sacharidov v strave a nedostatočným množstvom prijatého inzulínu. Hyperglykémia a nadmerná konzumácia cukru môže u detí spôsobovať hyperaktivitu a aj obezitu. Aj zdravý človek by si mal udržiavať čo najnižšiu hodnotu inzulínu v krvi a vyvarovať sa glykemickým šokom (konzumácia veľkého množstva cukrov v krátkom čase), môže tým predchádzať napríklad vzniku cukrovky druhého typu v neskoršom veku.

Dobre kompenzovaný diabetik môže byť úplne zdravý, len si musí dodržiavať diétu a časy pichania inzulínu. Ideálom je udržiavať hodnoty glykémie pod 10 mmol/l. Toto je pomerne náročná úloha pre deti a teenegerov, najmä v dnešnej dobe, kde máme cukor takmer v každom jedle. Preto som sa rozhodol vytvoriť aplikáciu, ktorá im pomôže a uľahčí manažovanie tohto ochorenia.

1.2 Analýza podobných riešení

V nasledujúcej kapitole sa zameriam na podobné aplikácie, aké sú ich výhody a nevýhody, ale najmä v čom je moja aplikácia odlišná. Analýza zahŕňa dve bakalárske práce s podobnou témou a jednu profesionálnu aplikáciu.

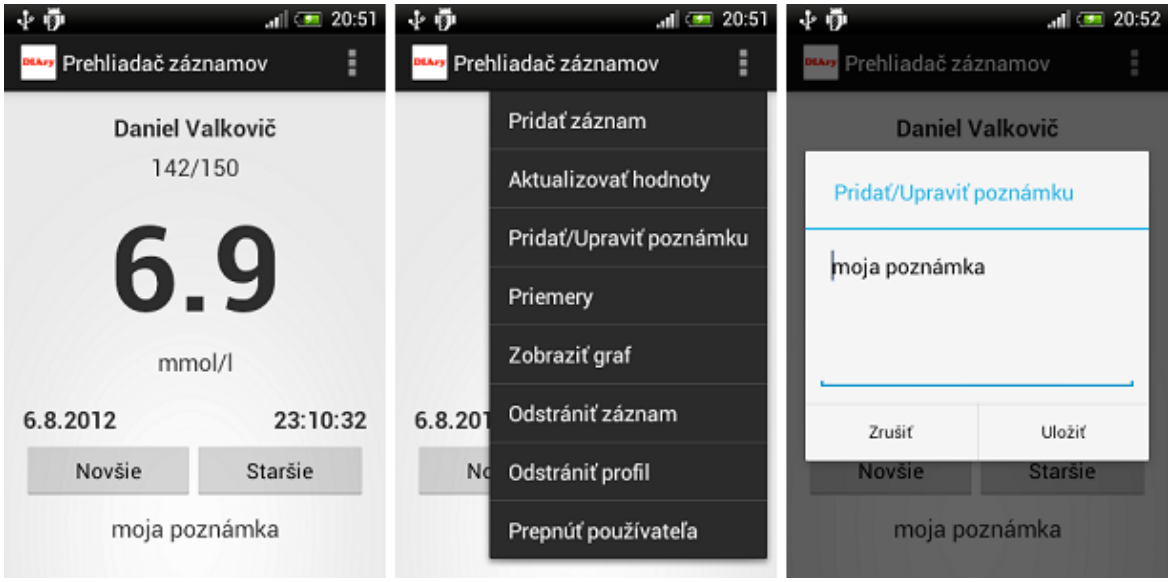
1.2.1 Elektronický denník diabetika (DIARy) – bakalárska práca

Cieľom bakalárskej práce bolo vytvoriť aplikáciu, ktorá umožní kopírovať namerané hodnoty cukru v krvi zo špecializovaných glukomerov, ktoré sa vedia bezdrôtovo pripojiť k mobilnému telefónu. Aplikácia umožní vytvoriť profil používateľa. Hodnoty nameranej glykémie sa dajú zadať aj manuálne a taktiež sa dajú editovať. Aplikácia vie zobraziť graf hodnôt glykémie a manažovať viacero používateľských profilov. Neobsahuje možnosť vytvorenia používateľa a prihlásenie pomocou mena a hesla a teda ani žiadne zabezpečenie.

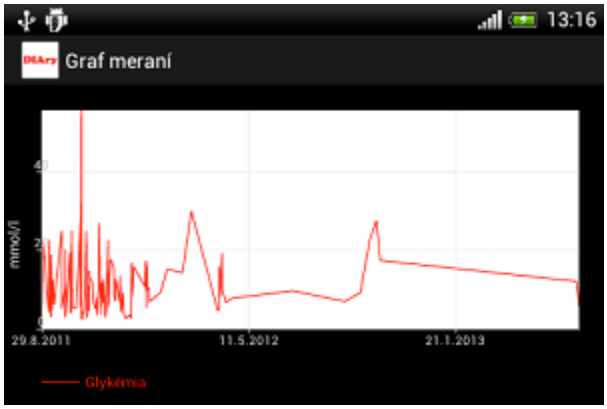
Aplikácia neobsahuje možnosť ukladať množstvo dôležitých údajov, ako napríklad hodnoty inzulínu, rovnako nemá ani webové rozhrnie. Tieto údaje sú pre diabetika dôležité. Uložené dáta ukladá iba lokálne v pamäti telefónu a nesynchronizuje ich s cloudovou databázou. Nevie pripomínať diabetikovi časy jedenia a merania inzulínu počas dňa. Slúži iba ako lokálna databáza hodnôt glykémie.

V porovnaní s mojou aplikáciou má DIARy iný účel, slúži primárne na sťahovanie dát zo špecializovaného glukomeru a ich uloženie do databázy. Moja aplikácia slúži primárne ako asistent pre diabetika, ktorý synchronizuje dáta aj s cloudom.

Na obrázkoch nižšie možno vidieť používateľské prostredie aplikácie. Obr.2 zľava je zobrazená konkrétna hodnota glykémie, v strede je menu aplikácia so všetkými možnosťami a napravo je dialógové okno na pridanie poznámky k nejakému záznamu gykémie. Na Obr.3 je možno vidieť zobrazenie grafu nameraných hodnôt glykémie.



Obr. 1 DIARy



Obr. 2 DIARy graf

1.2.2 Medicínsky softvér na tvorbu diét

Ide o webovú aplikáciu postavenú na Symfony PHP frameworku určenú na tvorbu diét. Kedže liečba cukrovky zahŕňa dodržiavanie optimálnej diéty, zahrnul som túto prácu do analýzy podobných prác. Táto aplikácia slúži na vytváranie jedálnych lístkov na základe vxstupných údajov, ktorými sú príjem kalórii, bielkovín, tukov a cukrov. Umožňuje generovať jednálne lístky a vytvárať "taniere" pre jednotlivé jedlá dňa. Aplikácia vie manažovať viacero používateľov, rozlišuje aj role používateľov (administrator, super používateľ a používateľ), ktoré majú rôzne práva. Ponúka možnosť vyhľadávania v jedlách cez kategórie ale aj pomocou jednoduchého filtra. Nájdené potraviny môže pridať do taniera. Na nasledujúcom obrázku je možno vidieť tanier aj s potravinami.

Obr. 3 Tanier

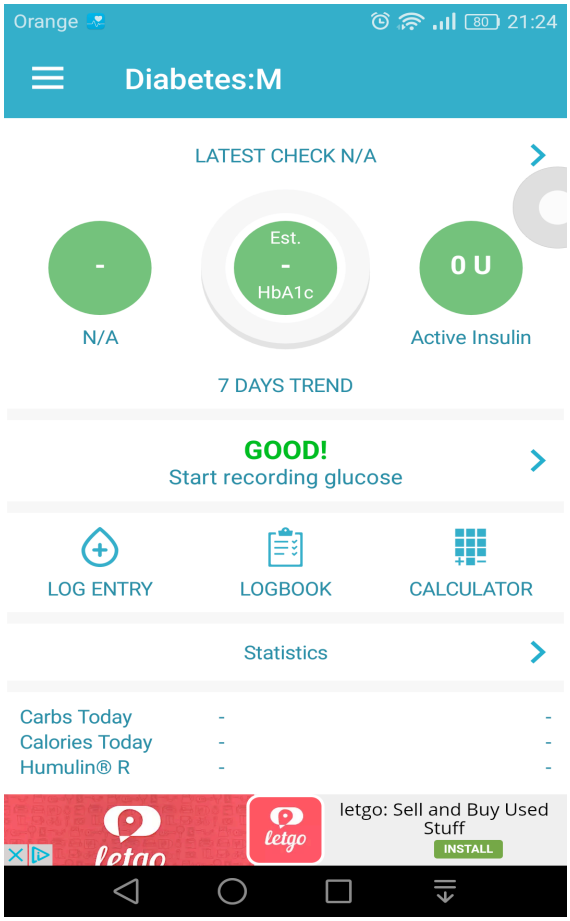
Ide o zaujímavú prácu, ktorá by mohla slúžiť ako doplnková aplikácia k mojej bakalárskej práci. Diabetikovi by uľahčila vytváranie jedálnych lístkov.

1.2.3 Diabetes:M

Diabetes:M je existujúca profesionálna aplikácia v anglickom jazyku určená pre diabetikov, zároveň je aj inšpiráciou pre moju bakalársku prácu. Slúži ako mobilný assistent, umožňuje zaznamenávať hodnoty glykémii, počítať glykemické hdonoty potravín pomocou kalkulačky, vytvárať pripomienky, zobrazovať grafy, exportovať údaje, vytvoriť podrobný profil a množstvo ďalších funkcii. Umožňuje zálohovať dáta pomocou rôznych cloudových služieb, ako napríklad Google Drive, alebo Dropbox. Ako hlavnú nevýhodu vidím komplexnosť, množstvo reklám vo free verzii a neprehľadnosť, čo nie je vhodné pre deti a dospievajúcich. Verzia bez reklám je spoplatnená.

Narozdiel od mojej aplikácie nemá webové rozhranie ani synchronizáciu v reálnom čase. Je to čisto mobilná aplikácia. Zaujímavou funkcionalitou tejto aplikácie je napríklad možnosť zaznamenávať si miesta jednotlivých vpichov na prstoch, takže diabetik vidí, na ktoré miesta si najčastejšie pichá do prstov.

Na nasledujúcom obrázku je vidno hlavná aktivita, ktorá sa zobrazí po spustení aplikácie.



Obr. 4 Diabetes:M

1.3 Analýza použitých technológii

V nasledujúcej kapitole popíšem technológie, ktoré som použil pri programovaní aplikácie.

1.3.1 Android

Android je mobilný operačný systém, vytvorený spoločnosťou Google, založený na linoxovom kerneli určený primárne pre mobilné zariadenia s dotykovým displayom. Aplikácie pre Android sa programujú v jazyku Java. Google v spolupráci s firmou JetBrains vytvoril integrované vývojové prostredie pre vývoj android aplikácii Android Studio, ktoré je dostupné zdarma. Obsahuje všetky potrebné nástroje pre vývoj aplikácii, ako napríklad debbuger, či manažér rôznych verzii Android SDK. Súčasťou inštalácie Android Studia sú aj simulátory androidových telefónov, na ktorých je možné testovať

aplikácie na rôznych verziach Androidu. Práve množstvo aktívne používaných verzii vidím ako hlavnú nevýhodu Andoidu. Problémy sú najmä s kompatibilitou jednotlivých verzii, v súčastnosti by mala aplikácia podporovať verzie 4, 5, 6, 7, čo je pre programátora značná komplikácia a množstvo testovania navyše.

Vyvíjať moju aplikáciu primárne pre Android som sa rozhodol z dôvodu veľkého podielu na trhu mobilných telefónov a tiež z toho dôvodu, že veľká časť mojej cieľovej skupiny (deti a teenegeri) používajú mobilný telefón s Androidom. V budúcnosti nevylučujem možnosť vytvorenia verzie pre zariadenia s operačným systémom iOS.

1.3.2 Firebase

Firebase je backendová platforma od spoločnosti Google, ktorá obsahuje nástroje na vývoj mobilných a webových aplikácii, ktoré využívajú backend systém na ukladanie údajov a ich synchronizáciu. Najdôležitejším nástrojom, ktorý firebase platforma ponúka pre moju aplikáciu je real time databáza, ktorá synchronizuje údaje medzi zariadeniami v reálnom čase, umožňuje tiež nadstaviť práva a obmedzenia prístupu (čítanie a zapisovanie) pre používateľov pomocou pravidiel. Slúži aj ako lokálna databáza, vie pracovať aj offline. V offline móde ukladá všetky vykonané zmeny, ktoré následne po pripojení k internetu zosynchronizuje, pričom má zabudovaný aj mechanizmus na riešenie konfiliktov, ktoré sa môžu vyskytnúť. Druhým užitočným nástrojom je autentifikácia používateľov. Firebase ponúka možnosť autentifikácie používateľa pomocou google, facebook, twittera alebo github účtu, prihlásenie ako anonymný používateľ alebo prihlásenie pomocou mena a hesla. Nechýba ani možnosť nadstaviť notifikačné emaily a resetovať heslo pri zabudnutí hesla. Firebase taktiež ponúka možnosť posielania notifikácii používateľom, kompletnú analytiku (štatistiky používania aplikácie, počet inštalácii, možnosť definnovať funnely na sledovanie práce používateľov s aplikáciou), možnosť meniť nastavenia aplikácie cez webové rozhranie, hosting, priestor na ukladanie multimediálnych dát aplikácie, zaznamenávanie pádov aplikácie, a testovač, ktorý vie automatizovane otestovať funkcionalitu GUI pomocou automatizovaného bota v rôznych verziach systému Android. Firebase ponúka API pre Android, iOS, Javascript, C++, Unity, Javu, Node.js a existujú aj knižnice pre PHP

a modul pre Angular. Pre potreby mojej aplikácie je to ideálne riešenie a ponúka, všetko čo potrebujem.

Firebase vo verzii zadarmo ponúka 1GB priestor v real time databáze, 10GB mesačne prenesených dát, 100 aktívnych pripojení súčastne, čo je pre moju aplikáciu viac ako postačujúce.

1.3.3 Angular

Angular je frontendový framework vyvíjaný spoločnosťou Google. Primárne je určený na vývoj komplexných webových aplikácii. Ako programovací jazyk využíva Typescript, ktorý je nadtavbou Javascriptu a pridáva mu typovanie. Typescript sa compiluje do čistého Javascriptu, preto nie je žiadny problém s kompatibilitou a skompilovaná Angular aplikácia pobeží na ľubovoľnom browseri.

Angular má oficiálne API pre prácu s firebase real time databázou angularfire2. Angular používam ako framework na vytvorenie webovej časti mojej aplikácie, pretože vie dobre spolupracovať s firebase a ponúka mi zaujímavé možnosti pri programovaní webovej časti aplikácie. V html elementoch umožňuje napríklad vytvoriť tzv. „Two way binding“, čo znamená, že keď používateľ píše do textového poľa, tak sa text automaticky synchronizuje napríklad s firebase a naopak, ak sa zmenia dáta vo firebase, tak sa zmeny prejavia real time aj v textovom poli. Angular obsahuje aj modul na vytvorenie jednoduchého routovania medzi stránkami, preto je ideálnym riešením pre moju aplikáciu.

1.3.4 Boostrap

Na vytvonie webovej časti používam frontendový framework Bootstrap. Je určený na tvorbu responzívnych webov. Obsahuje veľa komponentov a ponúka elegantné riešenie pre frontend mojej aplikácie. Je naprogramovaný v HTML, CSS a Javascript. HTML je značkovací jazyk, používaný na tvorbu webových stránok. CSS sú kaskádové štýly, pomocou, ktorých definujeme štýly pre jednotlivé HTML elementy na stránke. Najpoužívanejším komponentom Bootstrapu je grid systém, ktorý umožňuje vytvárať responzívne webové stránky. Často je využívané aj responzvíne navigačné menu a responzívne Bootstrap tabuľky.

2 Návrh

2.1 Špecifikácia

2.1.1 Cieľ projektu

Cieľom tejto bakalárskej práce je vytvoriť mobilnú aplikáciu, ktorá bude slúžiť/todo dokpisat specifikaciu

2.1.2 Cieľová skupina používateľov

Cieľovou skupinou používateľov sú ľudia, s ochorením diabetes. Aplikácia bude primárne určená pre nižšiu vekovú skupinu. Aplikácia je určená tiež pre rodičov, alebo zákonných zástupcov detí a dospievajúcich s diabetom. Môže poskytnúť užitočné informácie aj pre lekára, ku ktorému, dieťa chodí na prehliadky. Pre rodičov, zákonných zástupcov a lekára bude primárne určená webová časť aplikácie, na prístup k nej bude potrebný iba webový prehliadač a všetky zmeny budú synchronizované s mobilnou časťou aplikácie. Mobilná aplikácia bude plniť funkciu asistenta pre dieťa/dospievajúceho a bude mať obmedzené možnosti úprav údajov a pripomienok (niektoré údaje sa nebudú dať zmeniť v aplikácii, prípadne až po zadaní autorizačného kódu).

2.1.3 Používateľské scenáre

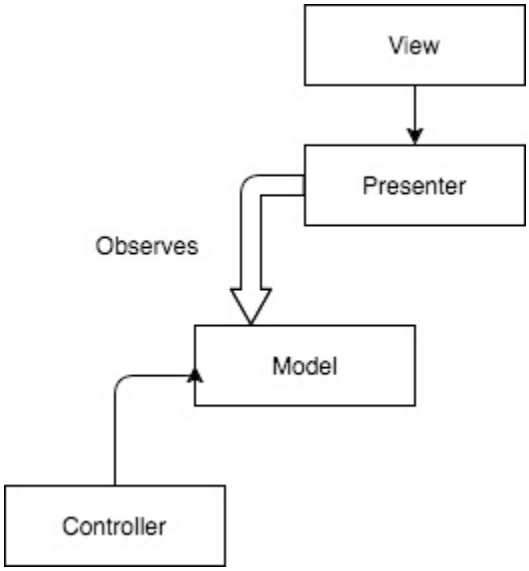
Dieťa s diabetom si musí počas dňa niekoľko krát pichnúť inzulín a zmerať hodnotu cukru v krvi. Mobilná časť aplikácie mu slúži ako asistent a pripomienkovač. Na používanie aplikácie je potrebné zaregistrovať sa a prihlásiť. Registrácia a prihlásenie prebiehajú online, preto je potrebné pripojenie k internetu. Po úspešnej registrácii a prihlásení sa do aplikácie, už ďalej pripojenie do internetu nie je potrebné. Aplikácia umožní definovať si pripomienky. Keď sa spustí pripomienka, mobilný telefón začne vibrovať a zvoniť a zobrazí obrazovku z ktorej diabetik bude môcť ľahko uložiť dôležité údaje, prostredníctvom pridania nového záznamu. Záznamy z meraní sa dajú pridať aj ručne. V aplikácii si ďalej vie zobraziť graf nameraných hodnôt cukru v krvi na ktorom môže vidieť ako dobre sa mu darí kompenzovať cukrovku. Diabetik si v mobilnej aplikácii vie prezrieť aj všetky záznamy, ktoré vie filtrovať podľa dátumu, čiže si vie prezrieť záznamy len z určitého obdobia.

Architektúra mobilnej aplikácie

Počas návrhu som sa rozhodol pre použitie viacerých návrhových vzorov. Narazil som aj na množstvo problémov a snažil som sa nájsť adekvátne riešenia.

Android API poskytuje vývojárom pomerne veľkú slobodu pri vývoji aplikácii. Nenúti nás používať MVC architektúru, či componenty ako Content Provider pri práci s lokálnou databázou. Napriek tomu je podľa mňa dôležité pri vývoji (nielen) mobilných aplikácii vhodné používať návrhové vzory, refaktoring a ďalšie princípy objektovo orientovaného programovania.

Model-View(Presenter)-Controller je návrhový vzor, ktorý sa používa v rôznych typoch aplikácii, populárny je najmä v backendoch webových aplikáci, ale najväčšie využitie má práve pri vývoji mobilných aplikácii.



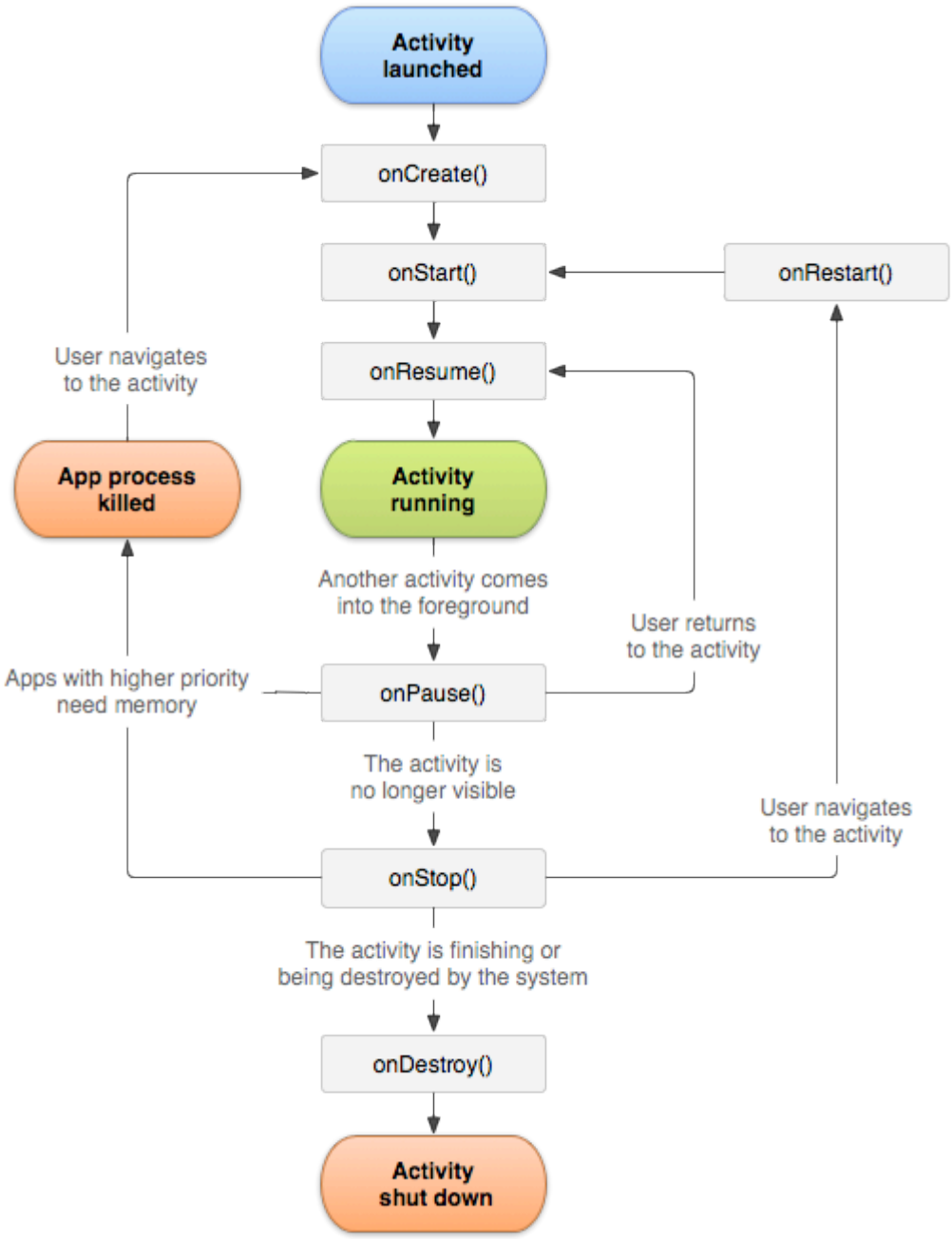
Obr. 5

Diagram na Obr. 5 znázorňuje komunikáciu a vzťahy medzi jednotlivými triedami v MVC architektúre. Štandartnú verziu MVC som rozšíril o presenter, ktorý má za úlohu formátovanie výstupných údajov, ktoré sa zobrazia vo viewe. Samotný view neobsahuje žiadnu aplikačnú logiku, iba zobrazuje dáta, ktoré mu poskytne presenter. V skutočnosti MVC v sebe skrýva na pozadí ešte jeden návrhový vzor **Observer**. Doležité je všimnúť si, vzťah medzi modelom a presenterom. Tieto dve triedy medzi sebou nekomunikuju priamo a nie je medzi nimi žiadna priama väzba. Presenter **sleduje** zmeny dát v modeli a

prezentuje ich následne viewu, čo vytvára dobrú modularitu aplikácie, kedže aplikačná logika nie je naviazaná na prezentačnú vrstvu aplikácie. Môžeme napríklad jednoducho zmeniť dizajn bez zmeny Modelov. Inak povedané, zmena stavu modelu notifikuje presenter, o zmene a ten následne aktualizuje zobrazené údaje vo viewe. Každá správne navrhnutá aplikácia pre Android pozostáva z niekoľkých menších MVC stackov.

Základnou stavebnou jednotkou každej mobilnej aplikácie pre Android je **component Activita**. Activita plní v Androide úlohu **controllera**. Každá Activita má svoj view. **View** je v Androide reprezentovaný v podobe xml súboru v ktorom je definovaný vzhľad pomocou tagov pre danú aktivitu, podobne ako HTML stránky. Úlohu modelov plnia ďalšie komponenty aplikácie, ako napríklad Adaptér-y, či Services, ktoré obsahujú aplikačnú logiku.

Každá activita má svoj životný cyklus, ktorý popisuje nasledujúci diagram.



- Autentifikácia používateľa

Autentifikáciu používateľa zabezpečujú dve activity, prvá slúži na prihlásenie do aplikácie a druhá slúži pre registráciu nového používateľa. Spolu tvoria veľmi jednoduchý autentifikačný modul aplikácie. Na prechody medzi jednotlivými aplikáciami používam **Intenty.** Intent je v Androide objekt, ktorý zvyčajne obsahuje nejakú správu a slúži na komunikáciu medzi rôznymi komponentami aplikácie (medzi komponenty patria napríklad

activity a services). Jedno s možných využití Intentu je práve vytváranie a spúšťanie nových activít. Intenty v mojej aplikácii používam najmä na prechod medzi activitami.

V prípade, že používateľ zadá nekorektné prihlasovacie údaje, zobrazí sa dialógové okno s upozornením pre používateľa. Na vytvorenie objektu dialógového okna je použitý návrhový vzor Builder. **Builder** má využitie v prípade ak potrebujeme pri vytváraní objektu zadávať veľa hodnôt rôznych povinných aj nepovinných atribútov. Pri bežnom prístupe by sme museli použiť konštruktor s veľkým množstvom parametrov s defaultnými hodnotami, čo nie je vhodné. Builder pattern nám umožňuje zadávať hodnoty atribútov ako parametre setterov v ľubovoľnom poradí a následne vytvoriť samotný objekt.

Príklad použitia:

```
AlertDialog.Builder builder = new AlertDialog.Builder(LogInActivity.this);
builder.setMessage(R.string.login_error_message)
    .setTitle(R.string.login_error_title)
    .setPositiveButton(android.R.string.ok, null);
AlertDialog dialog = builder.create();
```

Po úspešnej authentifikácii používateľa sa zobrazí activita s hlavným menu. Obsahuje tlačidlá na prístup k jednotlivým častiam aplikácie s tlačidlo na odhlásenie používateľa. Je hlavným prístupovým bodom aplikácie.

- Návrh zobrazenie údajov

Na zobrazenie zoznamu údajov som použil Android component ListView. Zaujímavé je ako ListView napĺňame dátami. Je potrebné použiť tzv. Adaptér. Adaptér je objekt, ktorý premosťuje ListView (alebo iný AdapterView) s dátami, ktoré má zobraziť. Na toto premostenie je použitý Adapter Pattern, ktorý vo všeobecnosti umožňuje spoluprácu dvoch tried s rozdielnymi interfejsmi (typmi). Adaptér okrem iného vytvára view pre každý riadok zobrazený v ListView componente.

V mojej aplikácii budem potrebovať zobraziť zoznam pripomienok a zoznam záznamov glykémii, každý zoznam má svoj vlastný Adaptér.

- Dáta a synchronizácia

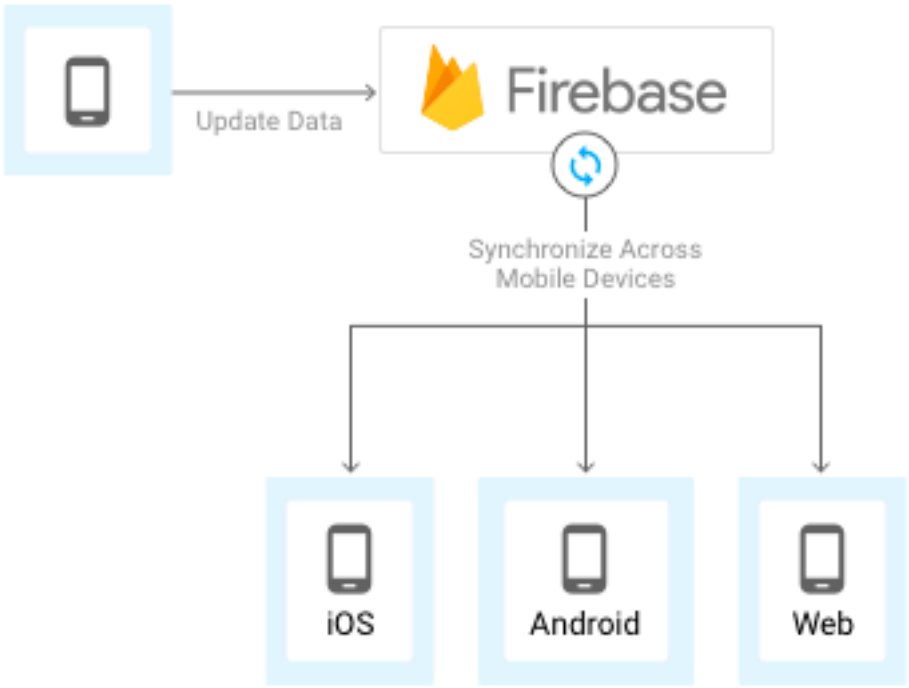
Môj prvotný návrh synchronizácie údajov aplikácie s webovou časťou spočíval v priamom pripojení sa na SQL databázu, ktorá by bežala na serveri. Synchronizácia dát mala prebiehať medzi dátami uloženými v lokálnej SQLite databáze a databázou na serveri. Plánoval som definovať listenery, ktoré by vedeli reagovať na zmeny dát v databáze a spúšťať synchronizáciu. Veľmi rýchlo som však zistil, že to nie je v mobilnej aplikácii možné. Zistil som, že jedinou cestou, ako riešiť online synchronizáciu dát medzi webovou a mobilnou aplikáciou, je použitie HTTP POST requestov. Pre Android existujú knižnice, ktoré ich vedia posielať requesty s dátami a spracovávať odpovede asynchrónne (Volley a Retrofit 2). Vo webovej časti aplikácie je možné použiť prakticky akýkoľvek jazyk, v mojom prípade som použil PHP. Na samotné spúšťanie synchronizácie ponúka Android riešenie v podobe synchronizačného adaptéru, ktorý beží na pozadí a spúšťa synchronizáciu v prípade zmeny dát, alebo v určitých časových intervaloch. Synchronizačný adaptér zaregistruje aplikáciu v nastaveniach telefónu v sekcii synchronizácia. Jeho implementácia je náročná, vyžaduje množstvo komponentov. Napríklad account či content provider nad lokálnou databázou. Content Provider v Androide plní funkciu dátového wrappera. Je to jediný spôsob ako môže aplikácia poskytnúť svoje dáta inej aplikácii a množstvo Adaptérov tiež vyžaduje použitie Content Providera, ktorý vráti objekt Cursor na dáta v databáze, Adaptér ich následne pomocou kurzoru vyberie z databázy. Cursor sa dá chápať ako ukazovateľ na jeden riadok v tabuľke, ktorý sa vie posúvať po ďalších riadkoch.

Nakoniec som zavrhol aj toto riešenie, pretože keď som začal s implementáciou, sa objavilo veľké množstvo netriviálnych problémov. Napríklad zaznamenávanie zmien v prípade, že je používateľ offline, potom ich následná synchronizácia, riešenie konfliktov pri synchronizácii v mobilnej aj webovej časti aplikácie, autentifikácia používateľa a pod... Aj po vyriešení všetkých týchto problémov, by som nedosiahol pravdepodobne synchronizáciu dát v reálnom čase. Preto som hľadal iné, vhodnejšie riešenie.

- Firebase Real Time databáza

Ponúka komplexné riešenie synchronizácie údajov v reálnom čase medzi mobilnou a webovou aplikáciou u všetkých pripojených klientov súčasne. Toto synchronizačné

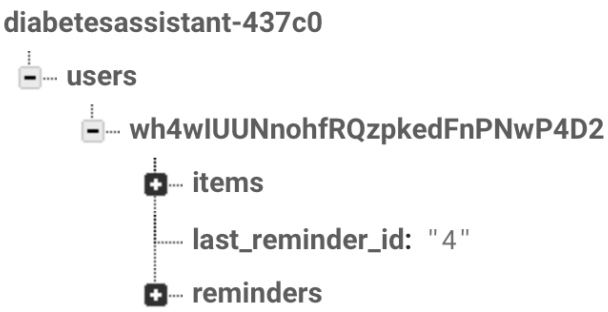
riešenie, spĺňa všetky moje požiadavky. Je to centralizovaná databáza, ktorá pre aplikáciu slúži ako backend a v ktorej sa údaje všetkých pripojených klientov synchronizujú v reálnom čase. Pre mobilnú aplikáciu poskytuje aj API, ktoré zároveň slúži aj ako offline databáza. Okrem databázy, ponúka aj modul pre online autentifikáciu používateľov. Nasledujúci diagram znázorňuje akú rolu zastupuje Firebase Real Time databáza v aplikácii.



Obr. 6

Firebase databáza nie je SQL databáza, je to dokumentová databáza a dáta v nej sú uložené v JSON formáte. V skutočnosti, celá databáza je jeden veľký JSON súbor. Ako hlavnú nevýhodu použitia Firebase vidím najmä v tom, že dáta sú uložené na serveroch Googlu a teda ich nemám plne pod kontrolou. Rovnako je zložitejšia tvorba komplexnejších dopytov do databázy a tiež nutnosť dávať pozor na štruktúru dát, keďže dokumentové databázy nemajú pevne danú schému. Pre potreby mojej aplikácie sú tieto nevýhody zanedbateľné, bolo však potrebné navrhnúť dátovú štruktúru databázy iným

spôsobom ako pri SQL databázach. Nasledujúca schéma znázorňuje základnú štruktúru dát vo Firebase databáze.



Obr. 7

Na najvyššej úrovni sú zaregistrovaní používatelia, aby som vedel ľahko rozlíšiť, ktoré dáta patria, ktorému používateľovi, jednotliví používatelia sú identifikovaní svojim ID. Pod každým ID používateľa sa nachádzajú jeho dáta, ku ktorým má prístup iba on a žiadny iný používateľ k nim prístup nemá. Dáta pozostávajú z dvoch častí, záznamy glykémii (items) a pripomienky(reminders). Pri každom používateľovi si ukladám aj id poslednej vytvorenej pripomienky, ktoré musí byť vždy unikátne.

- Webová časť aplikácie

Webová časť aplikácie slúži ako administračné rozhranie pre aplikáciu, primárne určené pre rodičov diabetika na kontrolu. Obsahuje niekoľko možností navyše oproti mobilnej aplikácii. Podobne ako pri synchronizácii, návrh nebol jednoduchý. Pri pôvodnom návrhu synchronizácie som plánoval použiť MySQL na serveri a PHP ako jazyk pomocou, ktorého by som vedel posielať dátové HTTP responsy a requesty späť do mobilnej aplikácie. Mala to byť bežná server-side webová aplikácia v čistom PHP, HTML a CSS. Už teraz mi bolo jasné, že to úplne synchronizácia v reálnom čase nebude a na aktualizáciu údajov na webe bude treba minimálne reload stránky.

Po rozhodnutí použiť Firebase ako databázu som prišiel na to, že robiť webovú časť aplikácie ako server-side aplikáciu nie je rozumné, Firebase platforma nemá ani natívne API pre PHP. Potom som objavil skvelý frontendový framework **Angular**, ktorý má

podporu pre Firebase a je ideálny na tvorbu webových rozhraní mobilných aplikácii. Je to deklaratívny componentový MVC framework v ktorom sa programuje v jazyku TypeScript, čo je rošírenie JavaScriptu o podporu typovania premenných. Kompiluje sa do JavaScriptu. Deklaratívny znamená, že programátor napíše kód, ktorý popisuje, čo chce aby robil, ale nie krok po kroku ako pri imperatívnom programovaní. Napríklad minimalizuje použitie cyklov a skôr využíva prvky funkcionálneho programovania. Angular umožňuje definovať vlastné html componenty, rovnako poskytuje aj router na jednoduchú navigáciu medzi stránkami.

// TODO implemetacne detaily

1. Dátum a čas kedy bol záznam pridaný
2. Kategória záznamu
3. Hodnota nameranej glykémie
4. Množstvo pichnutého rýchleho inzulínu
5. Množstvo pichnutého pomalého inzulínu
6. Textová poznámka

Activita ponúkne diabetikovi prehľadný zoznam záznamov nameraných glykémii.

- Activita so zoznamom pripomienok

Zobrazí zoznam nadefinovaných pripomienok, pričom každá položka zoznamu zobrazuje informácie:

1. Názov pripomienky
2. Kategória pripomienky
3. Čas

4. Prepínač pomocou, ktorého sa dá vypnúť, resp. zapnúť daná pripomienka

2.2 Návrh Real Time databázy a obmedzení prístupu

Firebase real time databáza je dokumentová databáza. Ukladá dáta v json formáte. Celú databázu tvorí jeden veľký json dokument. K samotnej databáze pristupujeme cez webovú konzolu, kde môžeme priamo pridávať, editovať, mazať, importovať a exportovať dáta.

Štruktúru databázy pre moju aplikáciu som navrhol, tak aby som vedel jednoducho odlíšiť dáta jednotlivých používateľov. Základom je node users, ktorý obsahuje nody s id jednotlivých používateľov. Pod nodami s id číslami sa nachádzajú dáta pre daného používateľa. Toto mi umožňuje jednoducho identifikovať dáta prihláseného používateľa, stačí mi poznať jeho používateľské id. Dáta používateľa pozostávajú z dvoch hlavných častí. Prvou sú záznamy a druhou sú pripomienky.

Definoval som aj prístupové pravidlá pre čítanie a zapisovanie, tak aby daný používateľ vedel čítať a zapisovať iba jeho dáta a žiadne iné.

2.3 Návrh webovej aplikácie

Webová časť aplikácie bude slúžiť ako administračné rozhranie, primárne určené pre rodičov a zákonných zástupcov. Podobne ako mobilná aplikácia, je prepojená s firebase autentifikáciou a real time databázou. Všetky dáta sú medzi webovou časťou aplikácie a mobilnou aplikáciou synchronizované v reálnom čase, pričom webové rozhranie bude umožňovať vykonávať viac operácii v porovnaní s mobilnou aplikáciou. Je to najmä z toho dôvodu, že dieťa by nemalo mať možnosť napríklad odstrániť nejaký záznam, či zeditovať čas nejakej pripomienky, toto bude možné iba cez webovú časť. (TODO, možno túto funkcionalitu bude obsahovať aj mobilná aplikácia, avšak bude chránená nejakým bezpečnostným kódom. Všetky stránky sú responzívne.

2.3.1 Návrh používateľského rozhrania

3 Implementácia

Android service, activity, fragmenty, broadcasty, receivery, intenty

Synchronizacia, firebase, listenery, persistencia dat (co sa deje ak som offline)

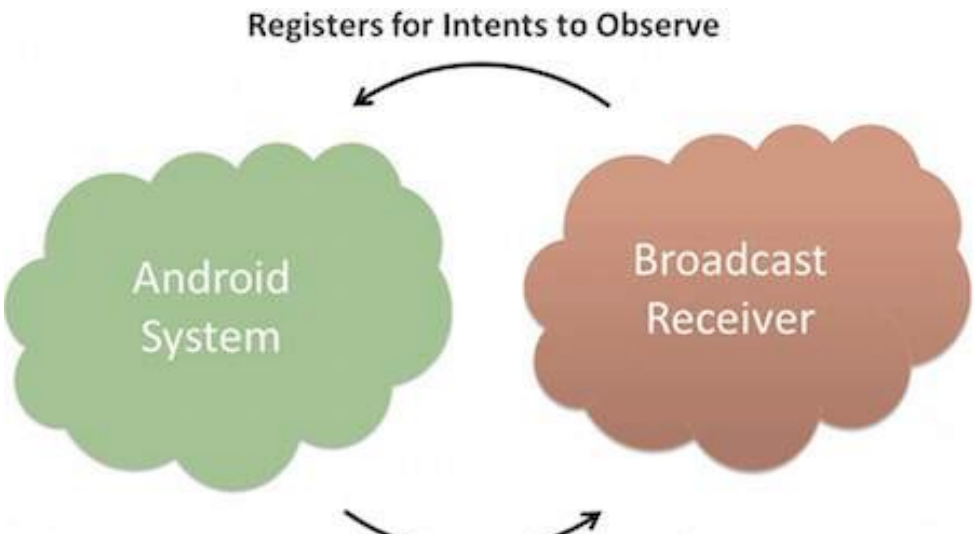
Two way data binding, dependency injection, componenty, typescript

3.1 Implementácia mobilnej aplikácia

Mobilnú aplikáciu som vyvíjal pre operačný systém Android.

- Implementácia pripomienkovača

Pripomienka je v podstate alarm notifikácia, ktorá používateľa aplikácie určitým spôsobom upozorní na udalosť počas dňa, v mojom prípade čas jedla, pichnutia inzulínu a podobne. Z hľadiska zariadenia to znamená, že sa musí prebudiť v určitý čas, zobraziť na obrazovke activitu, prehrať zvuk, vytvoriť notifikáciu a zavibrovať. Implementácia alarmov v Androide nie je triviálnou záležitosťou. Prvým krokom je pochopenie konceptu Broadcastov v Androide. Operačný systém komunikuje s aplikáciami, ktoré sú nainštalované v telefóne pomocou Broadcastov. Ak nastane určitá udalosť v systéme, ten rozpošle broadcastovú správu v podobe Intentu. Toto je ďalšie využitie Intentov. Každá aplikácia potom môže implementovať špeciálnu triedu BroadcastReceiver, v ktorej vie zachytiť danú udalosť v podobe intentu a vykonať určitú akciu. Je tu využitý návrhový vzor publish-subscribe, ktorý slúži na rozposielanie správ určitej množine potenciálnych príjimateľov.



Obr. 8

Android poskytuje API AlarmManager pomocou, ktorého je možné nastaviť čas kedy systém rozpošle broadcast (zapne sa alarm), tento broadcast je následne možné zachytiť v Receiveri, ktorý je potrebné definovať a vykonať požadované akcie (rozsvietiť display, zobraziť activitu, prehrať zvuk...).

Najväčší problém, ktorý som riešil pri návrhu pripomienkovača bol, akým spôsobom navrhnúť modul, ktorý bude vedieť manažovať stav viacerých alarmov súčasne.

3.1.1 Ukladanie a synchronizácia dát

Text

3.1.2 Implementácia pripomienok a ich synchronizácia

Text

3.1.3

3.2 Implementácia webovej aplikácie

3.2.1 Two way data binding a dependency injection v typescripte

4 Testovanie

4.1 Testovacie scenáre

- Scenar 1
- Scenar 2

4.2 Výsledky testovania

Záver

Text zaveru

Literatúra a internetové zdroje

[1] Zdroj

[2] Zdroj

[3] Zdroj

[4] Zdroj

[5] Zdroj

[6] Zdroj

[7] Zdroj

[8] Zdroj

Prílohy

Zoznam priloh k bakalarskej praci